PhD-FSTM-2026-051
The Faculty of Science, Technology and Medicine

DISSERTATION

Defence held on 20/04/2026 in Esch-sur-Alzette

to obtain the degree of

DOCTEUR DE L'UNIVERSITÉ DU LUXEMBOURG

EN INFORMATIQUE

by

Vincent Jacob DE WIT

PRINCIPLED DUAL-SCALE DETACHMENT FOR IHIBO

Dissertation defence committee

Prof. Dr. Leendert (Leon) van der Torre, dissertation supervisor
Professor, Université du Luxembourg

Prof. Dr. Jan Broersen
Professor, Utrecht University

Prof. Dr. Christoph Schommer, Chairman
Professor, Université du Luxembourg

Prof. Dr. Antonino Rotolo
Professor, Alma Mater Studiorum Università di Bologna

Dr. Amro Najjar
Researcher, Luxembourg Institute of Science and Technology

Affidavit / Statement of originality

I declare that this thesis:

- is the result of my own work. Any contribution from any other party, and any use of generative artificial intelligence technologies have been duly cited and acknowledged;
- is not substantially the same as any other that I have submitted, and;
- is not being concurrently submitted for a degree, diploma or other qualification at the University of Luxembourg or any other University or similar institution except as specified in the text.

With my approval I furthermore confirm the following:

- I have adhered to the rules set out in the University of Luxembourg's Code of Conduct and the Doctoral Education Agreement (DEA)[1], in particular with regard to Research Integrity.
- I have documented all methods, data, and processes truthfully and fully.
- I have mentioned all the significant contributors to the work.
- I am aware that the work may be screened electronically for originality.

I acknowledge that if any issues are raised regarding good research practices based on the review of the thesis, the examination may be postponed pending the outcome of any investigation of such issues. If a degree was conferred, any such subsequently discovered issues may result in the cancellation of the degree.

Approved on 2026-04-13

[1] If applicable (DEA is compulsory since August 2020)

Acknowledgements,

I acknowledge that I would not have been able to realize this project without the contributions of many people. Herewith, I express my gratitude to all those who supported me throughout this doctorate journey.

First and foremost, I thank my supervisors, Prof. Dr. Leendert (Leon) van der Torre, Dr. Réka Markovich and Dr. Amro Najjar, for their insights and for their time during meetings. Also, Prof. Dr. Herwig Hofmann, for joining my CET committee and for organizing the DILLAN project. I thank all CET members for their time and expertise. My appreciation extends to the DILLAN project team: fellow DILLAN PhDs and teachers for creating an inspiring interdisciplinary environment. I also thank the ICR and CLAIM groups, the Department of Computer Science, and the University for providing a research environment. Of course, I am also grateful to the members of the jury, Prof. Dr. Jan Broersen and Prof. Dr. Antonino Rotolo and Prof. Dr. Christoph Schommer, for their time in reading and commenting on my dissertation. In addition, Dr. Liuwen Yu and Dr. Aleks Knoks contributed significantly.

Also, special thanks to all my co-authors for writing papers with me. I am indebted to my colleagues, the postdocs (Davide, Pere, Luca, Apostolos, Kenneth, Matteo, Salma, Huimin) and PhD peers (David, Adam, Xu, Benoit, Richard, Thomas, Matti, Aria). Members of the research staff: Dr. Emil Weydert, Dr. Christian Franck, Dr. Sana Nouzri, Dr. Igor Tchappi, Dr. Joris Hulstijn and, of course, Prof. Dr. Dov Gabbay at the University of Luxembourg, for their stimulating discussions and collaboration.

I really appreciated sharing knowledge and camaraderie with the LAST-JD students(Liuwen, Cecilia, Alesandro, Marco, Asif, Giuseppe, Rachele, Rachele, Beatrice, Generoso, Nino, Jie, Qingjing), visiting students (Matti, and others), ZLAIRE (Muyun, Yini, and others), the researchers I met at Trois-Riviere University and Vienna University (Josephine, Dmitry, Dominik, Blaz and Luca) and lastly Joris Graff. I also acknowledge the administrative staff for their constant and efficient support. Additionally, I thank Dr. Mirko Zichichi for helping me get started with the smart contracts and Dr. Llio Humphreys for proofreading the text; they were very helpful to the project. Not forgetting the scholars I met and conversed with at conferences.

Finally, I am grateful to my family, friends, and the yoga community for their unwavering encouragement and balance during challenging times. This work would not have been possible without your support. Especially my parents and sisters, who have supported me and my interest in AI through many challenges, from school to university. And of course, I thank the helpful teachers, colleagues and other students I met and worked with along the way.

Vincent de Wit

Belval, Luxemburg, April 2026

Dedicated to the evolution of Reason.

Table of Contents

List of Figures

List of Tables

Summary

This work in Computer Science and more specifically Artificial Intelligence studies balance scales and applies them to a protocol for documenting decision-making processes. A balance scale has two scales for placing weighted elements, which are then aggregated and compared. Mathematically, this is a comparison between two weighted sums which evaluate to true (1) or false (0). These linear product comparisons occur across many fields of research, including artificial neural networks, linear algebra, and economics. This dissertation investigates the use of this method for solving multi-agent decision-making problems. This approach is particularly contrasted with Abstract Argumentation, and the new approach is computationally efficient.

These equations are also used in ethics, specifically to model the balancing view of the reason-based approach to deontic logic. A balance scale is used to weigh and compare weighted reasons for and against a particular action. If the weight of the reasons in support of the action is greater than the weight of the reasons against it, then the action is permitted. A recent book by Chris Tucker, "The Weight of Reason," shows, however, that there are two weights per reason and that two balancing scales are needed to determine the deontic status of an action. The question here is whether this new approach can be studied using so-called detachment functions. These functions have been developed by other authors in earlier work on the reason-based approach. The detachment functions provide a framework that allows for a principle-based analysis of the dual scale.

After this principle-based analysis, the dissertation turns to the development of a smart contract that implements balancing reasons for an existing documentation protocol(IHiBO). The documentation protocol has been developed to improve the trust relationship between investors and fund managers. It achieves this by documenting the fund managers' decision-making process before portfolio decisions. Here, the question is whether the balancing reasons approach can improve an already present smart contract. A proof-of-concept new smart contract that implements balancing reasons has been developed in this dissertation. The new smart contract is computationally less expensive in a limited test case than a comparable existing smart contract in the protocol.

Then, to better understand the smart contract, the case of multi-agent dynamic dual-scale detachment is discussed. What are the limitations of the multi-agent dynamic dual scale? Using the fund manager multi-agent circumstance in combination with the dynamic dual scale to better understand

what both have to offer each other. A main problem in multi-agent dynamic dual-scale is identified. It is possible for a group of agents to collectively accept an action as permitted, while each individual would reject its permissibility. Further current limits are identified that prevent the system from being more deeply compared with abstract argumentation and other approaches.

Chapter 1

Introduction

1.1 Introduction

Artificial Intelligence has seen multiple major advances over the past four years, including the release of the first LLM chatbots such as ChatGPT. With the release and continued development of chatbots, multiple problems are becoming less science fiction and more real problems for AI developers [1]. Some of these problems are major social problems and therefore also require legal intervention; this is already happening, for example, with the EU's AI-Act. The main problem addressed here is how to regulate the increasingly intelligent artificial intelligence agents being developed. This problem has been the subject of multiple films; however, the developers of these real LLMs are now encountering the first issues. Especially since the systems are very large and have many parameters, understanding their functioning is difficult and it is often computationally intractable to verify findings. However, as demonstrated by the AI-Act (and European Data Protection Supervisor, 2025), regulation focuses on the inputs and outputs of these systems and categorizing them in levels of danger. The discussion thus centers around systems for decision-making, the process of collecting information and then selecting an outcome based on that input.

Artificial Intelligence is the area that studies intelligence with the goal of recreating artificial forms of it. This includes: problem solving; knowledge, reasoning and planning; uncertain knowledge and reasoning; learning; and communicating, perceiving and acting (Norvig & Intelligence,

[1] https://www.lesswrong.com/posts/WewsByywWNhX9rtwi/current-ais-seem-pretty-misaligned-to-me

2002). To model these topics mathematics and computer science tools are used. Mathematics or more generally Logic is used to model the theory while Computer Science provides hardware and software that implement such models. With symbolic AI methods and formal theories of practical reasoning, decision-making can be made transparent and explainable, thereby increasing trust in the process. This dissertation investigates the implementation of such a formal theory into an existing decision-documentation protocol to enhance trust, expressiveness, transparency, and computational cost feasibility.

Fund management involves a group of managers making decisions about portfolios on behalf of investors (Maginn et al., 2007). Managers can control the portfolio's content, but they cannot control how the market perceives its value. When bringing in new investors, managers discuss multiple variables with their investors, e.g., risk, amount of money invested, and timeframe. This allows managers to make portfolio decisions aligned with strategies acceptable to investors. Some cases are simple, when all the managers agree; other cases give rise to disagreements among the managers. Some managers favor moving in one direction due to specific information while others justify moving in a different direction for other or the same information. Moreover, some information may carry more weight than others. The fund managers may identify information supporting different directions; however, they need to reach a consensus on the content of the portfolio.

To model this process, we turn to the literature on balancing weighted reasons. Recent work includes a book, "The Weight of Reason" by Tucker (2025), that deliberates on the dual-scale approach to balancing reasons, in which a reason has not one but two weights: a justifying weight and a requiring weight. A reason's justifying weight contributes to making the option it supports acceptable, whereas a reason's requiring weight may render the option required by rendering the alternative options unacceptable.

1.2 Background

This dissertation is mostly based on three strands in the literature: Weighted Reasons (Tucker, 2025), Detachment Functions (Knoks & Van der Torre, 2023), and IHiBO: the Intelligent Human-input-based Blockchain Oracle (Yu, Zichichi, Markovich, & Najjar, 2022).

Weighted reasons. These elements are studied in the balancing view to reason-based deontic logic. The balancing view approach assigns weights to reasons and determines the permissibility of each option by placing the reasons' weights on a balance scale. A recent book, "The Weight of Reasons" by Tucker (2025), synthesizes several articles on the weighted reason-based approach and elaborates on them. The book introduces many factors to consider when balancing reasons and outlines how to do so correctly. That makes it distinct from reason-based normative reasoning, which focuses on qualitative rather than quantitative evaluation and, as such, does not assign weights. In the latter approach, reasons are treated as foundational concepts that explain or deduce which actions are permitted and which are not in the deontic sense of "What action is permitted?" and "What action is not forbidden?". This line of research was established by Horty (2007); however, it has deep roots, as seen in Aristotle's work on the 4 causes (or reasons) and Aquinas's work on the unmoved mover (or first reason, or reason without reason).

Detachment Functions. These functions were introduced by Knoks and Van der Torre (2023), in which the detachment function takes a set of reasons and one option, and then "detaches" the deontic status of that option based on the reasons provided. Thus, the detachment function indicates which options are permissible and which are not. The article then shows multiple ways in which this detachment function can be instantiated, understood as different semantics or implementations of the function. The main result is a comparison between the different instantiation methods. The comparison is carried out using a principle-based analysis, a method described in the methodology section of this chapter.

IHiBO. The Intelligent Human-input-based Blockchain Oracle was introduced by Yu, Zichichi, Markovich, and Najjar (2022). IHiBO is a blockchain oracle that implements multi-agent abstract argumentation theory with preferred semantics, taking input arguments from managers and documenting a set of arguments representing permissible actions based on the inputs. IHiBO was developed to improve the trust relationship between trust fund managers and trust fund investors. By documenting decision-making, IHiBO contributes to improving investor trust. More precisely, the documentation process is implemented as a smart contract, making explicit what is documented and how it is documented before decisions regarding the trust fund are made. In this way, the documentation process can be reviewed by investors prior to their investment. This secures investors' legal right to seek insight into their fund managers' actions. Investors have the right to investigate whether managers act in their best interests, as the managers are required to do.

1.3 Research Question

The IHiBO protocol currently comprises two smart contracts that can be executed on the Ethereum blockchain: Negotiation and abstract argumentation. This code implements the respective theory and is executed on the Ethereum blockchain (on-chain), ensuring that it can be verified in advance and that a record is kept on the blockchain. Keeping a process on the blockchain is transparent because no third party can influence the execution of the code. The only risks are those inherent to DLT, namely attacks on the blockchain itself; many consider blockchain or DLT the state of the art of security. However, on-chain execution of code and documentation is expensive because DLTs rely on distributed computation for security. Therefore, minimizing on-chain documentation and computational costs is highly desirable. Furthermore, this cost minimization should not hinder users when specifying their inputs to the smart contract. Therefore, the main research question is the following:

RQ. How can dual-scale detachment improve IHiBO's input/output expressivity, transparency, and on-chain feasibility?

To address this research question, multiple objectives must be addressed. The first objective is to understand recent developments toward the balancing view in depth and to express it as a detachment function, enabling its study using principle-based analysis. This will allow the balancing model to be compared to other approaches. The second objective is to implement balancing reasons as a smart contract for IHiBO. The main objective here is to determine whether the calculations and data structures of balancing can be implemented as a smart contract. And to see how the smart contract's operation cost compares to at least one other smart contract of IHiBO. The third and final objective is to discuss the constructed system to identify potential caveats, limitations, and areas for future work. To formulate these as questions, the following three sub-questions are found.

1. How can the dual-scale approach be analyzed as a detachment function?
2. How does on-chain instantiation of balancing weighted reasons compare to the existing IHiBO-AA(Abstract Argumentation) smart contract?
3. What are the limitations of balancing reasons in IHiBO?

Principle-based analysis of Dual-Scale Detachment. The first sub-question is: How can the dual-scale approach be analyzed as a detachment function? Dual-scale has been succinctly formulated by Tucker (2025). The first goal is to express the dual-scale in terms of a detachment function (Knoks & Van der Torre, 2023). The second goal is to analyze the dual-scale as a detachment function using the principle-based approach. Using the principle-based analysis, dual-scale detachment can be compared to other detachment functions.

Reason-based balancing implementation in IHiBO. The second sub-question is: How does on-chain instantiation of dual-scale detachment compare with the existing IHiBO-AA (Abstract Argumentation) smart contract? By rendering dual scale as a detachment function and analyzing and comparing it through the principle-based analysis, it should be easier to implement the dual-scale approach in IHiBO. Before implementing balancing in IHiBO, the application domain (fund management) should be understood to ensure the approach is tailored to it. After this, the dual-scale implementation can be compared to a different comparable IHiBO smart contract.

Reconsidering the construction. The third and final sub-question is: What are the limitations of dual-scale detachment in IHiBO? Here, the goals are the following. 1) to see clearly the difference between the theory implemented for IHiBO and dual scale detachment. 2) to understand not just dual-scale detachment or just IHiBO, but to understand what IHiBO introduces to detachment and what detachment introduces to IHiBO. And 3) to better understand both detachment and IHiBO outside of their direct implementation area. This is also where additional problems and caveats arising from the system's use are highlighted.

1.4 Methodology

Each sub-question is answered using a different methodology. The first uses a logical method called principle-based analysis. The second uses a Computer Science method called the development cycle. The third question is addressed with a critical discussion.

Principle-based analysis. Principle-based analysis, or the axiomatic method, is a common method in mathematics and logic (Hintikka, 2011) and also used in the analysis of abstract argumentation (Amgoud et al., 2017; Baroni & Giacomin, 2007; Kampik, 2022; Van der Torre & Vesic, 2018). The first appearance of the principle based approach used to analyze abstract argumentation is (Baroni & Giacomin, 2007), the most complete survey of abstract argumentation using the principle based approach is the one by Van der Torre and Vesic (2018). The idea is to use properties to study systems; such properties can either be called axioms or principles. Often in mathematics, systems are constructed from axioms, e.g., the set of natural numbers with addition and multiplication defined on it. Axioms are also used in logic to specify how particular connectives work. However, a principle-based analysis takes, in some sense, the opposite direction by using principles (a different word for axioms and a different use). Instead of making these principles constraints in the system, an analysis is performed to determine whether the principle is satisfied (guaranteed to hold) by default under the system specification. In this case, the properties are often called principles rather than axioms.

Note that this work relates to ethics through its relation to weighted reasons. The term ”principle” also refers to an ethical concept. The type of principle-based analysis used in this dissertation is similar to that conducted in abstract argumentation by Van der Torre and Vesic (2018). The article by Spielthenner (2017) uses this ethical type of principles; it considers three concepts: norms, values, and ethical principles. The work presented in this dissertation does relate to ethics and norms, but the concepts, values, and principles are interpreted differently. Also, the work presented does not specify what constitutes a good norm but is instead concerned with the mechanisms underlying proper norms and ethics. A principle-based analysis of dual-scale detachment is presented, in which the dual scale serves as a model for deriving what is ethical using weighted reasons. This dissertation thus relates to the computation of ethics and does not directly assert what is or is not ethical. Nevertheless, the findings of this dissertation may contribute to ethics, particularly in cases involving the aggregation of reasons from multiple agents.

Development Cycle. To implement numerical balancing of weighted reasons as a smart contract for IHiBO, a development cycle is used: theory, implementation, and evaluation. First, the method is specified; this entails defining a feasible theory of numerical balancing of weighted reasons for IHiBO's multi-agent use case in fund management. Second, the method is implemented as a smart contract, in which explicit choices are made about the data structures and functions. The programming language's limitations may conflict with the initial specification, necessitating changes to the method. Third, the smart contract's performance is compared to that of another IHiBO smart contract.

IHiBO "lives" on the Ethereum blockchain, and the computational cost on Ethereum is measured with the so-called GAS metric. GAS is not an acronym just refers to gas in the fuel tank of an engine, in this case, the EVM: Ethereum Virtual Machine. This metric indicates the computational cost of executing the smart contract and is used to determine the financial cost. So far, IHiBO has two smart contracts that can serve as benchmarks to evaluate the new smart contract's performance. Since the code evaluated on the Ethereum network is executed on the Ethereum virtual machine, the results can be compared directly without reference to computer hardware.

Discussion. The developed system is discussed, with the emphasis shifted from obtaining new results to gaining a deeper understanding. Therefore, the analyzed detachment functions and the developed IHiBO smart contract will be elaborated upon and reviewed. This will reveal caveats and how the contribution relates to the literature. Ideally, this method helps those interested in detachment functions see what IHiBO brings to detachment functions, and those interested in IHiBO see how to use the new smart contract. The discussion elaborates on the scope and impact of the developments described in this dissertation on the literature.

1.5 Interdisciplinary Aspects

This work constitutes a scientific investigation in Computer Science, but, as is often the case in AI, it draws on multiple disciplines and approaches. The reason-based approach from philosophy and ethics is combined with blockchain technology from computer science. To achieve this, mathematical modeling is used to simulate the fund managers' process of reaching consensus and to make this process transparent to investors. The relationship between the manager and the investor is subject to legal regulation.

The reason-based approach treats reasons as a foundational concept and is mainly concerned with deriving permissions from reasons (Horty, 2007; Schroeder, 2021b, 2021c). The reason-based approach is broader than permissions and encompasses the general use of reasons to guide the selection of a course of action.

The detachment functions are implemented in first-order logic, the language of mathematics, and are subjected to principle-based analysis, also known as the axiomatic method. The axiomatic method is a staple in post-Russell-paradox set theory (Hintikka, 2011) and the analysis of abstract argumentation (Baroni & Giacomin, 2007).

Distributed ledger technology (DLT), often referred to as blockchain technology, is best known for its implementation in Bitcoin and other cryptocurrencies (Buterin et al., 2013; Nakamoto, Bit, et al., 2007).

IHiBO implements the simulation of a consensus process often referred to as judgment aggregation (Grossi & Pigozzi, 2022; List, Puppe, et al., 2009) in which humans seek consensus to reach a decision. For example, fund managers must agree on how to proceed with the funds they manage. IHiBO simulates and records their deliberation process.

Given the digital nature of financial markets, the IHiBO application has become increasingly relevant. Trading involves large amounts of data and often requires integrating multiple software packages into a single application, enabling traders to view the available information (tradeview.com). IHiBO further paves the way for documenting high-speed automated trading happening up to multiple times per second. This poses the risk of flash crashes (Kirilenko et al., 2017).

The investigation presented in this dissertation was executed as part of the Digitalization, Law, and Innovation PhD program. And indeed, the tools of Computer Science and approaches from philosophy and ethics enabled the aim of contributing to the enhancement of legally regulated situations, such as the investor's right to verify whether fund managers actually operate in the principal's best interest, as the law requires. Thus, this dissertation also addresses the interdisciplinary area of AI&Law.

1.6 Layout of the dissertation

In Chapter 2, the detachment functions for dual scale are specified, and a principle-based analysis is performed. Then, in Chapter 3, a smart contract is introduced and benchmarked that implements a proof-of-concept version of dual scale detachment. In Chapter 4, the developed system is discussed to elaborate on the caveats and other consequences of the contribution. Chapter 5 provides the conclusion. It is followed by the bibliography and the following appendices: A: proofs, B: code, and C: examples.

1.7 Contributions

In this section, a list of the articles is provided on which this dissertation is built. The author of this dissertation is one of the co-authors of these articles. For each article, the contribution of the author is elaborated upon.

1. Reasons in weighted argumentation graphs- Streit et al. (2023)
2. A principle-based analysis for numerical balancing- Knoks et al. (2024)
3. Balancing Normative Reasons for the Intelligent Human-input-based Blockchain Oracle(IHiBO)- de Wit, Yu, et al. (2025)
4. Dual Scale Detachment- de Wit, Knoks, and Van der Torre (2025a) and de Wit, Knoks, and Van der Torre (2025b)

In the first article, an approach was developed to provide a structural description of the contextual dependency of the weight of reasons to balance reasons expressed in first-order logic. I contributed to this article by participating in discussions on the design of the description and by writing.

In the second article, a second rendering of balancing reasons is developed. However, this time as a detachment function, so that it can be subjected to principle-based analysis for comparison with other detachment functions. Also, I participated in meetings and discussions to identify a design that met the needs; in this paper, I contributed little to the text and wrote the proofs.

The third article considers the IHiBO protocol and integrates ideas from balancing reasons and financial management to create a new smart contract for IHiBO. For this paper, I wrote the smart contract, adapted an existing experiment to it, and executed the experiment. Furthermore, I wrote most of the article, although not all of it. Also, there were meetings in which the article's concept was developed; this was not done by me alone.

The fourth article renders the new dual-scale approach to balancing reasons from Tucker's book as a detachment function and analyzes it using the principle-based approach. I participated in meetings to discuss the book and the rendering of the dual scale as a detachment function, and I participated in writing the article. I wrote the proofs by myself and checked their conceptual validity with a co-author.

To some extent, a fifth work should be named, which is my MSc thesis and the conference (de Wit et al., 2021) and journal article (de Wit et al., 2023) that are based on it, and an additional conference poster (de Wit, 2023) implementing it in abstract argumentation. The work develops probabilistic deontic logics that explicitly incorporate multi-agent perspectives, with two of four variants implementing multi-agency. Initial work on this dissertation started by continuing the development of this logic. However, the probabilities and the multi-agent perspective did not readily enable the operations we were interested in. The presented work generalizes beyond weight as probability, possibly still implementing probability by selecting particular axioms. While taking a different stance on the multi-agent part. Also, the approach presented in this dissertation is less focused on syntax and semantics than in normal logical approaches, though elements of this do remain.

Chapter 2

Contextualization of Contribution

This chapter introduces broader background topics that are important to the field and provide context for the rest of the dissertation.

2.1 Pillars of AI

The basic idea of AI has existed for a long time, perhaps as long as humans have made tools, i.e., tools that are themselves intelligent and can modify/improve themselves or create new tools. However, modern AI research is commonly considered to have started in 1956 at the Dartmouth workshop (McCarthy et al., 2006; Moor, 2006). A timeline of this research history is shown in figure 2.1 from (Francesconi, 2022). The graph indicates on the y-axis the amount of research and investment into AI. The graph goes up-to 2020 on the x-axis the paper that introduces this graph is from January 2022. After this date the first major LLM's were released to the public. The inclusion of the LLM companies should strongly increase the rate at which this graph is growing. This increase is due to the risks but also the financial value of LLM's. The timeline shows when research was growing (so-called AI summers) and declining (AI winters). Notable starting point publication is McCarthy's introduction of recommendation systems (1959), and a seminal book on AI research that covers all notable subfields (Norvig & Intelligence, 2002), with a strong focus on AI as the development of agents.

AI research has developed along multiple lines 2.2. Here, learning does not necessarily equate to what is now called machine learning. Learning here means the ability of intelligent beings to

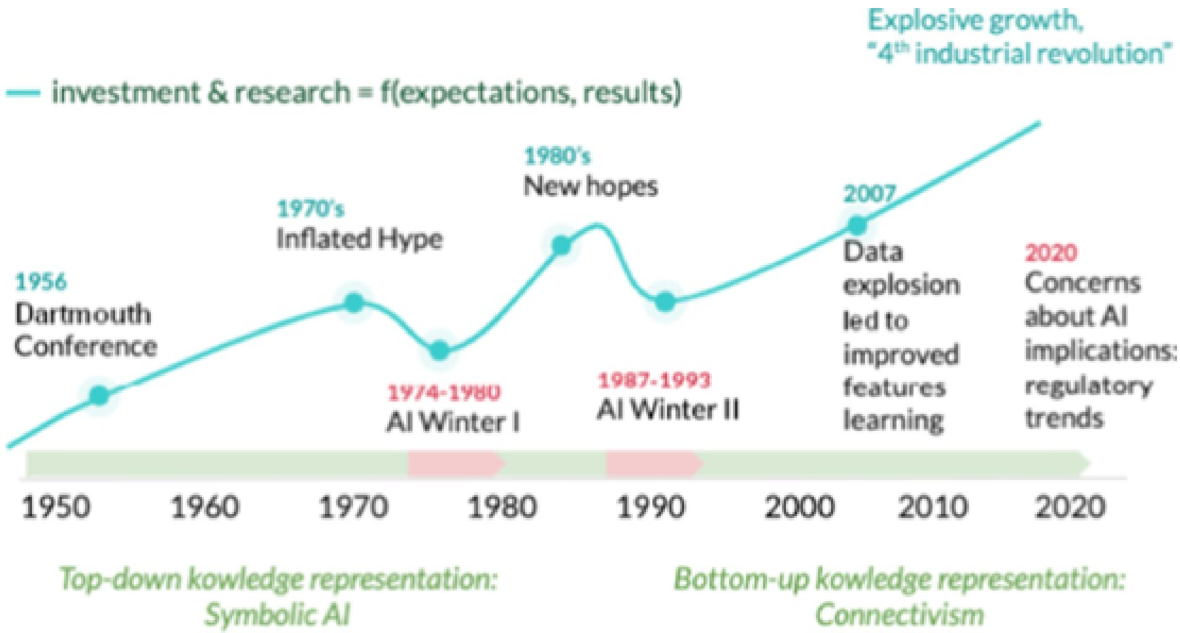


Figure 2.1: Timeline of AI between 1950 and 2020 (Francesconi, 2022)

learn, unlearn, and refine skills. One can call it m-learning as being implemented for machines and to distinguish it from machine learning that implements types of it. However how learning really works is not completely understood as neuroscience is a relatively young field. Therefore saying that machine learning is m-learning is not exactly true.

This list could be longer and more precise; however, it mostly needs to include agency and multi-agent systems (MAS). The latter concerns societies, groups, or collectives of agents, however this may be understood under point 4 natural interaction. An agent is a system that has at least some degree of agency, autonomy, or freedom of action. Studying the interaction of groups of such agents is important. As groups of agents become larger, they may exhibit emergent properties or behavior that can be explained by the group size and that would otherwise be unaccounted for.

The work presented in this dissertation relates to all these elements. It would be overselling to claim that it unifies all these areas in a comprehensive way. However it would also be underselling to claim that it does not unify all these areas in a comprehensive way. A lot of research has been done preceding this work to claim that it combines everything is simply not true. However the contribution does allow concepts of each area to find its way into the picture in a non-trivial way.

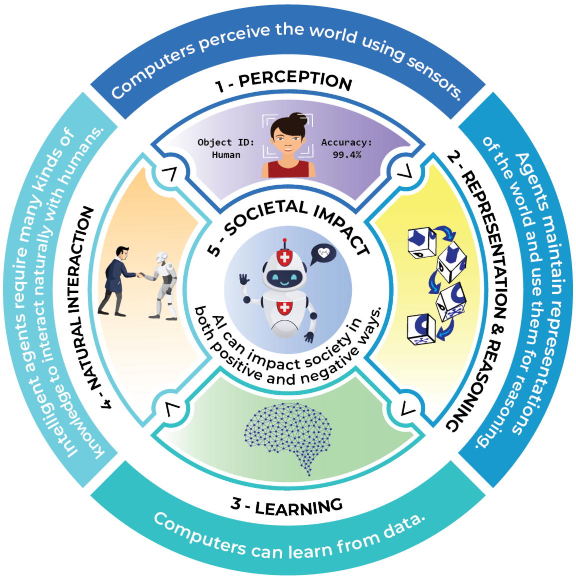


Figure 2.2: Five Big Ideas (AI4K12)

1. Perception: The input elements will be cast as reasons, the colors that are seen are then reasons for them to be viewed as having those colors. Furthermore if reasons are fundamental then maybe we may go as far as saying that the inputs where already reasons before they reach the eyes or ears.

2. Representation and Reasoning: The received inputs are represented as reasons data structures. Then the reasons are used to implement reasoning. Whether the reasoning described captures all possible types of reasoning is a different question. However the approach considers relations among any set of premise reasons and consequences.

3. Learning: The adaptivity of a system can be understood as its capacity to learn. In machine learning this often entails modification of weights since weights are the main carrier of information there. In logic this trait may be approached with non-monotonic logic. Where the consequences can change when new information is added. The proposed methodology implements weights and modification of the weights of weighted elements. Also the methodology implements a non-monotonic relation between premises and conclusions.

4. Natural Interaction: Agent interaction occurs in IHiBO through the smart contract. And is

theoretically than depicted as adding weight systems of different agents. The interaction allows interaction between human and artificial agent since the inputs to the smart contract are submitted as reason data-structures.

5. Societal Impact: The impact of artificial systems can be complex to measure and the AI-act is about this. Also the influence of an artificial system may not be immediately obvious. The methodology however relates to permissions and allows agents to express that pieces of information are reasons to permit particular options. In this case the option has to be made explicit. And the information used as ground for that option needs to written down; whether the information is implicit or explicit.

6. The interaction of these elements: The interaction of these elements is somewhat difficult to study since often these are implemented by different fields of study. The approach presented here may be understood as a unifying method. However accounting for the expression of all these research field in the proposed methodology is a bridge too far. We merely show the methodology, and its properties. And show the tangible contribution that has been made to the literature.

2.2 Symbolic vs Sub-Symbolic Dichotomy

The main split found in AI research is a supposed dichotomy between symbolic and sub-symbolic (typically connectionist) approaches. Many researchers working on connectionist approaches believe that connectionism is the way forward, perhaps even leading to Artificial General Intelligence. Although many researchers working on symbolic approaches also tend to believe that logic is the way to reach this goal, there is some friction between these approaches that can be perceived when working in the area. There are also many other researchers, probably most, who have a more mixed perception of this. Either they do not perceive a strict distinction, or they consider a hybrid approach to be optimal.

Logic is a primary example of a symbolic approach. Logic uses axioms and a consequence relation to relate formulas. This allows users to take a set of formulas and obtain a set of formulas derived as a consequence, thereby gaining knowledge. Volume 1 of the Handbook of Philosophical

Logic (D. M. Gabbay, Guenthner, et al., 2002) provides a thorough introduction to first- through higher order logics. Logic is often considered an old approach, it has roots dating back to the ancient Greeks, i.e., Aristotle. But the way logic is studied in modern times, i.e., modern logic, is relatively new. First-order logic, for example, originated between 1917 and 1928.

Artificial Neural Networks are a primary example of sub-symbolic systems and was first developed in 1957 by Frank Rosenblatt. These systems are instantiated as a directed, often acyclic graph in which each edge has a weight. The vertices of the graph, often called nodes, can be "on" or "off" depending on whether a threshold is met, given an activation function. The instantiation of the weight typically requires large amounts of data specifying the output given an input. The process of finding weights tends to be called "learning" and uses backpropagation, a technique that relies on the computation of many derivatives. Schmidhuber (2015) gives an overview of developments in this area. The geometric deep learning approach is intended to unify and systematize research in this area (Bronstein et al., 2021). The landmark paper Attention (Vaswani et al., 2017) provides an excellent introduction to the transformer architecture that LLMs use.

In recent years, the formal argumentation approach has gained considerable traction. Although formal argumentation is relatively new, argumentation as a field of study is long established. In this approach, two concepts are fundamental: conflict and argument. The central idea is that arguments can be proposed and may be in conflict with other arguments. Some arguments can be accepted together with other arguments, while some arguments cannot be accepted when some other arguments are accepted. Dung introduced abstract argumentation with his paper on the acceptability of arguments (Dung, 1995), Modgil and Prakken 2014 introduced ASPIC+, a system that creates structured arguments using logic to instantiate abstract argumentation frameworks. There is a series of handbooks on formal argumentation, starting from volume 1 (Baroni et al., 2018).

The common feature between these approaches is that they instantiate a relation between an input and an output. An artificial agent can be instantiated with one of the systems so that, given an input, it can return outputs. Furthermore, it can be observed that current numerical approaches are defined in first-order logic set theory, just like argumentation theory is expressed in first-order logic. First-order logic, as the name implies, is a type of logic. This shows that the dichotomy between sub-symbolic and symbolic is superficial to some extent since both are modeled in first order logic. However, there are differences between the systems, for example, in the way they are instantiated, as described above. The problem of explaining AI systems is therefore more about how large systems of elements function. The book Artificial Intelligence: A Modern Approach (Norvig & Intelligence, 2002) is a well-known and respected work laying out the foundations of AI and agent design. Makinson and van der Torre (2000) developed input/output logic, introducing a notation for reasoning with input/output relations and adding more set operations that can be applied to a given input/output relation.

The research area of deontic logic, also known as normative systems, is found at the intersection between first order logic, multi-agent systems and the law. This field is about logics and/or systems that govern norms and is often concerned with specifying definitions and models to reason about what is and is not permitted. The book Introduction to Deontic Logic and Normative Systems (Parent & Van der Torre, 2018) gives a concise introduction to deontic logic and input/output logic. Also, there is a series of deontic logic handbooks, starting from Volume 1 (D. Gabbay et al., 2013). There is also a handbook on multi-agent systems and normative systems (Chopra et al., 2018).

2.3 Input Output Relations

In an artificial intelligence system, the system is given an input and produces an output. Often, the black-box versus white-box system is referred to here. In figure 2.3, the system is colored gray to

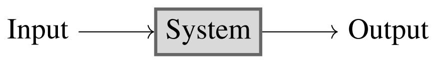


Figure 2.3: I/O system

indicate that its internals may be understood to a limited extent. Input/output logic (Makinson & Torre, 2001; Makinson & Van der Torre, 2000; Parent & Torre, 2013) can instantiate and model such systems. The most basic way of using input-output logic is to simply instantiate the system by setting the output as a pair $(\text{input}, \text{output})$ of the logical system for each input. The output is obtained by feeding the specified input to the system and registering its output.

One can also consider the system as an agent inside an environment. In this case, there is a world state, and the agent takes particular parts of the world state as percepts and then modifies the world in particular ways through actions. See figure 2.4.

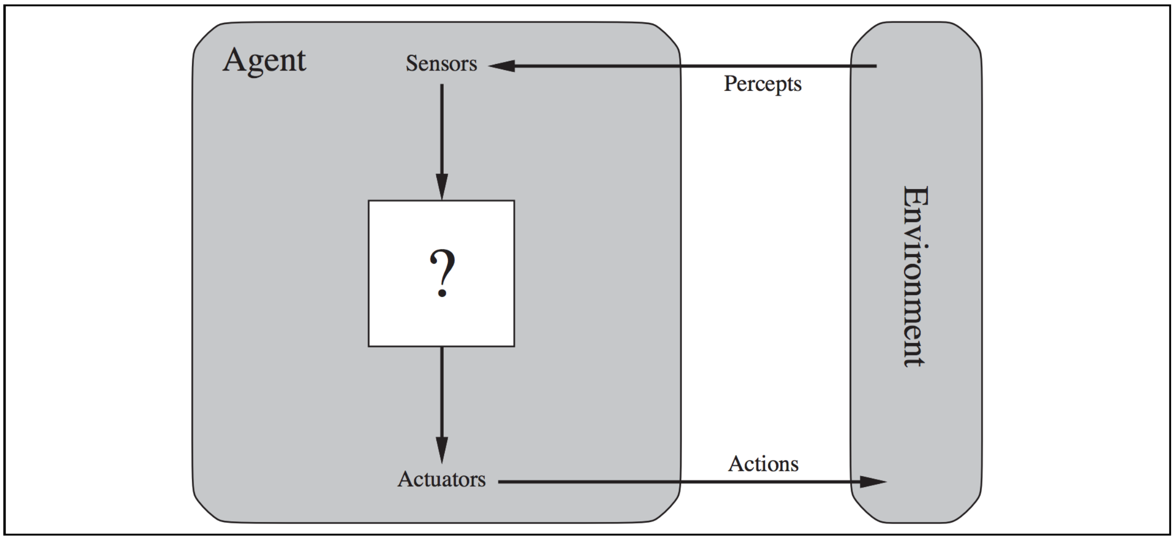


Figure 2.4: Basic Agent

Here, the question mark indicates that it is unclear how the sensors' perceived values relate to the actions the actuators execute. To not tie ourselves to the premises defined in input/output logic, the relation may be called a detachment relation. Here, detachment refers to the basic idea in logic of detachment, also known as *modus ponens*. One may then say that, given the input, the agent "detaches" the given output. The logical notation is then that of implication, i.e. one can informally write input $\rightarrow$ output or $A \rightarrow B$: "if you say A, you should also say B". The relation can be instantiated through function notation. If "?" is a function instantiated by D, then $D(\text{input}, \text{output}) = true$ and the relation notation is $(\text{input}, \text{output}) \in D$.

To ask "How does this system work?" is to ask how "?" or D is instantiated and what its content is.

2.4 The Control Problem or Principal-Agent Problem

The Control problem in AI is about the challenges that arise from granting an agent autonomy and delegating tasks to it. This requires the person delegating the tasks to trust the agent to complete those tasks. Outside AI, this problem is known as the principal-agent problem. This problem occurs, for example, in portfolio management, where investors entrust their money to portfolio managers who they expect to act in their best interests. In portfolio management, the investors are commonly known as principals, and the managers as fiduciaries.

A popular example in Artificial Intelligence is a paperclip machine. First, there is a paperclip factory where humans work to make paperclips. Then automation comes along and improves certain tasks, increasing human productivity. Eventually, humans will be replaced by more productive machines. This is, of course, not so nice for humans; however, this is not where the story ends. The machine is further improved by adding basic AI enhancements to increase pipeline performance. Then, at some point, they also connect the system to the internet and make the AI self-learning, thereby automating the system's improvement. However, at this point, things get out of hand. The machine's goal is to produce paper clips and optimize paperclip production. Things escalate, and the machine evolves into a universe-destroying machine that takes whatever it can to make paperclips.

This example is somewhat extreme, but it illustrates several aspects of what autonomy in autonomous systems can entail. The case of portfolio management is interesting because the portfolio's content can be managed in different ways. Managers use multiple metrics to find an optimal composition for a particular customer at a given time. Over time, the market may change, as may customers' preferences. Therefore, multiple other portfolio compositions may become viable. For managers to inform their customers of their decisions, they need to be able to specify concretely why the decisions were made.

2.5 The Blockchain Oracle Problem

Distributed Ledger Technology (DLT), more commonly known as blockchain, is a relatively new technology that is said to improve upon current security systems. It does so by introducing distributivity to previously centralized systems. For example, banks are centralized and keep records of their customers and their security keys. However, this record can be stored in a decentralized manner, thereby removing the bank as a middleman. The main idea is that the bank is easy prey, since attackers need only attack a single location to recover all keys. If this record of keys is decentralized, attackers will need to do much more work to obtain all the keys. Blockchain or DLT implementations serve as decentralized ledgers. However, these technologies are not without problems. One problem is the oracle problem. In the context of blockchain, oracles are systems that ensure that specific real-world information is available on the blockchain. However, since what is on the blockchain is secure, the act of putting information on the blockchain is sensitive. The oracle introduces a point of failure. If the blockchain is secure and the oracle moves real-world information onto it, that does not guarantee that the information was accurate, trustworthy and secure before it was moved onto the blockchain. If the information were not secure, the blockchain would lose its selling point.

2.6 Commensurability

The work presented here also enters the arena of commensurability, or rather incommensurability. In law, the concept of incommensurability relates to proportionality and legal rights. Law is organized in a particular way and uses rights to grant powers to people or other entities. In certain situations, rights may conflict with one another. To resolve such conflicts, proportionality can be applied to identify a solution that acknowledges the conflict while avoiding unnecessary harm to rights. In the legal domain, there is debate about whether proportionality is appropriate, as it can conflict with absolute rights (Tsakyrakis, 2009). The key concept of commensurability can be understood as comparability. Proportionality assumes that rights can be compared, but some people argue that they cannot, that they are incommensurable. In this dissertation, numbers are assigned to formulas and are balanced against one another. While this approach may suffice for simple cases, such as choosing between a cup of coffee and a cup of tea, there are important legal caveats when applied to more complex problems. As we will see, the theory in this dissertation, which uses numbers to make decisions, stems from ethics. I do not know whether the literature has a strong position towards numerical approaches in law and more specifically proportionality. There is a paper by Sobek and Montag (2018) where the numerical approach to rights is considered. However it is not clear how big the body of literature in this area is.

2.7 Reflection on Research Methodology

This work brings together several strands of research (weighted formal reasoning, blockchain, multi-agent decision-making), which hold great potential for a wide range of applications. Some of these have been briefly outlined, but we believe that the work would benefit from a more detailed description. Some application areas are: politics, neural networks, reinforcement learning, formal argumentation, game theory, and consensus systems(blockchain). These areas all involve multi-agent systems decision making using various methods to represent reasoning. To see this, we may consider what can be understood as an agent. In principle, an agent needs to have some degree of agency, which tends to be understood as the capacity to act. The neuron may then be understood to have some degree of agency. A person is understood to have agency. A collective of people can be ascribed agency. Also, when a group of agents performs an action, this may be explicitly a group action, though may also be an implicit group action that emerges from the individual actions of the agents, i.e. without explicitly determining a collective action.

The question regarding the agents and their actions is what reasoning do they use to make decisions. And in the multi-agent setting there is also the question of what kind of reasoning the multi-agent system uses to reason with each other. The approach taken here is that we reason using so-called reasons and that we ascribe weight to those reasons to reflect how important they are in comparison to other reasons with which they appear together. The weight may be derived from many things. We may understand money as a proxy for weight; however, this does not always work. However, if you want to buy something, you better ascribe the same amount of weight to having the money and the cost of the object you would like to buy.

In general, we will thus discuss relations that consider a particular input. That input will be used to vie for a particular output. If the input is sufficient for, or permits the output, then the pair is part of the relation. If not, then the input-output pair is not part of the relation.

$$D : I \times O \to \{0, 1\}$$

When each input only has one output element in the relation, we tend to call the relation a function. We thus study these relations for agents and multi-agent systems. And therefore, the named fields come together. Either they relate to this like politics, where the different perspectives are merged, or they model it in a particular way, for example, formal argumentation. Blockchain also cares about this, considering multiple potential next states(block) suggested by different agents, and using it to determine which state is the next state. It turns out that we can simulate or implement – if you will

– such relations with different approaches. Furthermore, if we model it in a particular way it has consequences for the types of way we can combine the models in the multi-agent case.

The initial motivation of this work was to study and contribute to trustworthy AI by developing hybrid sub-symbolic symbolic systems so that AI output can be better understood. Using such a system the hope is that sub-symbolic systems can benefit from the transparency benefits of symbolic systems thereby improving explainability. This then leads to a better legal basis for legal disputes involving AI systems.

Before letting you go, I would like to give an example to simplify understanding how the work ties together. The simplest way to understand this is through the idea of triage. In triage, multiple factors are given weight based on how important people consider it. Instead of directly voting for conclusions, we can thus give weight to particular properties and then consider how much weight each outcome has. People can construct this weight assignment together or separately and then add the weights into one system. This weight assignment should be done preemptively, then when a decision moment arrives, the weight system is used to attribute weights to the elements relevant taking the context into account. The idea of triage here is that by understanding how important something is or how attached we are to things; we can understand what we will let go of first in case it is needed. In other words if we need to choose between things that we can not really choose between, we can understand what we can let go of before the other by evaluating the weight we attach to the elements involved in the choices. The movie IRobot presents an example of this. In a car accident, a robot needs to make a split second decision about saving two different people and it cannot save both. It decides to save the person with the highest chance of survival. However, the human saved made a different assessment, he accepted the probabilistic assessment, however, thought the other person –a young girl– had the right on having a chance at surviving over him. The robot rejected the input from the human and made its own decisions. However in hindsight we may thing about whether the robot should have allowed the human's input into the decision process with more weight. Another more painful example that I hesitated to write here is the assessment that occurred in hospitals during COVID. During this time triage was used to decide who would receive treatment from the hospital. This was based on multiple factors that were given weight among which the chance of recovery but also factors such as the quality of life after said recovery.

Enjoy reading!

Chapter 3

Balancing Weighted Reasons

This chapter examines the concept of balancing weighted reasons and applies the principle-based approach. It develops the dual-scale as a foundational detachment function, distinguishing between justifying weights and requiring weights. Then, using principles that constrain the weight function, the single-scale and uniform-scale are specified. The maximum function is used as a benchmark detachment function. Nine normative principles (Groundedness, Anonymity, Unanimity, Polarity Monotony, Union Monotony, Sensitivity, Input Symmetry, No Dilemmas, One Permission) are rendered and applied in a principle-based analysis to evaluate different detachment functions. The results show that dual-scale detachment offers greater expressiveness, capturing dilemmas and other permission states, while relaxing constraints such as Unanimity. The chapter establishes the theoretical foundation for the implementation of normative reasoning in practical systems.

3.1 Introduction

This chapter concerns a principle-based analysis of the dual-scale approach to balancing weighted reasons. The dual-scale approach is a relatively new approach to balancing weighted reasons. The ideas underlying both the dual-scale approach and the balancing of the weighted reasons that support it have been presented in articles and, more recently, in a book. This idea of balancing weighted reasons follows from other work on reasons. In particular, the book "Reasons as Defaults" Horty (2012) is a starting point for this line of research. It combines the study of reasons with the more established study of defaults and treats reasons as the foundational notion of normative reasoning. The idea is further developed in the book "Reasons First" Schroeder (2021a) and in the article "The Fundamental Reason for Reasons Fundamentalism" Schroeder (2021b). Other works also outline foundation of reasons fundamentalism, such as the article "Reasons and Rationality" Dancy (2009). Here, the connection between reasons and oughts is highlighted. In short, the benefit of reasons fundamentalism is that it simplifies normative theories to be about reasons that then lead to oughts. There is a substantial body of work in this area that considers the relation between reasons and oughts, as well as how reasons lead to oughts. For example, we might consider that what one ought to do is, in general, what is supported by the strongest reason. Alternatively, some reasons may be regarded as more important than others in certain situations.

The study of oughts dates back further than the modern version of the reasons project (which explains obligations in terms of reasons). Historically, the study of oughts is often called deontic logic or normative systems and is often considered to have begun with von Wright's(1951) work on deontic logic. This logic describes the concepts of obligation, permission, and prohibition. The field has grown considerably since that first article, as the handbooks show (Chopra et al., 2018; D. Gabbay et al., 2013). But the study of these concepts goes back even further, as do the concepts of reasons and reasoning. The field of normative reasoning is important for understanding agent and group behavior. Agents here include artificially intelligent agents and humans but can also be extended to other entities with varying degrees of autonomy; some even consider a light switch or atoms as being examples of agents. The concept also extends to organizations of people, such as companies and departments within companies, to countries, to the law and social norms. For artificial intelligence agents, this is important, as they must know what is permitted and what is not. If the AI considers the act of killing permitted, then it may lead to it killing living beings. Therefore, the choices made on to implement how permission works in turn determine how certain premises lead to consequent permissions. Some designs have been shown to cause issues with the resulting permissions. One of the more famous deontic logic paradoxes is Chisholm's paradox (Chisholm, 1963; Parent & Van der Torre, 2018).

This chapter looks at a new approach called detachment functions (Knoks & Van der Torre, 2023). Key to these systems is the idea of detachment taken from logic (Logic, 2006). More specifically, it refers to *modus ponens*, commonly known as the rule of implication. Implication works such that if we have A, and $A \rightarrow B$, then we can derive B. In this way, the detachment function produces a mapping that determines the deontic status of options based on the specified reasons. In other words, it determines whether a particular action is permitted, obligatory or forbidden given the reasons under consideration. However, as stated earlier, there are many views about how reasons make optional actions permissible. The detachment functions are therefore designed to be implemented using a variety of approaches, enabling comparison of the different methods.

For this chapter, the new dual-scale approach will be implemented as a detachment function. Then, other approaches to reason-based detachment will be implemented in the same system for comparison. Although multiple approaches will be introduced, the comparison focuses on: dual-scale, single-scale, Uniform weights, and Maximum weights. The different detachment function implementations are compared using multiple principles through a principle-based analysis.

Tucker's(2025) book introduces a dual weight approach to the reason-based balancing approach. Dual weights comprise two components: the justifying weight and the requiring weight. The justifying weight makes particular actions or options more permissible; a reason's justifying weight justifies choosing that option as an action. The requiring weight of a reason, on the other hand, makes an alternative option impermissible, thereby making the option that the reason supports a requirement. It makes the option a requirement because, if alternative options are not permitted, the action becomes mandatory, i.e., an obligation.

Figure 3.1: The single-scale model

In the past, the single-scale model (figure 3.1) was used to balance reasons to determine the deontic status of options. Here, options are actions, and reasons speak for or against them. The single-scale model is a balance scale in which, on one pan, the reasons for one option are placed, and on the other pan, the reasons favoring the alternative option are placed. The balance scale indicates which option should be chosen. Since it is a single-scale model, the weights of the reasons favoring the first option are compared with those favoring the second option. Furthermore, the single-scale has three outcomes: either the two scales have equal weight, or one is heavier than the other. If they are of equal weight, neither has an advantage over the other, and either is a permissible option; if one has more weight than the other, then that option is obligatory. Since the option carries more weight, there is no reason to choose the alternative over it. One of the main observations made in "The Weight of Reason" by Tucker (2025) is that a reason has two weights: the Justifying weight and the Requiring weight. In the single-scale model, these weights can both be identified; however, they are always equal, a condition known as the **single proportion assumption**. One way to build intuition for these types of weights is to consider a scenario in which you vote on which restaurant to have dinner at with friends. Not only can you vote to go to one restaurant, but you can also vote against going to a particular restaurant. The single-proportion assumption is that a vote for one restaurant is, at the same time, a vote with the same weight against the other restaurants. The dual weights theory requires an additional balance scale; therefore, when comparing two options, there are two balance scales (or a dual scale) and four pans. This model is called dual-scale (figure 3.2). In one balance scale, the justifying weight of an option is pitted against the requiring weight of the alternative. This balance scale determines whether the first option is permitted. The second balance scale determines whether the first option is in that comparison an obligation by considering whether the alternative option is permitted. This is done by comparing the justifying weight of the alternative option with the requiring weight of the first option. Together, these scales determine the deontic status of the options using pairwise comparisons. Therefore, if there are more than two options, multiple pairwise comparisons must be performed.

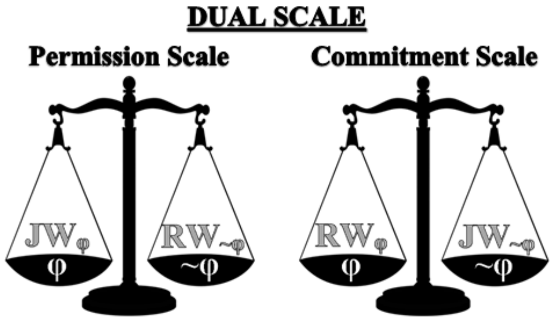


Figure 3.2: A dual-scale model

In this chapter, the concepts introduced in the book “The Weight of Reason” by Tucker(2025) are expressed in mathematics as a detachment function. Afterward, principles are introduced and applied in a principle-based analysis of the definitions of the balance scale models. To perform the principle-based analysis first, the concepts of “The Weight of Reason” are specified in terms of the detachment functions. Detachment functions, or systems, as they were originally named, were introduced by Knoks and Van der Torre(2023). Detachment functions focus on detachment in logic, a superclass of inference patterns. The idea is that these functions contain many types of inference patterns, including the ones considered in the reason-based approach to deontic logic. More formally, a detachment function specifies a value for the relationship between a precondition and a postcondition. This value is then interpreted as the deontic state of the postcondition given the precondition. So, if the detachment function gives a value that says that the postcondition is permitted given the precondition, then the detachment function assignes the value “1”’to the postcondition in other words permits the postcondition given that the precondition is satisfied.

The methodology section provides the formal definitions of dual-scale balancing of reasons as a detachment function. Further definitions are also provided, including other detachment types. Afterward, the Dynamic Scale is introduced and specified; this scale implements tournaments among the possible options. Lastly, the principles are introduced: three sets, namely context principles, detachment principles, and weight principles. The results section then discusses the performance of the detachment functions on the principles. Which detachment functions satisfy which principles? That section is followed by the Discussion, which presents the results, delves deeper into the concepts introduced, and addresses other relevant topics not previously discussed. Lastly, the chapter concludes with a section that summarizes the work and lists the main findings.

3.2 Background

In Tucker's(2025) book, multiple components are introduced that are relevant to weighing reasons correctly. This section introduces all these concepts and discusses them in relation to their specification as a detachment function. This is in preparation for the next section in which the formalism is specified in first-order logic as a detachment function.

3.2.1 Context

The general idea of a context is that it surrounds the crux of a matter. The way one looks at the crux can be influenced by the context surrounding it. However, if we can distinguish one context from another, then we can determine exactly where the two differ. To do that, we should be able to formulate the structure of the context and make concrete statements about the differences between two contexts. The notion of a context was used in an article introducing detachment functions by Knoks and Van der Torre(2023), which evaluate options (understood as actions) in the context of reasons. In this case, they explicitly define a context as a set of reasons together with an option at the center of the matter.

The article by Streit et al.(2023) shows that it is possible to take a structural approach to the context. However, that approach is not fully aligned with the dual-weight approach. Later, contrastivism will be introduced as a property relevant to the dual-scale approach specified by Tucker (2025). In the article by Streit et al.(2023), contrastivism is not taken into account. On the subject of contrastivism, it may be worth researching contrastivism more deeply; maybe contrastivism is an emergent property of reasons with the same ground pushing in opposing directions. The Ph.D. dissertation of Streit(2024) goes beyond the article cited above and shows that dual weights are feasible in the framework of weighted normative graphs. This is accomplished by allowing each reason to have a role, either positive or negative, that is descriptively sufficient to express justifying and requiring weights.

3.2.2 Options

The simplest conception of an option is to understand it as an action. Then, we have a set of elements that refer to options that can be selected at a decision point. However, one can easily see that multiple options will have a similar effect when considering them on particular finite-time scales; for example, two paths to the supermarket. Also, there may be domains with clear, discrete, finite sets of options, just as there are domains with infinite options. An example of a finite domain is the case of taking one of the paths at a finite junction, i.e., a four-way street crossing. An example of infinite options is setting a course in an airplane; there are, in theory, infinite directions available, but in practice, pilots will not consider all these options.

If one wants to be more precise, there are two ways. The first is to define a highly selective, engineered finite set of options. The other is to specify an algebra that describes the options in greater detail. However, obtaining a set of all options may be difficult when the domain under consideration is infinite. This is a research field in its own right. For this work, it is preferable to consider a finite set of discrete elements, each of which refers to or symbolizes a distinct option, and leaving the infinite case out of scope.

Maximalism. Maximalism, as used by Tucker (2025, p 43, 117), is the idea that the options should be maximally consistent. This means that if one has a set of maximally consistent options, then one can not choose more than one option. In addition, the option cannot be extended in any way so that it contains more elements (Tucker, 2025). This refers to the notion of maximally descriptive; for example, if one can choose between "A" and "B" and then after choosing "A" one has the choice between "C" and "D", then the options are: "A" then "C", "A" then "D", and "B". Here, "A" cannot be chosen by itself, since it is not maximal in the domain of possible options.

3.2.3 Grounds

Grounds are elements that can be considered as reasons for choosing a particular option. The importance of a ground for an option is expressed as the weight of that reason. However, a ground may be relevant when an option is compared to one particular alternative and irrelevant when the same option is compared to a different alternative. It could be that the ground loses relevance for that comparison, or that the ground's weight is canceled because the ground is a relevant reason with equal weight for the alternative option. As a result, a ground is typically understood as a reason for only one of two options in a comparison.

Given this theory's origins in ethics, the grounds are conceptualized as ethical. For example, the value of a life is a ground for saving a life. The grounds are reasons arising from the ethical theory. They are not subjective perceptions but rather objective reasons for particular options. However, if this theory is applied to a different domain, one can make a choice about what exactly constitutes a ground. For example, in fund management, one could choose the closing price of a stock, or opt instead to use a metric such as the relative strength index as a ground, $RSI > 70$, for investing in a stock. In general a ground is conceived of as a fact. Similarly as options we will understand them as unique elements however alternatively one may develop a syntax involving disjunction and conjunction to give grounds more structure. For us it is merely relevant to be able to distinguish one ground from another even if the difference is extremely small or debatable.

3.2.4 Reasons

The idea of a reason comes from sentences like "being tired is a reason to go to sleep". A reason, then, is the relation of a ground to an option. The ground is a reason for choosing that particular option. Normative reasons occur when the deontic status of an option is considered. That each life has value is a reason for killing to be impermissible. That each life has value is a reason to permit saving someone's life.

Reasons for and against. Sometimes, a reason is given as a basis that favors a particular option. For example, liking a particular cuisine is a reason to go to a restaurant of that cuisine. However, a ground can also be a reason not to choose a particular option. These ideas are about whether reasons can be for and against options, which can be expressed in the syntax by adding a negation to the language. It also requires additional work on the detachment functions, since they do not handle negation. A reason against could then be expressed as $(g, \neg o)$. This reason means that the ground g is a reason for choosing alternatives to option o.

3.2.5 Weight

The weight of a reason is given by a function specifying, for each reason, the weight in relation to the context and pairwise comparison. Following the theory of Tucker (2025), the function yields two weights: a justifying weight and a requiring weight. These weights can vary across contexts, and can change when a different alternative is used in the pairwise comparison. In principle, the weights can be any non-negative number.

However, it may be more interesting when the weights are subject to additional constraints, for example, when a part of the weight of a reason is immutable. This will constrain the set of possible weight functions, providing more to work with in uncertain circumstances. *For example, it may be reasonable to base the weight of reasons on weighted grounds, where the weight of the ground is intrinsic to that ground, which then translates to the weight of reasons using a particular strength between the ground and the option for that particular context and pairwise comparison. For this work, this has not been implemented.* More important in understanding the weight of reasons are weight pluralism, contrastivism, and comparativism.

Weight pluralism. Reasons can have more than one weight: justifying, requiring, commending, etc. This yields pairwise comparisons among all options in which each option is compared with every other option. This particular way depends on the embeddings of the weights and on what the weights are connected to. The weights are then aggregated to evaluate the relative importance of the options.

Justifying weight is a weight that makes an option permissible. An example is having sufficient funds to purchase a product. Another example is that saving a life is a good thing.

Requiring weight is weight that makes an option required or obligatory, and so makes alternatives forbidden, since an option is obligatory if there are no alternatives. Making a promise obliges the promisor to keep that promise. The weight must be overcome by justifying the alternative options as being permissible.

Commending weight is weight that allows for comparison between permitted options. If two options are permitted, a choice must still be made. To break the tie, the commending weight may be used.

Contrastivism. When comparing two options, a function yields a set of reasons that relate to both options. When there are three options, the function generates three sets. Contrastivism is a statement about the content of these sets. More precisely, it states that a reason for one option need not occur in all comparisons. Therefore, for example, a reason in one comparison does not necessarily occur in comparison with a different alternative. This is because the option it supports is in contrast, in pairwise comparison, to particular options. Also, the weight can vary; in one comparison, the reason is mainly justifying, whereas in most other comparisons, it is mainly requiring.

Comparativism. In a similar vein, the idea of comparativism is that even when a reason supports both options in the comparison, only one can have weight. A reason is always such that it favors one option more than another; otherwise, it would not be a reason in that comparison. For a ground to be a reason for an option means that it has more weight for that option than the other option to which it is being compared. It may possibly favor both options, but one more than the other, so it is a reason for that option in the comparison. A reason can favor more than one option, but the reason represents the difference in terms of one option being better than the other in that particular regard. In addition, comparativism implies contrastivism (Tucker, 2025, pp 122).

This idea also yields a different version of a reason being canceled out — equal weight or greater weight for the alternative. The influence of a reason on the balance scale can be neutralized by counterweighing the weight of that reason with a different reason for the alternative. Therefore, if comparativism is dropped, the grounds can serve as reasons for both options in a pairwise comparison. If one strictly outweighs the other, the net weight of one reason is positive, and the other has no weight; comparativism remains implicit. However, some new cases may arise in which the ground is a reason for both and has sufficient weight to yield a net positive justifying weight for both options, making both options permissible.

The Shield. In (Tucker, 2025, pp 54), it is explained that Holism is a shield against additive aggregation, not a threat to it. Additive aggregation is the aggregation method used in the next section as per definition 4. This means that the method supports the idea that, when reasons are aggregated, their weights are added. When a ground is a reason for a particular option, it will have a particular weight.

If another option is brought forward that is supported by the ground twice, for example, a particular benefit is given twice, then the weight also changes. If Atomism is supported, one might say that the weight should be doubled; however, this is not always sensible. Holism offers a way out of this conundrum by allowing the weight to be contextually defined, without requiring a linear increase. In this way, Holism protects additive aggregation from complex problems that arise when one supports linear scaling of weights with respect to quantities in grounds. The definitions allow the expression of both holistic and atomistic views of the weights of reason.

3.2.6 Detachment

Detachment functions were introduced by Knoks and Van der Torre (2023), and were followed by two articles (de Wit, Knoks, & Van der Torre, 2025a; Knoks et al., 2024). The main purpose of the detachment function is to indicate what options are permitted and not permitted, given the reasons in the context. It does this by assigning a value to each comparison between two options. The output values of balancing indicate the balance between the two pans, i.e., whether one side is heavier or if they are equally heavy. We write $+$ for the heavier side, $-$ for the lighter side, and 0 when the pans are equally heavy. Therefore, a pairwise comparison has nine possible status assignments: $(+,+)$, $(+,0)$, $(+,-)$, $(0,+)$, $(0,0)$, $(0,-)$, $(-,+)$, $(-,0)$, $(-,-)$. The value $+$ is interpreted as permission, and the value $-$ is interpreted as impermissible or forbidden; when only one option is permitted, then that option is obligatory. The value 0 can have two interpretations; in Tucker's(2025) book, this value is interpreted as permission, although in the deontic logic literature, the notion of weak permission also exists. Since 0 is also the output when there are no reasons in the comparison, it makes sense to see 0 as a weak permission; nevertheless, either way, 0 also counts as a permission.

Some of the nine deontic statuses are not accessible by some detachment functions. For example, if a detachment function implements single proportion weights, then only three deontic states possible are: $(+,-)$, $(0,0)$, $(-,+)$. In Tucker's(2025) book, the dynamic scale is explained. This is a dual-scale with a tournament of pairwise comparisons. This system is used to simplify the valuations to only $-$ and $+$ and, above all, to move from pairwise evaluations to all-things-considered evaluations.

Deontic logic has ongoing discussion about what all the possible deontic statuses are and how they relate. For example: regarding the idea of weak permission, but also some hold that obligation and permission are duals while others deny this. If more deontic statuses are possible, it is possible to change the set of possible deontic statuses. Then, in the final evaluation, whether an option is permitted or not, the evaluation may be extended to take more values, and possibly a different logic for lifting it into all-things-considered evaluations.

A notable deontic status is the **dilemma case** $(-, -)$, which arises when using dual-scale detachment. This occurs when single-proportion weights are dropped, and the output indicates that neither option is permitted when compared to the other. When there are more than two options, this is acceptable; however, if no option is permitted, then that can be a problem. If there are only two options, it would imply that nothing is permitted. Normally, in deontic logic, there is the axiom of seriality, which makes it so that there is always something that is permitted. If nothing is permitted, this would be very problematic; nevertheless, it is a good thing that the definitions are such that they can capture such occurrences. Then, the system can notify when a dilemma is reached. How such an input—one that outputs a dilemma—should be handled is a decision for the systems users that use this output and/or the makers of those systems. From an ethical perspective, it is debatable whether these cases actually exist or not (Tucker, 2025, pp 76, 77). From a practical perspective, even if these cases do not exist, the output may still occur in implementations due to an incomplete or imprecise representation of the problem. The weights, options, grounds, or reasons may be missing or incorrect. Alternatively, grounds or other elements may be misinterpreted due to false information. For the formalization it does not matter whether dilemmas exist, or do not exist, since dual scale can describe both cases.

Detachment in Logic. The term detachment is somewhat loaded. Of course, this word is used in relation to meditative practices in which people understand detachment, as not being attached in a way that makes object on is attached to have some degree of control over them. More importantly for this dissertation, detachment also occurs in logic. Detachment is often called factual detachment, but also deontic detachment in deontic logic. Detachment can also be understood as *modus ponens* (Makinson, 2005, pp 17), which is also called material implication, or simply implication. This is often written in logic as $A \rightarrow B$, which reads as A implies B, which means that given the precondition A, the postcondition B is implied or detached. In deontic logic, deontic detachment means that the implication is a deontic formula, $O(A \rightarrow B)$, which states that it should be the case that A implies

B, so when A is the case or $O(A)$ is the case then an obligation for B is derived, i.e. $O(B)$.

The detachment relation in this work relates contexts to options and attaches values to these relations. In the work of Knoks and Van der Torre (2023), their definitions of detachment systems are shown to be expressible in standard logic notation. The value $+$ can be understood as indicating that the context is a sufficient precondition for detaching the permissibility of the particular option as a postcondition. However, since $+$ means permission, this is a sufficient precondition for the postcondition and does not necessitate the option itself.

Jørgenson's dilemma. Detachment functions bypass to some degree the problems of Jørgenson's dilemma (Jørgensen, 1937). Jørgenson's dilemma concerns the assignment of truth values to deontic formulas. The dilemma is that imperatives cannot have truth values. It is debatable whether detachment functions assign truth values to imperatives. Although the assigned values, i.e. $\{+, 0, -\}$, are not introduced as truth values, they might be interpreted given additional surrounding definitions. One could map $+$ to "the option is permitted" being true and $-$ to "the option is permitted" being false, i.e., "the option is not permitted". An imperative may then be understood as a restriction on what is permitted and what is not permitted. If your boss issues an imperative, you are required to adhere to new deontic statuses for specific options in specific contexts.

Deontic Paradoxes

Deontic logic, over the years, has found multiple paradoxes that show that particular deontic logics have particular problems. Since detachment functions assign deontic statuses, they can be tested on deontic logic paradoxes. Below is merely a brief reflection intended for those familiar with these problems. The other paradoxes can easily be found in the work of Meyer et al.(2001), in the deontic logic entry of sep.com [1] and in the handbooks (D. Gabbay et al., 2013, 2021).

Ross's paradox. You ought to mail the letter or burn it (Ross, 1944). This paradox, known as Ross's paradox, concerns deriving oughts to actions that are obviously forbidden. The paradox starts with the obligation to mail the letter. However, in logic, you can use disjunction to expand the formula to

[1] https://plato.stanford.edu/entries/logic-deontic/#ChisPuzzSDL

include that you ought to burn the letter. Due to how disjunction (or) works, the truth of "you ought to mail the letter" translates to the truth of the disjunction. This sentence is paradoxical because, in the axiomatizations of deontic logic at the time, it permitted burning the letter, even though it was clear to any person that that was not intended. Part of the problem is that burning the letter lacks a deontic status, and its deontic status is inferred from what is given. If the deontic status of burning the letter had been specified, it would have been impossible to derive the permission from the disjunction.

As will become clear, all options have a deontic status by default when using the so-called dynamic scale, since even if there are no reasons favoring the option or its alternatives, pairwise comparisons yield the value 0, leaving the option with output $+$ at the dynamic detachment level. However, this means that if you add the option of burning the letter, it would possibly receive no reasons while mailing the letter would. Then, if mailing the letter is required, there is a reason for mailing that applies required weight opposing burning the letter, so burning the letter has the value of $-$ compared to mailing it. If there are no reasons because the system did not account for this option, then burning the letter might receive a value of 0 in each comparison. Therefore, it may be advisable to include 0 as a possible output of the dynamic scale, or to let 0-valuations evaluate to the value of $-$ at the dynamic level.

Free choice permission. The free-choice permission problem also applies to disjunctions of permissions. In the free-choice permission problem, one is permitted to have coffee or tea. The problem is in interpreting this permission. Are you allowed to have only one, both, or none? There are different perspectives on this dilemma (Kamp, 2013).

The detachment functions define the permission for each option. Here, multiple options can be distinguished: o_1 nothing, o_2 coffee, o_3 tea, o_4 coffee and tea. The detachment function will assign each a deontic status. One could set the system so that it is permitted to choose either option o_2 or o_3, in which case it would be permitted to choose coffee or tea. However, one could also include o_1 and o_4 as permissible options. It depends on the weights specified.

3.2.7 The Complexity Cost of Balancing

Using a dual-scale instead of a single-scale means twice as many balancing pans are used. In addition, this doubles the number of weights that reasons have. The dynamic scale does not need to consider more outputs; it only exists so that the outputs that are there have more possible values, that is, among others, $(0, +)$, $(0, -)$. Pairwise comparisons increase the number of weights, since for each reason (g, o) the reason has two weights per alternative o', o'' etc. (g, o) may have two different weights.

If there are m grounds, then for any two options, if a ground can only be a reason for one option, the two options will have at most m reasons; otherwise, $2m$ reasons at most. The weight function assigns two weights to each reason, so the total number of weights per comparison is at most $2m$. For n options, there will be $\frac{n*(n-1)}{2}$ comparisons, since there are n options that can be the first, and then there are $n - 1$ left as an alternative, divided by 2 for doubles.

3.2.8 Weight of Reason book examples

Tucker's book uses multiple examples to explain the problems of weighing reasons and why the dual-scale approach is needed. They all highlight either a problem or a puzzling aspect of the balance-based approach.

Dorsey's car case. This problem concerns the transitivity of justification and is a version of the justification intransitivity paradox.

Tucker writes: ***Justification Transitivity:*** *If R1 makes permissible acting against R2 and R2 makes permissible acting against R3, then R1 makes permissible acting against R3*

He states the example in the following way:

> Suppose that purchasing a car would cost $17,000, and it would allow you to accept a more satisfying position and participate in your preferred leisure activities. Intuitively, it is permissible to buy the car for these self-interested reasons, as long as you've already made sufficient efforts to help others and give to charity. Yet, instead of buying the car, you also have the option to donate that $17,000 to the Malaria Consortium. GiveWell (2020) estimates that every $3,373 donation to that foundation's chemo prevention pro-

gram saves a child from death. $3,373 x 5 is approximately $17,000, the cost of the car. Hence, it is morally permissible to get a car rather than save the lives of five people

Although this is already remarkable from an ethical perspective, the paradox emerges in the next paragraph.

> The lives of five people nonetheless have significant weight in their own right, as it is permissible to beat up an innocent person, Jerry, to save the lives of five people. So far, so good. The paradox emerges when we notice that it is impermissible to beat up Jerry to get a car.

Here "(5lives) the saving of lives", "(car) the buying of a car", and "(respect) the beating up of Jerry" stand in a relation to each other. One can permit another; however, they show that transitivity does not hold, since beating up Jerry is not sufficient reason to get a car, at least ethically.

> **Car** Justifies Against **5Lives**: It is morally permissible to get a car rather than save the lives of 5 people.
> **5Lives** Justifies Against **Respect**: It is morally permissible to save 5 people's lives even if you have to beat up an innocent person, Jerry, to do it.
> **Justification Transitivity**: If Car makes permissible acting against 5Lives and 5Lives makes permissible acting against Respect, then Car makes permissible acting against Respect. And yet:
> **Respect** is Required Against **Car**: It is not morally permissible to beat up an innocent person, Jerry, to get a car.
>
> Dorsey's Car Case is the conjunction of the three deontic verdicts, i.e., Car Justifies Against 5Lives, 5Lives Justifies Against Respect, and Respect is Required Against Car. If these three verdicts are true—as I and others assume them to be—Justification Transitivity and Simple Weight Monism are false.

The normative significance of small improvements. This puzzle arises when comparing multiple options, each of which has an alternative that offers a slight improvement at no additional cost. The problem suggests that the weights of reasons can be incommensurable, leading to outcomes that cannot be found. The following example is put forth in Tucker's book:

> To explain the normative significance of small improvements, we must explain both the *pro tanto* maximization of certain reasons and the stability of moral options involving those reasons. The puzzle begins with two options that involve acting on different morally relevant considerations: altruistically saving five lives (A) and respecting Jerry's right not to be beaten up (R). (Again, to prevent confusion about the nomenclature, 'A' and 'R' refer to options. 5Lives is the reason to choose A, and Respect is the reason to choose R.) Then consider modestly improved versions of these considerations, such as saving those five lives and an additional life (A+) and respecting both Jerry's right not to be beaten up and his right not to be insulted (R+). Pro tanto maximization is illustrated by the requirements to choose A+ over A (saving 6 rather than 5 lives) and R+ over R (respecting both rights instead of just one). When we aren't trading morally relevant considerations off against one another, small improvements tend to generate requirements.
>
> Nonetheless, we have the moral option to choose between A and R, between saving 5 or respecting Jerry's right not to be beaten up. The stability of moral options is illustrated by the moral option persisting after we improve either of the reasons. The moral option persists if we have to choose between A and R+, and it persists if we have to choose between A+ and R. When we are trading morally relevant considerations off against one another (e.g., altruism vs respect), small improvements tend not to generate requirements. They tend to illustrate the stability of moral options. Whether the improvement, A+, and R+ generate a requirement seems to depend on whether we are trading off morally relevant considerations. This is puzzling, but it's a puzzle that we've already solved.

The puzzle shows that the weights associated with options A and R are not known exactly. Therefore, they cannot be compared so that we know which one is heavier. However, they are of a comparable size, so it is difficult to choose one over the other. The weights added to both $A+$ and $R+$ are comparable, leaving the initial problem intact; i.e., $w(A+) - w(A) = w(R+) - w(R)$. Thus, $A+$ is an obligation relative to A, just as $R+$ is an obligation relative to R. However, whether A or R

should be chosen remains an open problem, corresponding to the comparison between $A+$ and $R+$.

Kamm's café or the Kid case. This problem concerns comparativism and shows that pairwise comparisons allow the modeling of the details of some circumstances more accurately.

> Consider Kamm's Café or Kid Case, which we will explore more deeply in chapter 9. There is a child stuck in a burning building. Your options are Bystander (hang around and watch the events unfold), Save1 (save the child and get severely burned in the process), and Café (keep your promise to meet your friend at the café).
>
> When Café competes with Save1, the child's life is a reason against Café because she dies in Café but not Save1. Comparativism entails, then, that it is a reason for Save1 in that pairwise competition. This result seems plausible. It can seem arbitrary whether we talk about the child's life as a reason for Save1 or as a reason against Café. It is just two different ways to talk about the same reason (§5.1.2).
>
> In contrast, when Café competes with Bystander, the child dies either way. Comparativism holds, then, that the child's life is neither a reason against Café nor a reason for Bystander. The child's death in Café is not a way that Café is worse than Bystander, nor a way that Bystander is better than Café.
>
> Kamm's Café or Kid case illustrates why comparativism answers WhichAlt at the level of individual competitions. The child's death is a reason against Café in the pairwise competition with Save1, but not the pairwise competition with Bystander. Per Comparativism's Answer to WhichAlt, we should expect the child's death to be a reason for Save1 (in Café vs Save1) but not a reason for Bystander (in Café vs Bystander).

In a further study of this case, Save2 is added. In this case, there is also the option to save 2 people. In that circumstance, it is obligatory to Save2 if you commit to Save1, since the detriments are the same; however, you save an additional life at no cost.

The all-or-nothing ranking reversal. This puzzle is based on Kamm's cafe puzzle and includes the Save2 option. In that case, you have to save all children or none. It's an all-or-nothing situation, even though it still seems good to save just one child. The following weights are suggested by Tucker:

Option	**JW**	**RW**
Bystander	500	0
Save1	100	100
Save2	200	200

Table 3.1: All or Nothing Weights (Tucker, 2025)

The two-option case only considers Bystander and Save1; in that case, Tucker evaluates in the following way:

- Save1 is permissible, because 100 justifying weight for Save 1 is greater than 0, requiring weight for Bystander.
- Bystander is permissible, because 500 justifying weight for Bystander is greater than 100 requiring weight for Save1

In the three-option case, when Save2 is also considered Tuckers evaluation changes to the following:

- Save2 is permissible, because 200 justifying weight for Save 2 is greater than both 100 requiring weight for Save1 and 0 requiring weight for Bystander.
- Bystander is permissible because 500 justifying weight for Bystander is greater than both 200 requiring weight for Save2 and 100 requiring weight for Save1.
- Save1 is impermissible. While 100 justifying weight for Save1 is greater than 0 requiring weight for Bystander, it is less than 200 requiring weight for Save2.

These examples have not yet been studied in this dissertation using the first order logic rendering of dual scale. However, it would be a viable extension to see if the methodology can shed the same or a different light on these puzzles. Tucker's answer to these puzzles is a dynamic dual scale rather than a single scale. In what follows, the dual and dynamic scales are developed so that the dual scale can be subjected to a principle-based analysis. Given that the dual scale is modeled, these examples should yield the same results when rendered using the function discussed in the next section.

3.3 Reason-Based Detachment Functions

In this section, the reason-based detachment methodology (de Wit, Knoks, & Van der Torre, 2025a; Knoks & Van der Torre, 2023; Knoks et al., 2024) is introduced. However, their methodology is modified, and the system still supports the detachment types studied in their articles, while also accounting for other literature. In particular, the given specification follows de Wit, Knoks, and Van der Torre (2025a) and aims to model Tucker's approach to weighted reasons, as expressed in his book "The Weight of Reasons" Tucker (2025).

3.3.1 Reason-Based Detachment

As mentioned, the dual-scale model of weighing reasons is only one approach to assigning deontic status to options in contexts. Although the formalization focuses on this model, it should be open to the other approaches and respect their effects. Because of this, the formalization might initially seem more complicated than necessary; however, it is (slightly) more complicated for good reasons.

The dual-scale weighs a set of reasons supporting one option, o_1, against a set of reasons supporting another, o_2. So, one might naturally think that the set of reasons that are relevant for the choice between o_1 and o_2 can be defined as the set of reasons supporting o_1 and the set of reasons supporting o_2, i.e., as $R(o_1, o_2) = R(o_1) \cup R(o_2)$. However, Tucker emphasizes that this is not always the case and, therefore, the more general two-place function R is used. In Tucker's view, it may well happen that something is a reason supporting option o_1 when o_1 is being compared to option o_2 but is no longer a reason supporting option o_1 when o_1 is being compared to, for example, option o_3. This is known as *contrastivism about reasons*.

In addition, in Tucker's view, the reasons do not exhaust the set of considerations that can be normatively relevant to the assignment of deontic status to options. Following a long tradition in philosophy, he allows "disablers" and "modifiers" to affect the reasons that support options before they are compared on the balance scales: these can, respectively, nullify the weight of reasons or alter their weight. (These contextual factors are also used as a "shield" against standard purported counterexamples to additive aggregation of weights of reasons.) Disablers or modifiers will not be explicitly modeled. Instead, the specification focuses on reasons after the effects of disablers and modifiers have taken place.

With these preliminaries out of the way, the framework specification can be given. Reasons are understood as a bipartite graph defined over grounds (or facts) and options (against a background context). For now, grounds, options, and contexts are left abstract, as this is all that is needed to carry out the principle-based analysis. It is merely necessary to be able to point to a particular one so that it is certain that either the ground, option, or context that is considered does not change. However, in Section 3.6, future work is discussed in which grounds can be understood as propositional formulas with logical connectives, options as complex actions built from atomic actions, and contexts as sets of disabling and modifying reasons.

Definition 1 (Reasons)**.** *Let C, O and G be three sets called* contexts*,* options *, and* grounds*, respectively. Let $\mathcal{P}(G \times O)$ be the power set of $G \times O$. A* reasons function $R : O \times O \to \mathcal{P}(G \times O)$ *associates each pair of options with a set of ground-option pairs, called the* set of reasons *(pertinent to the given pair of options). Let SR be the set of all such functions R. Refer to the elements of C, O, and G using c, o, and g with subscripts.*

Example 1 (To have coffee or not to have coffee)**.** *To give an example of the intuition behind the reason-based approach, consider the option of drinking a beverage, maybe a cup of coffee. Such an action is an option. A ground, then, is a fact or something similar that functions as a reason for choosing that option. If you are thirsty, this can be a reason for drinking something, such as a cup of coffee. However, you may need to pay for the coffee but don't have sufficient money to hand, which is a reason for not drinking coffee. Context can alter the interpretation of the reasons present. For example, there may be contexts in which it is too late to drink coffee. Even though the time of day may always be a reason for not drinking coffee too late, this may, in some contexts, be evaluated differently.*

$$G = \{\textit{time, thirsty, sufficient-money}\}$$

$$O = \{\textit{drink-coffee}\}$$

$$R(\textit{drink-coffee}, \neg\textit{drink-coffee}) = \{r_1 = (\textit{time}, \neg\textit{drink-coffee}),$$

$$r_2 = (\textit{thirsty}, \textit{drink-coffee}), r_3 = (\neg\textit{sufficient-money}, \neg\textit{drink-coffee})\}$$

Using these concepts, a relation between preconditions and postconditions can be instantiated. Here, we refer to them as detachment functions, in accordance with the literature. We connect reasons in a context to an option and assign a value. Here, pairwise comparisons are used as described earlier.

Definition 2 (Detachment Functions)**.** *A detachment function $D : SR \times O \times O \times \mathcal{P}(G \times O) \times C \rightarrow V \times V$ assigns to each pair of options ($o_1, o_2 \in O$) a pair of scale values $v_1, v_2 \in V (= \{+, -, 0\})$ by considering a subset of reasons that are relevant to these options ($RR \subseteq R(o_1, o_2)$), against the backdrop of a context ($c \in C$).*

$$D(R, o_1, o_2, RR, c) = (v_1, v_2)$$

Example 2 (To have coffee or not to have coffee)**.** *Consider the case of drinking coffee in the scenario described earlier.*

$$D(R, o_1, \neg o_1, \{r_1, r_2\}, c_1) = (+, -)$$

$$D(R, o_1, \neg o_1, \{r_1, r_2\}, c_2) = (-, +)$$

In context c_1, drinking coffee has the value $+$ and not drinking has the value $-$. This can be interpreted as drinking coffee is a good idea, and not drinking coffee is a bad idea. In context c_2, this has changed, so in that context, drinking coffee is a bad idea, while not drinking coffee is a good idea.

3.3.2 Weighted Reasons

The specification of the weigh*t*ing and weighing functions follows the work of Tucker (2025). Note that these are partial functions in the sense that they are specified for a limited set of reasons, namely those that are relevant for choices between two options.

The weighting of reasons—or the assignment of weights to reasons—is based on the following idea. In the comparison between options o_1 and o_2, a reason for o_1 is also a reason against o_2. However, the metaphorical weight or strength of the positive impact on o_1 may be different from the reverse impact on o_2. Therefore, in this work, the Tucker model is followed, and two weights are associated with each reason: a justifying weight and a requiring weight. (The weights are "embedded" or "final" because these are the weights that reasons have after the effects of the modifiers.) The only criterion that must be imposed on the set of weights assigned to reasons is that the addition is defined and that the weights can be compared using a binary relation $\geq$. In the following definition, real nonnegative numbers are used so that each pair of weights is comparable.

Definition 3 (Weighting (of Reasons))**.** *Let $\mathbb{R}_{\geq 0}$ be the nonnegative reals. A* weight function $w : SR \times O \times O \times C \times (G \times O) \to \mathbb{R}_{\geq 0} \times \mathbb{R}_{\geq 0}$ *assigns to each reason a function, a pair of options o_1 and o_2, context c, and two weights w_1 and w_2 for every reason $(g, o) = r \in R(o_1, o_2)$ pertinent to the options o_1 and o_2, with the proviso that at least one of w_1 or w_2 is positive. The first weight is called the* justifying weight*, and the second weight is called the* requiring weight*: if $jw(R, o_1, o_2, c, r) = w_1$ and $rw(R, o_1, o_2, c, r) = w_2$ only if $w(R, o_1, o_2, c, r) = (w_1, w_2)$.*

When the choice options o_1 and o_2, the set of relevant reasons $R(o_1, o_2)$, and the context c, are clear from the surrounding text, one can also write $w(r) = (w_1, w_2)$, $jw(r) = w_1$, and $rw(r) = w_2$.

Example 3 (To have coffee or not to have coffee)**.** *Through weight, the differences in outcomes across contexts are explained. For example, in context c_1, the weight of the reason for option 1 may be twice as heavy as the weight of the reason for option 2 (negation of option 1). Whereas in context c_2, it is the other way round.*

$$2 * w(R, o_1, \neg o_1, c_1, r_1) = w(R, o_1, \neg o_1, c_1, r_2) = (2, 2)$$

$$w(R, o_1, \neg o_1, c_2, r_1) = 2 * w(R, o_1, \neg o_1, c_2, r_2) = (2, 2)$$

Because disablers may be present, not all reasons in $R(o_1, o_2)$ need to be in force when weighing options o_1 and o_2. This is why the *set of relevant (or non-disabled) reasons $RR \subseteq R(o_1, o_2)$* is defined. This set RR can be thought of as the reasons used to compare options o_1 and o_2 within the dual-scale model. Multiple reasons may be available in a particular context; however, only the weight of the reasons in RR is placed onto the pans. Many of the principles defined in Section 3.3.6 have to do with varying this set RR.

3.3.3 Balancing Operators

Multiple so-called balancing operators can be specified. Notably, we can use addition and other operations, such as the Max function or multiplication. The additive dual scale operator is considered the default for detachment; the other operators, when used, will be indexed on the detachment function.

According to the dual-scale model, an option is permissible if the sum of the justifying weights of reasons supporting this option is at least as great as the sum of the requiring weights of the reasons supporting the alternative option. This implies that both options can be permissible simultaneously, just as neither is permissible. An option is obligatory when it is permissible and the alternative option is not. This is captured by the next definition. Notice that the three (scale) values assigned to the options correspond to three possible positions on the metaphorical balance scale: the o_1 pan is lower than the o_2 pan; the o_1 pan is higher than the o_2 pan; and the pans are in balance.

Definition 4 (Additive Dual-Scale Weighing (of Options))**.** *Let* V *be the set* $\{+, -, 0\}$*, called* (scale) values. *An additive dual-scale weighing model is a* weighing function $D_w : SR \times O \times O \times \mathcal{P}(G \times O) \times C \rightarrow V \times V$ *that assigns to each pair of options* o_1 *and* o_2 *a subset of reasons* $RR \subseteq R(o_1, o_2)$ *that are relevant to this pair, a context* c*, a weight function* w*, and a pair of scale values* (v_1, v_2) *as follows:*

$$v_1 = \begin{cases} + & \textit{if } \Sigma_{x \in JW_{o_1}} x > \Sigma_{y \in RW_{o_2}} y \\ - & \textit{if } \Sigma_{x \in JW_{o_1}} x < \Sigma_{y \in RW_{o_2}} y \\ 0 & \textit{otherwise} \end{cases}$$

$$v_2 = \begin{cases} + & \textit{if } \Sigma_{x \in JW_{o_2}} x > \Sigma_{y \in RW_{o_1}} y \\ - & \textit{if } \Sigma_{x \in JW_{o_2}} x < \Sigma_{y \in RW_{o_1}} y \\ 0 & \textit{otherwise} \end{cases}$$

where $JW_o = \{jw((g, o)) \mid (g, o) \in RR\}$, $RW_o = \{rw((g, o)) \mid (g, o) \in RR\}$. *When the choice of options, the set of relevant reasons and/or the context are important for the weight function* w *under consideration, we write* $D_w(R, o_1, o_2, RR, c) = (v_1, v_2)$ *or* $D_w(R, o_1, o_2, RR, c, v_1, v_2)$. *When these are clear from the surrounding text, one may also write* $D_w(RR, v_1, v_2)$ *or* $D_w(RR) = (v_1, v_2)$.

Example 4 (To have coffee or not to have coffee)**.** *The weights of the reasons serve a role analogous to that of the weights in balancing scales, as specified in the weighing specification. To determine earlier outcomes, the justifying weight for option 1 is compared with the requiring weight of option 2, and vice versa. The initial detachments can now be implemented with weights.*

$$D_w(R, o_1, \neg o_1, \{r_1, r_2\}, c_1) = (+, -)$$

Since $jw(c_1, r_2) = 2 > 1 = rw(c_1, r_1)$*, this yields* $v_1 = +$*. Since* $jw(c_1, r_1) = 1 < 2 = rw(c_1, r_2)$*, this yields* $v_2 = -$*.*

$$D_w(R, o_1, \neg o_1, \{r_1, r_2\}, c_2) = (-, +)$$

Since $jw(c_2, r_2) = 1 < 2 = rw(c_2, r_1)$*, this yields* $v_1 = -$*. Since* $jw(c_2, r_1) = 2 > 1 = rw(c_2, r_2)$*, this yields* $v_2 = +$*.*

Following Tucker (2025), the weights are not reified—see, e.g., the discussion in Chap. 1 of the book. Therefore, in the principle-based analysis, there is no explicit reference to the weights. Instead, the set of detachment functions is defined so that it can be expressed using all possible weighting functions. Accordingly, the definition of dual-scale detachment proceeds in two steps. The first is the introduction of detachment functions, and the second is their implementation through these operators in the subsection Detachment Functions.

In maximum weighing, only the largest weights are compared to determine the output. Other than that, the comparison of options works similarly to that in additive dual-scale detachment. Later, the idea of the strongest reason is developed: in that case, the justifying and requiring weights must come from the same reason. In maximum weighing, the maximum justifying and the maximum requiring weight for a particular option may come from different reasons. The detachment function is indexed with an M as a superscript to indicate the use of the max function.

Definition 5 (Maximum Weighing)**.** *Let* $Max(S)$ *give the maximal element of a finite set of nonnegative real numbers* S*. A maximum detachment function* D_w^M *assigns to each pair of options* o_1 *and* o_2 *a subset of the reasons* $RR \subseteq R(o_1, o_2)$ *for this pair, a context* c*, a weight function* w*, and two scale* *values (v_1, v_2) using Max in the following way:*

$$v_1 = \begin{cases} + & \textit{if } Max(JW_{o_1}) > Max(RW_{o_2}) \\ - & \textit{if } Max(JW_{o_1}) < Max(RW_{o_2}) \\ 0 & \textit{otherwise} \end{cases}$$

$$v_2 = \begin{cases} + & \textit{if } Max(JW_{o_2}) > Max(RW_{o_1}) \\ - & \textit{if } Max(JW_{o_2}) < Max(RW_{o_1}) \\ 0 & \textit{otherwise} \end{cases}$$

Example 5 (To have coffee or not to have coffee)**.** *In the example so far, only 3 reasons were introduced, of which only 2 were considered in this 2-reason case. Max detachment will yield the same outcome. However, if three reasons, or perhaps more, are considered, max detachment can be used to obtain a different outcome. MAX detachment may remove the reason that uses "time" as ground for a reason from consideration, that is r_1, because it is less weighty than r_3.*

The definitions are quite general in the sense that, for example, grounds can be instantiated with propositional or first-order formulas, options can be instantiated with expressions from an action calculus, and contexts can be instantiated with default weights and modifiers. However, the definition of weighing is also quite specific, in that thresholds are not assumed (either for individual reasons or for comparing the two pans), nor does it include default values or modifiers in the weighing function. Tucker discusses some of these extensions; some of them are included in the earlier analysis of Knoks and Van der Torre (2023). From a technical point of view, these extensions would significantly complicate the principles and the principle-based analysis.

Additional Operators

Additional operators exist; a few discussed in the work of Tucker(2025) are shown below. Although they are specified here and the analysis can be extended to them, the analysis presented in this dissertation does not discuss these operators. They have been analyzed using slightly different specifications of detachment functions by Knoks et al. (2024).

Instead of additive aggregation, one can consider multiplicative aggregation. In this way, the reasons for the same option are aggregated by multiplying the weights.

Definition 6 (Multiplicative Weighing)**.** *Let $\Pi(S)$ be the product of a finite set of real numbers S, with the borderline case $\Pi(\emptyset) = 1$. A multiplicative detachment function D_w^{Π} assigns to each pair of options o_1 and o_2 a subset of the reasons $RR \subseteq R(o_1, o_2)$ for this pair, a context c, a weight function w, and two scale values (v_1, v_2) using Π as follows:*

$$v_1 = \begin{cases} + & \textit{if} \prod\limits_{x \in JW_{o_1}} x > \prod\limits_{y \in RW_{o_2}} y \\ - & \textit{if} \prod\limits_{x \in JW_{o_1}} x < \prod\limits_{y \in RW_{o_2}} y \\ 0 & \textit{otherwise} \end{cases}, \quad v_2 = \begin{cases} + & \textit{if} \prod\limits_{x \in JW_{o_2}} x > \prod\limits_{y \in RW_{o_1}} y \\ - & \textit{if} \prod\limits_{x \in JW_{o_2}} x < \prod\limits_{y \in RW_{o_1}} y \\ 0 & \textit{otherwise} \end{cases}$$

where $JW_o = \{jw((g, o)) \mid (g, o) \in RR\}$, $RW_o = \{rw((g, o)) \mid (g, o) \in RR\}$.

One may also consider using a threshold to place weights on the scales. This removes small weights because these weights are considered too small to be relevant.

Definition 7 (Additive Threshold Weighing)**.** *Given a number $t \in \mathbb{R}$, an additive threshold detachment function D_w^t assigns to each pair of options o_1 and o_2 a subset of the reasons $RR \subseteq R(o_1, o_2)$ for this pair, a context c, a weight function w, and two scale values (v_1, v_2) using $\sum$ as follows:*

$$v_1 = \begin{cases} + & \textit{if} \sum\limits_{x \in JW_{o_1} \wedge x > t} x > \sum\limits_{y \in RW_{o_2} \wedge y > t} y \\ - & \textit{if} \sum\limits_{x \in JW_{o_1} \wedge x > t} x < \sum\limits_{y \in RW_{o_2} \wedge y > t} y \\ 0 & \textit{otherwise} \end{cases}, \quad v_2 = \begin{cases} + & \textit{if} \sum\limits_{x \in JW_{o_2} \wedge x > t} x > \sum\limits_{y \in RW_{o_1} \wedge y > t} y \\ - & \textit{if} \sum\limits_{x \in JW_{o_2} \wedge x > t} x < \sum\limits_{y \in RW_{o_1} \wedge y > t} y \\ 0 & \textit{otherwise} \end{cases}$$

where $JW_o = \{jw((g, o)) \mid (g, o) \in RR\}$, $RW_o = \{rw((g, o)) \mid (g, o) \in RR\}$.

Another function used in "The Weight of Reason" by Tucker (2025, p 88) is Relative Weight Satisficing. This function states that the justifying weight must be at least twice that of the requiring weight for a permission to be detached. The requiring weight should be at least twice as great as the justifying weight for impermissibility to be detached.

Definition 8 (Relative Weight Satisficing Weighing)**.** *A relative weight satisficing detachment function D_w^{RWS} assigns to each pair of options o_1 and o_2 a subset of the reasons $RR \subseteq R(o_1, o_2)$ for this pair, a context c, a weight function w, and two scale values (v_1, v_2) using $\sum$ and scalars as follows:*

$$v_1 = \begin{cases} + & \textit{if} \sum\limits_{x \in JW_{o_1}} x > 2 * \sum\limits_{y \in RW_{o_2}} y \\ - & \textit{if}\, 2 * \sum\limits_{x \in JW_{o_1}} x < \sum\limits_{y \in RW_{o_2}} y \\ 0 & \textit{otherwise} \end{cases}, \quad v_2 = \begin{cases} + & \textit{if} \sum\limits_{x \in JW_{o_2}} x > 2 * \sum\limits_{y \in RW_{o_1}} y \\ - & \textit{if}\, 2 * \sum\limits_{x \in JW_{o_2}} x < \sum\limits_{y \in RW_{o_1}} y \\ 0 & \textit{otherwise} \end{cases}$$

where $JW_o = \{jw((g, o)) \mid (g, o) \in RR\}$, $RW_o = \{rw((g, o)) \mid (g, o) \in RR\}$.

Another function discussed in "The Weight of Reason" by Tucker (2025, p 88) is Absolute Weight Satisficing. This function detaches a permission when the justifying weight is sufficient, for example, at least 100. If the justifying weight is less than 100, then impermissibility is detached. Requiring weight is not considered in this function.

Definition 9 (Absolute Weight Satisficing Weighing)**.** *An absolute weight satisficing detachment function D_w^{AWS} assigns to each pair of options o_1 and o_2 a subset of the reasons $RR \subseteq R(o_1, o_2)$ for this pair, a context c, a weight function w, and two scale values (v_1, v_2) using $\sum$ and a threshold of* 100 *units in the following way:*

$$v_1 = \begin{cases} + & \textit{if} \sum\limits_{x \in JW_{o_1}} x > 100 \\ - & \textit{if} \sum\limits_{x \in JW_{o_1}} x < 100 \\ 0 & \textit{otherwise} \end{cases}, \quad v_2 = \begin{cases} + & \textit{if} \sum\limits_{x \in JW_{o_2}} x > 100 \\ - & \textit{if} \sum\limits_{x \in JW_{o_2}} x < 100 \\ 0 & \textit{otherwise} \end{cases}$$

where $JW_o = \{jw((g, o)) \mid (g, o) \in RR\}$, $RW_o = \{rw((g, o)) \mid (g, o) \in RR\}$.

3.3.4 Weight Function Constraints

The specified weight functions may or may not satisfy particular constraints. Here, multiple constraints are specified for a weight function to satisfy. One may instantiate a weight function such that a particular constraint applies or, through analysis, find that a weight function satisfies a particular constraint. These constraints may be understood more generally as weight-function principles. However, here they are introduced as constraints that characterize particular detachment functions and are used to constrain a weight function to instantiate a particular detachment function.

A natural idea that the definitions do not specify is that the weight of a reason is the same within the same pairwise comparison. That is, the order in which the options are given to the weight function does not matter.

Definition 10 (Symmetrical Weight Function)**.** *A weight function w is symmetrical when for each r and each two options $o_1, o_2 \in O$, it satisfies $w(o_1, o_2, r) = w(o_2, o_1, r)$.*

The single proportion assumption is a central idea in Tucker's(2025) book and formulates the common assumption that the justifying weight of a reason is equal to the requiring weight of a reason.

Definition 11 (Single Proportion Weight Function)**.** *A weight function w is a single proportion weight function when, for each r, it satisfies $jw(r) = rw(r)$.*

In (Knoks & Van der Torre, 2023), a balancing operator called simple counting is introduced. Simple counting can be implemented by assigning equal weight to each weight-term.

Definition 12 (Uniform Weight Function)**.** *A weight function w is uniform when it is a single proportion weight function, and if a reason has weight, then it is the same weight, we call it $w_0 \in \mathbb{R}_{>0}$, so for all reasons, $jw(r) = rw(r) = w_0$.*

More experimental types

This idea of creating constraints or principles for weight functions can be continued. Below are a few additional ideas for specifying the principles of the weight functions which can be useful when constructing a weight function.

The specification of Atomism is given by Tucker (2025) in the following way:

> "Atomism: a feature that is a reason in one context is a reason with the same polarity and weight in every other context. Suppose that your promise to ϕ is a reason for ϕ in some

> context. Atomism holds that it is a reason and, more specifically, an equally weighty reason for ϕ in any choice context."

A principle for weight functions that can be used to describe Atomism is given below, and we call it the context-free weight function principle. It may be understood as an attempt to define an atomistic weight function; if accepted, the function can be called an atomistic weight function.

Definition 13 (Context-Free Weight Function). *A weight function w is context-free when, for each $c, c' \in C$, and each $(g, o) = r \in RR$, with $o = o_1$ and $o_2 \in \bar{o_1}$ or $o = o_2$ and $o_1 \in \bar{o_2}$, it satisfies $w(R, c, RR, o_1, o_2, r) = w(R, c', RR, o_1, o_2, r)$. Here, $\bar{o}$ signifies the set of options part of the context that are not o.*

The specification of contrastivism by Tucker (2025) is given in the following way:

> "Contrastivism about Reasons: for some possible ϕ, the reason for ϕ can vary as you vary the alternative. That is, as you vary the alternative, you can vary whether something has (justifying or requiring) weight for ϕ or you can vary how weighty some reason is."

It is easier to specify non-contrastivism as a principle because the weights do not change when the alternative is changed.

Definition 14 (Non-Contrastivist Weight Function). *A weight function w is non-contrastivist when for all options o, o', o'' and for each reason $r = (g, o)$ it satisfies $w(o, o', r) = w(o, o'', r)$.*

In abstract argumentation, arguments attack each other. An attack means that the attacked argument is not acceptable. A weight system inspired by this idea is defined as follows. A ground attacks an option if it is a reason for the option to be impermissible; the reason pushes the option toward impermissibility. This occurs when a reason for an alternative option has requiring weight against the "attacked" option.

Definition 15 (Attack). *A weight function w encodes attacks when for all reasons r the weight is $w(r) = (0, 1)$. One can say that ground g attacks option o when $rw(o, o', (g, o')) = 1$, $jw(r) = 0$.*

In artificial neural networks, each node influences a successor node with a single real number. This means that the number, being real ($\mathbb{R}$), can be positive or negative. The weight function can sustain such a relationship at the dynamic level; dynamic detachment is discussed later. This comes down to representing the influence of a ground on an option with a single real number across comparisons and contexts.

Definition 16 (Single Representatives)**.** *If, for all g and o, the influence of g on the status of o can be represented by a single number, then for all alternatives o', o'', find $w(o, o', (g, o))$ and $w(o, o', (g, o'))$. The influence of g on o is then $jw(o, o', (g, o)) - rw(o, o', (g, o')) = jw(o, o'', (g, o)) - rw(o, o'', (g, o''))$.*

3.3.5 Detachment Functions

By using the weight constraints or principles for the weight functions together with the balancing operators, detachment functions can be specified. Detachment functions implement detachment relations. One can think of this as a collection of elements that together simulate, produce, or implement the detachment relation. One may add the symmetrical weight-function constraint to these specifications, though Tucker (2025) does not enforce it; therefore, it is not enforced here either. However, for the principle-based analysis, a symmetrical weight system is assumed.

The first detachment function specified here is dual-scale detachment. This detachment function requires the fewest principles to specify.

Definition 17 (Dual-scale Detachment)**.** *Let D be a detachment function. D is a* dual-scale detachment function *if and only if there exists a weight function w such that $D_w = D$.*

In single-scale detachment, the justifying and requiring weights of reasons are the same. This means that the two balancing scales mirror each other, effectively becoming a single scale.

Definition 18 (Single-scale Detachment)**.** *Let D be a detachment function. Then D is a* single-scale detachment function *if and only if there exists a single-proportion weight function w (i.e. $jw = rw$) such that $D_w = D$.*

In uniform detachment, it is not only the requiring and justifying weights of reasons that are the same. Rather, the (justifying and requiring) weights of all reasons are the same. In fact, this method effectively simply counts and compares the number of reasons that support each option.

Definition 19 (Uniform Detachment)**.** *Let D be a detachment function. D is a* uniform weight detachment function *if and only if there exists a uniform weight function w (i.e. $jw = rw = w_0$) such that $D_w = D$.*

As a benchmark, the well-known maximum function is used to implement the maximum detachment function.

Definition 20 (Maximum Detachment)**.** *A detachment function D is a* maximum weight detachment function *if and only if there exists a weight function w such that $D_w^M = D$.*

3.3.6 Principles

This section provides the principles that will be used in the principle-based analysis. These principles describe properties of detachment functions that are not necessarily desirable, but that reveal something about their behavior. The principle-based analysis itself is discussed in Section 3.4. The attentive reader may observe that the detachment functions are also sets of weight function principles, and so the analysis revolves around testing which of these principles can coexist and which principles cancel each other out.

Nine principles are presented. The first three restate (some of) the principles formulated by Knoks and Van der Torre (2023) in the present framework; the next three restate (some of) the principles formulated by Knoks et al. (2024); and the final three principles are presented by de Wit, Knoks, and Van der Torre (2025a). Some of the other principles of Knoks and Van der Torre (2023) and Knoks et al. (2024) are also listed afterward.

The principles are always specified for detachment functions in the form $D(R, o_1, o_2, RR, c) = (v_1, v_2)$. It is important to note that the context c is kept fixed, ensuring that the weight function, on which the detachment function implicitly relies, assigns consistent (embedded) weights to the relevant reasons. This does not mean that the detachment function is restricted to a single context. The definition of detachment functions allows the weight function to account for context. Because the weight function is context-sensitive, the weights assigned to the reasons may vary across contexts. Many principles modify the set of input reasons RR, making it unlikely that these principles will hold universally as the context changes, since the weights of the relevant reasons may also vary in those cases.

Our first principle, Groundedness, states that an option can obtain the value $+$ or $-$ only if there is at least one reason supporting that valuation.

Principle 1 (Groundedness)**.** *A detachment function D satisfies* Groundedness*,* `Gr`*, if and only if*

1. *for every $(v_1, v_2) \in D(R, o_1, o_2, RR, c)$ such that $v_1 = +$ or $v_2 = -$, there exists at least one reason $(g, o_1) \in RR$, and*

2. *for every $(v_1, v_2) \in D(R, o_1, o_2, RR, c)$ such that $v_1 = -$ or $v_2 = +$, there exists at least one reason $(g, o_2) \in RR$.*

The principle of Anonymity expresses the idea that grounds (of reasons) can be replaced by other grounds without changing the outcome.

Principle 2 (Anonymity)**.** *A detachment function D satisfies* Anonymity*,* `An`*, if and only if, for any bijection $\pi : G \mapsto G$, then $D(R, o_1, o_2, RR, c) = D(R, o_1, o_2, RR^\pi, c)$ where $RR^\pi = \{(\pi(g), o) : (g, o) \in RR\}$.*

The principle of Unanimity expresses the intuitive idea of a unanimous vote. If all relevant reasons favor one of the two options, that option should be required.

Principle 3 (Unanimity)**.** *A detachment function D satisfies* Unanimity*,* `Ua`*, only when RR is non-empty such that:*

1. *if $RR = \{(g, o) \in RR \mid o = o_1\}$, then $D(R, o_1, o_2, RR, c) = (+, -)$, and*

2. *if $RR = \{(g, o) \in RR \mid o = o_2\}$, then $D(R, o_1, o_2, RR, c) = (-, +)$.*

The next principle, Polarity Monotony, states that the valuation of an option can only increase from $-$ to 0 to $+$ as reasons favoring that option are added.

Principle 4 (Polarity Monotony)**.** *Let $- < 0 < +$. A detachment function D satisfies* Polarity Monotony*,* `PoMn`*, only where:*

1. *if $D(R, o_1, o_2, RR, c) = (v_1, v_2)$ and $D(R, o_1, o_2, RR \cup \{(g, o_1)\}, c) = (v_1', v_2')$, then $v_1' \geq v_1$ and $v_2' \leq v_2$, and*

2. *if* $D(R, o_1, o_2, RR, c) = (v_1, v_2)$ *and* $D(R, o_1, o_2, RR \cup \{(g, o_2)\}, c) = (v_1', v_2')$*, then* $v_1' \leq v_1$ *and* $v_2' \geq v_2$.

The principle of Union Monotony expresses the following idea: if a detachment function assigns the same scale values to two choice options involving two different sets of reasons, then it also assigns the same scale values to a set of reasons that contains all of those different reasons (assuming that a valuation for such a set exists).

Principle 5 (Union Monotony)**.** *A detachment function* D *satisfies* Union Monotony*,* `UnMn`*, only if (1)* $D(R, o_1, o_2, RR, c) = (v_1, v_2)$ *and (2)* $D(R, o_1, o_2, RR', c) = (v_1, v_2)$*, and then also (3)* $D(R, o_1, o_2, RR \cup RR', c) = (v_1, v_2)$.

The principle of Sensitivity expresses the following intuitive idea: if one starts with a pair of balanced pans (equal weights on each side) and adds a new reason, the balance is disturbed.

Principle 6 (Sensitivity)**.** *A detachment function* D *satisfies* Sensitivity*,* `Se`*, if and only if for every* $(0, 0) \in D(R, o_1, o_2, RR, c)$*,* $r \in R(o_1, o_2)$*, and* $(v_1, v_2) \in D(R, o_1, o_2, RR \cup \{r\}, c)$*, then* $v_1 \neq 0 \vee v_2 \neq 0$.

The principle of Input Symmetry holds that the order of the options is not important. Input Symmetry is motivated by the overall shape of the framework used here and expresses a natural constraint on detachment functions.

Principle 7 (Input Symmetry)**.** *A detachment function* D *satisfies* Input Symmetry*,* `InSym`*, if and only if when* $D(R, o_1, o_2, RR, c) = (v_1, v_2)$*, then* $D(R, o_2, o_1, RR, c) = (v_2, v_1)$.

The next principle is motivated by Tucker's(2025) book. As its name suggests, the principle of No Dilemmas expresses the idea that a detachment function rules out dilemmas or cases in which both choice options are associated with negative scale values.

Principle 8 (No Dilemmas)**.** *A detachment function* D *satisfies* No Dilemmas*,* `NoDi`*, if and only if there is* ***no*** $(v_1, v_2) \in D(R, o_1, o_2, RR, c)$*, such that* $(v_1, v_2) = (-, -)$.

Our final principle, One Permission, is also new and was introduced by de Wit, Knoks, and Van der Torre (2025a). This principle expresses the intuitive idea that the detachment function allows for only one kind of permission.

Principle 9 (One Permission)**.** *A detachment function D satisfies* One Permission*,* `OnPe`*, if and only if there is no $(v_1, v_2) \in D(R, o_1, o_2, RR, c)$ such that $v_1 \neq v_2$ and $v_1, v_2 \geq 0$.*

Undiscussed Principles. In the articles (Knoks & Van der Torre, 2023; Knoks et al., 2024), other principles are also studied. So far, they either did not fit the new specification of the detachment function well enough to translate or were not sufficiently insightful to merit proofs. However, here is a list of the principles. They featured in the articles (Knoks & Van der Torre, 2023; Knoks et al., 2024) but were not presented above. Further principles can be found in the abstract argumentation literature, for example: Amgoud et al. (2017), Baroni and Giacomin (2007), and Van der Torre and Vesic (2018).

UD Universal Domain

Do all inputs have a valuation?

RUD Reason Universal Domain

Is the set of inputs that have a valuation closed?

Re Relevance

Do only relevant reasons have influence?

FiVa Fixed Value

Does each ground always support the same options?

Ne Neutrality

Do the same amount of justifying weight and requiring weight have the same magnitude of influence?

Mn Monotony

Does adding reasons change the output value?

PoCu Polarity Cut

Does removing reasons that have the opposite value from the output value change the outcome?

PoSw Polarity Switching

Is it possible to modify the input by adding or removing reasons so that the output value changes from $-$ to $+$ or vice versa?

Null Nullification

Is it possible to turn a non 0 output value into a 0 output value?

CoRe Commensurate Removal

Does removing the same amount of weight from either side of a scale lead to the same output values?

Sym Symmetry

If the reasons change sides, does the output valuation then also change sides?

Further detachment principles may involve specifying the negative version of previously presented principles. For example, the principle of anti-monotony may be specified. This is useful because the principle-based analysis either provides a proof of the principle or a counterexample. When a counterexample is provided, it is sufficient to reject the principle, it is not necessary to adopt the negation of the principle because counterexamples to the negation of the principle may also exist. It is good practice, therefore, to also develop and find proofs for the negative formulation of interesting principles.

Other Classes of Principles. The principles specified above work on the detachment function and mainly modify the RR set. However, the specified weight-function constraints may also be understood as principles. Further areas of principles may be considered: the set of grounds G, the set of options O, the set of values V, the set of contexts C, and the reasons function R. At the moment, this is still somewhat difficult to imagine. However, for options, the idea of "Maximalism" may be turned into a principle regarding the set of options, whereas "Minimalism" may be specified as another principle regarding the options. Furthermore, the reasons function currently is defined as a function, though this may also be turned into a principle, i.e., a reasons relation that can satisfy the principle of being functional. To do this, the grounds and options will need to be in the same set, call it the set of arguments $A = G \cup O$. Or alternatively, $A \not\subseteq G \cup O$, and they are imaginary or hypothetical elements to facilitate a particular detachment relation.

3.3.7 Dynamic Scale

The attentive reader may have noticed that, so far, only pairwise comparisons between options have been discussed. However, when there are more than two options, more than that is needed. The current detachment functions create a relation over all options that looks like a preference relation of sorts, and there are nine possible relations between two options, that is, $\{-, 0, +\}^2$. This is already a lot, but what is needed is the "all things considered" deontic status of each option in the set. To do this, Tucker (2025, Ch. 9) uses a dynamic scale; this scale can lift the deontic status of pairs to the deontic status of individual options. However, the only possible deontic statuses considered are: permitted and not permitted; that is, $\{-, +\}$.

Definition 21 (Dynamic Scale Detachment)**.** *Let the values $V = \{+, -\}$ be positive and negative, or permissible and impermissible. Then, given a set of options O, a context $c \in C$ and a weighing function D_w associated with it, a dynamic detachment function D^* assigns values to each option o in O as follows: $D^*(R, O, RR, c)(o) = +$ if $D_w(R, o, o', RR, c) = (+, v)$ or $D_w(R, o, o', RR, c) = (0, v)$, for all $o' \in O$ such that $o' \neq o$; $D^*(o) = -$ otherwise.*

Alternatively, one can change the behavior around the case where a scale is in balance. Instead of detaching $+$, one may choose to also let this count as $-$. This is not unreasonable because if an option has no reasons in its favor, it may not automatically be considered permissible. Often, people do not consider something rational if they do not have a reason for the behavior, just as when the justification is counterbalanced by the required weight. Nevertheless, it may also be strange to say something is not permitted without specifying reasons.

One may also define the dynamic scale such that 0 is included as a value. This allows a way out of the problem of setting a good value when the scale is in balance. This also provides the space to introduce weak permission as an interpretation of 0.

Definition 22 (Dynamic Scale Detachment)**.** *Let $V = \{+,\ 0,\ -\}$. Given a set of options O, context $c \in C$, a reasons function R, and a detachment function $D(R, o, o', \mathrm{RR}, c) = (v_1, v_2) \in V \times V$,* *define*

$$D^{\star}(R, O, \mathrm{RR}, c)(o) = \begin{cases} + & \textit{if for all } o' \neq o\textit{: } D(R, o, o', \mathrm{RR}, c) \in \{(+, \cdot), (0, \cdot)\}, \\ - & \textit{if there exists } o' \neq o\textit{: } D(R, o, o', \mathrm{RR}, c) = (-, \cdot), \\ 0 & \textit{otherwise.} \end{cases}$$

3.3.8 Additional Specifications

Further interpretations and ideas for working with weight and reasons are presented here. Four additional ideas came up: intervals of weight, weight as a relation, weight as a ranking, and weight as probabilities.

Intervals of weight. The idea of weights as intervals is that a weight becomes an interval. The smallest interval is then presented practically as a single number, for example 5, but then as an interval as $[5, 5]$. The interval can be larger, for example, $[1, 5]$. This can be interpreted in a couple of ways. The first way is to view intervals as a form of leeway. This would mean that it can be modified by an agent to get the outcome they see fit. Alternatively, it can be used to give way to change by specifying a lower and upper bound that are acceptable to an agent. A second way to look at these intervals is to understand them as implementing uncertainty. For example, when reasoning backward about the weight of reasons, intervals may be set bounds on what the reasons' weights can be.

Relational Detachment. The work of Knoks and Van der Torre (2023) goes beyond numerical weighing. They instantiate a binary defeat relation on reasons and use this to determine the deontic state of options. Using the weight of reasons, it is possible to instantiate a defeat relation on reasons. Since each reason has two weights, a reason defeats another opposing reason when the justifying weight is strictly larger than the requiring weight of the opposing reason, and the requiring weight is larger than the justifying weight of the opposing reason. This is very similar to max detachment, though here the comparison is of two reasons, their justifying and requiring weights, while in max detachment, all the justifying weights for an option are put together in a set, and then the max value is compared to the max value of the other reasons' sets of weights. Here, the justifying and requiring weights remain connected to reasons, and the reasons are compared.

Definition 23 (Defeating Reasons)**.** *Given two options $o_1, o_2 \in O$, a weight function w, a reasons function $R \in SR$, a context $c \in C$ and a set of reasons RR, and given two reasons $r, r' \in RR$ such that $r = (g, o_1)$ and $r' = (g', o_2)$; we write $r' \prec r$ and say that r defeats r' if and only if $jw(r) > rw(r')$ and $rw(r) > jw(r')$.*

Notice that two reasons can stand in the $\prec$ relation only if they do not favor the same option in a pairwise comparison. Instead of $(r, r') \in \prec$, we write $r \prec r'$. An expression of the form $r \prec r'$ can be thought of in terms of r' having strictly more weight than r, or r' defeating r. With this in mind, the general definition of relational detachment can be stated.

Definition 24 (Relational Detachment)**.** *To use all 9 possible deontic states, relational detachment can be defined as follows.*

- *If there $\exists(g, o_1) = r \in RR$ such that $\forall(g', o_2) = r' \in RR$ are weaker than r, i.e. $r' \prec r = (g, o_1)$, then $(v_1, v_2) = (+, v_2)$.*
- *if $\exists(g, o_1) = r \in RR$ such that $\forall(g', o_2) = r' \in RR$ are stronger than r, i.e. $r \prec r'$ then $(v_1, v_2) = (-, v_2)$.*
- *Otherwise, $(v_1, v_2) = (0, v_2)$*
- *If there $\exists(g, o_2) = r \in RR$ such that $\forall(g', o_1) = r' \in RR$ are weaker than r, i.e. $r' \prec r = (g, o_2)$, then $(v_1, v_2) = (v_1, +)$.*
- *And if $\exists(g, o_2) = r \in RR$ such that $\forall(g', o_1) = r' \in RR$ are stronger than r, i.e. $r \prec r'$, then $(v_1, v_2) = (v_1, -)$.*
- *otherwise $(v_1, v_2) = (v_1, 0)$.*

This definition aligns with the definition of relational detachment under single proportion weights introduced by Knoks and Van der Torre (2023). Furthermore, it accepts dual weights and acknowledges the different possible outputs for the dual-scale.

Ranking based interpretation. Ranking is a common approach to defeasible reasoning. Ranking assigns elements a rank, with 0 or 1 as the lowest, and the other ranks increase from there. The lowest rank is understood as the most normal situation. Then, more abnormal cases occur in higher ranks; only ranks above can be empty, i.e., no empty ranks exist between two non-empty ranks. The idea is to have specific information and then assume the most normal situation consistent with that information. The approach of Tucker (2025, Ch. 4) also has a section that compares the dual-scale with the ranking-based approach. However, the given ranking type assumes that a higher rank is better than a lower one. This relates to how preference is typically modeled. Here, the idea is that more preference is better; also, there are no empty ranks because they have no purpose in ranking because one looks at outranking rather than aggregating ranks.

To perform a ranking where a low rank is better, different machinery needs to be implemented. Instead of more weight leading to $+$, less weight will lead to $+$. This seems possible, though it may require more work than is apparent at face value.

Probabilistic Weights. Probabilities are also often referred to as weights. However, probabilities come with additional requirements. Probabilities exist in the domain $[0, 1]$, and sometimes $[0, 1)$ is used instead. In addition, the probabilities satisfy sigma-additivity, are nonnegative, and sum to 1 over the entire sample space.

These requirements are relatively complex to implement for the specified systems. Limiting the weight domain to $[0, 1]$ is easy and can be expressed as a principle/constraint on the weight function. However, the other two are more complex. In particular, sigma-additivity might be difficult to implement, since the reasons lack the background structure needed to support it. Setting the probability of the entire sample space to 1 may also be difficult, as it requires specifying the entire domain. Nevertheless, there are domains for which these requirements are achievable.

Assuming that these constraints can be met, another problem surfaces— aggregation. The specified aggregation methods do not apply because they require sigma-additivity. And if multiplication is used, the probabilities become very small as reasons are added, since the weights are between 0 and 1. A possible alternative operator is the function $\min()$, which returns the lowest probability. This is sensible, since the probability cannot exceed that for the most improbable reason.

This may be a valuable direction; however, it seems complex at face value and will not yield transferable results on the principles, as it requires a different aggregation method. However, it will be valuable, for comparison purposes, to determine which principles this method follows.

Research has focused on weighted systems with weights in the domain $[0, 1]$, for example, in the article by Amgoud et al. (2017).

3.4 Principle-Based Analysis

The main result consists of a principle-based analysis of the balancing operators: additive and max; these results were obtained by de Wit, Knoks, and Van der Torre (2025a). The additive balancing operator is supported by Tucker (2025); the book also uses max. In the analysis presented here, max serves as a benchmark because it is common in the satisficing literature. The principles are properties that can be satisfied by the detachment functions; however, they are not necessarily desirable. The four detachment functions that were defined in Sections 3.3.5 (dual-scale, single-scale, uniform detachment, maximum) are analyzed using the 9 principles of Section 3.3.6. These principles were selected because they reveal an interesting property of the functions. Other principles exist, but some have the same outcome and thus do not add further differentiation between the functions. Ideally, the principles can single out a particular function. That way, proving that a detachment function satisfies a particular principle allows one to postulate that it uses or is simulated by a particular operator. Table 3.2 summarizes the results of the analysis. The detachment functions are listed in the upper row; the principles are listed in the left column. A $\checkmark$ in a cell means that the principle listed in the left row is satisfied (i.e., guaranteed) by the detachment function listed in the column. A $\times$ in a cell means that the given principle is not satisfied by the given detachment function, i.e., that a counterexample exists. Proof theoretically, this is a safe way to obtain results. There also exist opposite versions of some principles. However, to determine whether such a principle is satisfied, it should be formulated and subjected to the same process.

The experiments in this table assume a symmetric weight function. This is a sensible requirement for weight functions and makes it so that $w(o_1, o_2, r) = w(o_2, o_1, r)$. Dropping this constraint, or principle, if you will, allows implementation of different weights when considering o_1 versus o_2

rather than o_2 versus o_1—similar to a home team benefit in football, if you will—expressed in changed weights. This means the result can differ when comparing option 1 with option 2 rather than when comparing option 2 with option 1.

	Dual-Scale	Single-Scale	Uniform	Maximum
1.Gr	✓	✓	✓	✓
2.An	×	×	✓	×
3.Ua	×	✓	✓	×
4.PoMn	✓	✓	✓	✓
5.UnMn	×	×	×	✓
6.Se	✓	✓	✓	×
7.InSym	✓	✓	✓	✓
8.NoDi	×	✓	✓	×
9.OnPe	×	✓	✓	×

Table 3.2: Summary of the principle-based analysis

The table shows that dual-scale and single-scale are distinguished by three principles: Unanimity, No Dilemmas, and One Permission. These principles are satisfied by single-scale and not satisfied by dual-scale. The principle of Unanimity is not satisfied by dual-scale detachment because one of the weights may be 0. Therefore, even if there is a unanimous vote for, say, o_1, the output can still be $(0, +)$ if the justifying weight is 0, or it can be $(+, 0)$ if the requiring weight is 0. With single-scale detachment, this is impossible, since the justifying and requiring weights are always the same. Dual-scale detachment does not satisfy No Dilemmas because it allows for the aggregate requiring weights of reasons supporting each option to outstrip the aggregate justifying weights of reasons supporting each option. Single-scale detachment, by contrast, does not allow for such a possibility because, with it, the aggregate justifying weights of reasons supporting each option equal the aggregate requiring weights of reasons supporting each option. Thus, whereas dual-scale detachment admits dilemmas, single-scale detachment avoids them by design. Tucker also discusses this issue in his book as an illustration of the expanded domain of the dual-scale model of weighing reasons. The fact that dual-scale detachment does not satisfy One Permission, whereas single-scale detachment does, serves as an illustration.

Uniform detachment satisfies all of the principles except for Union Monotony. This principle is about combining sets of reasons and assessing whether the output remains the same. What can happen is that both RR and RR' support the same output, e.g. $(+, -)$, but their union $RR \cup RR'$ supports the opposite output because there is a considerable overlap (i.e. $RR \cap RR' = \{r = (g, o) \mid o = o_1 \vee o = o_2, r \in RR \vee r \in RR', RR, RR' \subseteq R(o_1, o_2)\} \neq \emptyset$) between the reasons that support o_1, but not between the reasons that support o_2. Uniform detachment is similar to single-scale detachment. The only (revealed) difference between them has to do with the principle of Anonymity, which single-scale detachment does not satisfy. In single-scale detachment—and contrary to uniform detachment—reasons can have different weights. Thus, it should not come as a surprise that swapping them can alter the output.

As Table 3.2 makes clear, maximum detachment does not satisfy No Dilemmas and One Permission. More interestingly, it does not satisfy the Sensitivity principle. This is due to the fact that the addition of a new reason does not necessarily increase the maximum weight amongst all weights, whether justifying or requiring—of the reasons that are already present in RR. So, maximum detachment is not necessarily sensitive to new reasons. It is also monotonic under the union of reasons, whereas additive aggregation-style scale-based detachments are not.

The principle-based analysis shows that to obtain dilemma-free outputs, a single-proportion weight function should be used. If, however, representation of potential dilemmas is included rather than excluded, then single proportion weights should be dropped explicitly.

The proofs supporting the principle-based analysis are fairly straightforward. Two examples are presented here, with the remainder in Appendix A.

Proposition 1. *There is a weighing function such that for the dual-scale detachment function D_w, there are R, o_1, o_2, RR, c such that $(v_1, v_2) \in D(R, o_1, o_2, RR, c)$ where $v_1 = -$ and $v_2 = -$.*

Proof. Proof by counterexample; consider the following weighing function that shows that dual-scale detachment fails No Dilemmas. Let $r_1 = (g_1, o_1)$ and $r_2 = (g_2, o_2)$. And let $(v_1, v_2) = D_w(R, o_1, o_2, \{r_1, r_2\}, c)$ be such that $w(R, o_1, o_2, c, r_1) = (1, 3)$, $w(R, o_1, o_2, c, r_2) = (1, 3)$. Then since $jw(r_1) - rw(r_2) = 1 - 3 = -2$, it follows that $v_1 = -$. And since $jw(r_2) - rw(r_1) = 1 - 3 = -2$, it follows that $v_2 = -$, therefore $D_w(\{r_1, r_2\}) = (-, -)$, which completes the counterexample.

We conclude that dual-scale detachment fails No Dilemmas. $\square$

Proposition 2. *Single-scale detachment satisfies No Dilemmas.*

Proof. Suppose, toward a contradiction, that single-scale detachment does not satisfy No Dilemmas. This means that there is some single detachment function D such that, for some o_1, o_2, then $RR \subseteq R(o_1, o_2)$, and as for c, there is $D(R, o_1, o_2, RR, c) = (-, -)$. By Definition 18 (single-scale), there exists a weight function w such that $D_w(R, o_1, o_2, RR, c) = (-, -)$ and $jw = rw$. Definition 4 (Additive dual-scale Weighing), gives, first, that $\Sigma_{(g_1,o_1)\in RR} jw((g_1, o_1)) - \Sigma_{(g_2,o_2)\in RR} rw((g_2, o_2)) < 0$, as well as, second, that $\Sigma_{(g_2,o_2)\in RR} jw((g_2, o_2)) - \Sigma_{(g_1,o_1)\in RR} rw((g_1, o_1)) < 0$. But $jw = rw$ tells us that $\Sigma_{(g_1,o_1)\in RR} jw((g_1, o_1)) = \Sigma_{(g_1,o_1)\in RR} rw((g_1, o_1)) = x$. Similarly, it tells us that $\Sigma_{(g_2,o_2)\in RR} jw((g_2, o_2)) = \Sigma_{(g_2,o_2)\in RR} rw((g_2, o_2)) = y$. Therefore, it follows that $y < x$ and $x > y$, which is a contradiction. □

The dual-scale detachment function shows the least amount of constraints of the three additive detachment functions. This function can model multiple assignments of deontic status that the other two cannot, including the notorious dilemma case $(-, -)$. In doing so, it provides more room for the concept of permission to exist; as a consequence, the value 0 can be interpreted as a weak or implicit permission. If one has no convictions to support the uniform weight or single proportion assumption, then there is no reason not to use the dual-scale function.

That input symmetry is satisfied is not strange since symmetrical weight functions were enforced.

Proposition 3. *Input symmetry requires a symmetric weight function.*

Proof. When the weight function is not symmetric, then it can be $w(R, o_1, o_2, c, r) \neq w(R, o_2, o_1, c, r)$. Therefore, when $D_w(R, o_1, o_2, RR, c)$ is called, it will use for each $r \in RR(o_1, o_2)$ the weights of $w(R, o_1, o_2, c, r)$.

When $D_w(R, o_2, o_1, RR, c)$ is called to check for input symmetry, $w(R, o_2, o_1, c, r)$ weights are used. They may be different because the weight function is not symmetric, therefore it is not guaranteed that $D_w(R, o_1, o_2, RR, c) = D_w(R, o_2, o_1, RR, c)$. □

3.5 Discussion

The framework is designed to implement key ideas from "The Weight of Reasons" (Tucker, 2025). It does so by understanding the balancing process as the implementation of a detachment relation, as introduced by Knoks and Van der Torre (2023). Here, detachment is based on the standard logical concept known as *modus ponens* or implication. The relation, in its first conception, relates options to a value interpreted as a deontic status, i.e., permissible or impermissible. To determine this, reasons are presented that support the options and carry dual weights. The work implements functionality for: pairwise comparison, context, *ceteris paribus*, weights, and balancing operators. To understand the work of dual-scale, one can view it as making explicit particular assumptions in the single-scale model. The implementation can impose specific principles as constraints on one parameter, thereby enforcing a single-scale detachment relation. To understand the impact of principles, the principle-based analysis is crucial. By recognizing particular systems as implementing particular assumptions, these assumptions can be translated into principles that serve as optional constraints on the "unbiased" model. It remains possible that further assumptions may be identified in this "unbiased" base model. Though this seems unlikely.

So far, the principles have focused on the detachment relation and how the reasons and valuations of options relate. To do this, the *ceteris paribus* style of reasoning needed to be possible. The main idea here is that weight is context-dependent and that reasons occur in that context. Therefore, if the set of reasons is modified, the context is modified. However, if the principles concern modifying the set of reasons, the assumption must be that only that modification changes, and the context remains the same. Whether this is philosophically acceptable is a matter for discussion. The path this work has taken is to separate the idea of context from that of placing reasons on a balance scale. This means that a particular degree of agency is created by allowing an agent to line up all the reasons that arise and choose a particular subset to place on the scales. This means that the context remains the same while allowing modification of the reasons considered in the detachment process. This also models the case of incomplete knowledge: if an agent is unaware of a particular reason in the context, they lack the access needed to place that reason on the scales.

The work can be taken a step further by focusing on the reasons function R. This function is a bipartite graph that yields a set of reasons based on pairwise comparisons of two options. The context, together with the weight function, yields the weight of reasons. The reasons function therefore needs to be such that reasons that do not have weight do not occur in the set. It may be that the same pairwise comparison of options in a different context has different reasons, each with its own weight.

Secondly, many graph-based approaches use more complex graphs, for example, those with cycles or intermediate elements. It is possible to argue whether this is already part of the complete graph implemented by R. However, it may also become a more explicit part of R. By formulating principles for the reasons function, the property of R being bipartite can become a principle. The following principles may be: acyclic graphs, the existence of intermediate elements between the grounds and options, and the intersection of grounds and options being empty.

Also, initial work has begun on principles for weight functions, here denoted as weight-function constraints. The weight principles here are relatively simple since uniform weights imply single-proportion weights. Symmetric weights are important because they denote a property that can be easily assumed. This would also open the door to options that stand on their own, (o, o), implying an option that is its own reason. And grounds (g, g) that are "optionally" true.

The context has not been studied and has remained abstract; the context has been structurally approached by Streit (2024) and Streit et al. (2023). However, it should be possible to study the relations between contexts more closely by, for example, defining the sub- and super-set relations on contexts, or studying principles and relations among disjoint contexts. Also, for the set of grounds and options, there should exist principles that implement modifications to the sets considered. Maximalism of options is one principle, and its opposite may be minimalism. Other principles may help clarify the different foundational sets. A further idea is to determine the influence of context through the specified detachment process, i.e., determine the weight of a reason through detachment based on reasons that occur in the context of that reason and that support or diminish its weight.

Leaving aside principles and turning to dual weights, it seems that dual weights is a simple modification; however, they may have a great impact on other domains. Typically, in mathematical approaches, a single weight is used. This is somewhat of an overly general statement; however, the idea of dual weights may have consequences for, e.g., statistics. However, whether this is true should be studied. Perhaps the single-proportion weights principle is often true, and thus many implementations may not have immediate influential effects. Given that, certain modeling problems may be solved by implementing dual weights.

Speaking of modeling, some steps have been taken to link the system presented in this dissertation to abstract argumentation and artificial neural networks. Abstract argumentation seems to use, to some extent, a small part of the expressivity of the specified detachment functions. Arguments can be seen as the union of the set of options and the set of grounds. Conflict arises when a ground is a reason for alternatives, making it required over the "attacked" argument. Thus, abstract argumentation does not use justifying weight and can express conflict with a requiring weight of 1, and no other numbers are needed. However, the graph is more complex because abstract argumentation typically allows arbitrary graphs over the arguments as vertices. Extension of the reasons function may enable the full incorporation of argument graphs. Most importantly, principles need to be found that can distinguish between particular argumentation semantics.

Artificial neural networks do not use dual weights, though they do use negative numbers. Still, artificial neural networks use only a single scale at each node, and there is no pairwise tournament. A layer of an artificial neural network that implements dual-scale may be understood as more densely connected than a "dense" layer. A dense layer in an artificial neural network connects each input node to each output node, i.e., it forms a fully connected graph. The grounds can be seen as the nodes on the input side, and the options as the nodes on the output side. Then the reasons are the relations between the grounds and the options. However, in the given system, the reasons have two weights and can differ across pairwise comparisons, i.e., the contextual reasons (cogs) have weight. Therefore, there are many more weights than in a dense layer of artificial neural networks.

The primary benefit of these frameworks, which reside within the dual-scale detachment functions, is that they can be studied using the specified principle-based approach. And, because of that, they can be compared with each other directly. Furthermore, the specification of the detachment function always confers a deontic status at the dynamic level, thereby benefiting decidability.

The final idea is to let the outcomes of the dynamic scale influence the context-and-reasons graph. The idea can be understood as letting the system eat its own tail. In this sense, the context serves as a form of memory that is modified as the system operates. The idea is simple; however, implementing it is complex. An ad hoc way to do this is to understand memory as a maze the agent walks through, leaving clues along the way. If a proper rendering of this is possible, one may study how the system behaves in the limit. Does the system converge on a particular value assignment, or does it diverge? Perhaps it finds a stable cycle along which it rotates?

Finally, the framework may be able to model the transplantation of options into a particular context, thereby engaging in a particular type of counterfactual or what-if type of reasoning. By using familiar options, reasons, and contexts, a new valuation can be crafted that relates an option to a context that has not been connected before. This line of research will enable verification of models with respect to their potential functioning. For example, "How would this model behave in this context when it has to choose between saving person 'A' and person 'B'?" This only becomes possible when the weights have been specified, so it might be that the scenario imagined does not explicitly exist yet in the detachment relation of the model.

3.5.1 Related Work

This chapter is built on the reason-based detachment framework proposed by Knoks and Van der Torre (2023) which has been further explored by Knoks et al. (2024) and by de Wit, Knoks, and Van der Torre (2025a), by implementing the ideas of Tucker (2025). This work goes beyond earlier articles in at least three ways. First, Tucker's dual-scale model of weighing reasons was specified in the reason-based detachment framework. Second, the framework was generalized to allow justifying and requiring weight of reasons, which led to restating the ideas and principles from earlier articles in a more general setting. And third, abstract contexts have been incorporated into the definition of detachment functions. Although the issue is not discussed in detail here, this opens the door to a more elegant way of addressing Holism regarding the weights of reasons, as outlined by Knoks et al. (2024).

Thematically, the work presented here is closely related to the formal model(s) of interaction between normative reasons developed by Streit (2024, Ch. 3) and Alcaraz et al. (2024) and Streit et al. (2023). Whereas the input-output logic methodology was followed, Alcaraz et al. (2024) and Streit et al. (2023) make use of formal argumentation theory (Baroni et al., 2018) and, more specifically, weighted argumentation (Amgoud et al., 2022; Amgoud et al., 2017). Whereas here the focus was on the dual-scale model, Streit et al. focus on the standard single-scale model of weighing reasons. They also explicitly model the interaction between disablers, modifiers, and reasons. The work also differs from that of Nair (2021), who explores issues relating to weights of reasons drawing on probability and confirmation theory; Pandžić and Graff (2024), who combine justification logic with numerical representation of strengths of reasons; and Sher (2019), whose thinking about weights of reasons draws on tools from decision theory.

Insofar as Tucker was followed and the balancing metaphor taken seriously, the approach parts way with the influential default logic-based model of Horty (2012) and the formal work it has inspired. A topic of further research may be to strengthen this relation by understanding how the triggering of reasons and the undercutting of reasons relate to weights and a contextual graph.

3.6 Conclusion

This chapter examined the formal foundations of reason-based detachment functions introduced by Knoks and Van der Torre (2023) and their role in normative reasoning. Building on the dual-scale model of Tucker (2025), a framework was specified to distinguish between justifying and requiring weights, enabling richer representations of permissions and obligations. A family of detachment functions was defined, including additive, maximum, multiplicative, and satisficing variants, as well as a dynamic-scale operator for all-things-considered evaluations.

To assess these models, nine principles—such as Groundedness, Input Symmetry, and Sensitivity—were formulated and applied in a principle-based analysis. The results show that dual-scale detachment offers greater expressive power than single-scale models, capturing dilemmas and weak permissions while relaxing constraints like Unanimity and One Permission. The formalism allows a single scale to be achieved by constraining the dual scale with single-proportion weights.

These findings clarify the trade-offs between simplicity and expressiveness in normative reasoning frameworks. They also provide a foundation for implementing detachment functions in practical applications, which is the focus of the next chapter. The implementation will use a simplified version of the balancing operation introduced and will also extend the system to fit the application domain in a particular way. In particular, the dynamic scale will be beyond the scope of the current implementation due to its additional cost, and additional valuations will be considered. The next chapter will add aggregation of multiple agents and their weight functions. This introduces some problems, most of which are investigated in Chapter 4. However, a principle-based analysis of that process remains as future work. Such an analysis can reveal which principles change their proof status under those multi-agent conditions.

Chapter 4

Smart Contract Implementation

This chapter describes the integration of reason-based detachment into IHiBO (short for Intelligent Human-input-based Blockchain Oracle), a blockchain-based oracle designed for trustworthy decision-making. The implementation is based on the theory of the previous chapter. However, it is mainly informed by ideas from fund management and takes computational costs into account. It is therefore not an exact implementation of (dynamic) dual scale. The implementation uses the Solidity smart contract format to embed weighted reasons and balancing operators into portfolio management workflows. A comparative evaluation of computational efficiency is conducted against abstract argumentation-based smart contracts. The implementation is a proof of concept and still requires several steps before it can be used effectively in practice. The results show that the additive balancing operator exhibits linear growth in gas cost, in contrast to the exponential growth of an argumentation-based approach. The chapter demonstrates that numerical balancing is more scalable for blockchain deployment while maintaining transparency and auditability. Furthermore, Ethereum blockchain smart contracts allow users to agree on the usage of a particular smart contract specification. All participants can read the code before the interaction, which improves trust among participants.

4.1 Introduction

This chapter concerns the implementation of numerical balancing of weighted reasons as a smart contract for IHiBO. IHiBO is a recently developed application introduced by Yu et al. (2021) to improve the trust relationship between investors and their portfolio managers. IHiBO combines blockchain technology with abstract argumentation theory to simulate and document fund managers' decisions, providing investors with concrete guarantees about what is documented, enabling them to access a record of the decisions if they have concerns about the investments. The functions discussed in the previous chapter may be well-suited to this application. A smart contract is developed to implement numerical balancing of reasons for IHiBO. After implementation, the balancing reasons approach can be compared with the abstract argumentation approach.

At a time when decision-making processes increasingly require transparency, security, and accountability, blockchain technology has emerged as a potential transformative solution (Manolache et al., 2022). The decentralized nature of the blockchain ensures that decisions made on it are transparent, private, and immutable, potentially enhancing trust among stakeholders. This is particularly relevant in adversarial and collaborative multi-agent systems where decision-making involves numerous participants with different interests, expertise, and reputations.

This chapter describes the implementation of the theory of balancing reasons presented in the previous chapter in an IHiBO smart contract adapted for trust in fund management. This work was presented in (de Wit, Yu, et al., 2025) in which a smart contract was developed for the IHiBO (Intelligent Human-input-based Blockchain Oracle) architecture, a technology introduced by (Yu, Zichichi, Markovich, & Najjar, 2022; Yu, Zichichi, Réka, Amro, et al., 2022; Yu et al., 2021; Zichichi, 2021). IHiBO is implemented with fund management as the application domain, a domain in which compliance with regulations such as AIFMD[1] and UCITSD[2] require fund managers to handle both portfolio and risk management tasks on behalf of investors. Such funds can be managed by a team of three or more managers who conduct research, select suitable securities such as stocks or bonds, and execute

[1]Directive 2011/61/EU of the European Parliament and of the Council of 8 June 2011 on Alternative Investment Fund Managers (AIFMD) http://data.europa.eu/eli/dir/2011/61/oj

[2]Directive 2009/65/EC of the European Parliament and of the Council of 13 July 2009 on the coordination of laws, regulations, and administrative provisions relating to undertakings for collective investment in transferable securities (UCITSD). http://data.europa.eu/eli/dir/2009/65/oj

transactions aligned with the fund's strategy. The fund managers acting on behalf of the investors constitute a principal-agent problem in which the investors are the principals and the fund managers are the agents. In this context, the fund managers are called fiduciaries. The agent acts on behalf of the principal; the principal needs to trust the agent. The problems associated with this trust relationship are commonly analysed within the framework of the principal-agent problem. This work integrates reason-based balancing into IHiBO rather than relying on its abstract argumentation mechanisms, showing how blockchain-based decision-making can become more efficient, transparent, and accountable in complex financial environments.

The model was implemented as Solidity-based smart contracts by de Wit (2024) (see the code in Appendix B), enabling agents to provide explicit reasons for or against proposed actions. The system aggregates these reasons, each weighted by the agents' reputations and confidence in their reasons. The actions are then evaluated by comparing the total weight of positive reasons to that of negative reasons. This framework aligns with the intuitive notion that decisions are the result of weighing the pros (reasons in favor) against the cons (reasons against) and that they reflect underlying values and beliefs (Bratman, 1987; Tucker, 2025).

Although much of the scholarly work on reasons and normative decision-making has been conceptual, (Knoks & Van der Torre, 2023), and later (Knoks et al., 2024) and (de Wit, Knoks, & Van der Torre, 2025a), as described in the previous chapter, it has provided a formal structure for such a reason-based evaluation. Their notion of numerical balancing operators offers a direct way to aggregate weighted reasons. In this chapter, these formal foundations are used to show how a reason-based balancing model can be integrated into blockchain-based decision-making. In this chapter, the permission scale of a single option is placed at the center of the smart contract. Compared to the previous chapter, the permission scale considers the permission of a selected option in pairwise comparison with its negation. The two pans of this permission scale are understood as the positive weight pan and the negative weight pan. A reason supporting the option is placed on the positive pan and can be understood as a justifying weight. The reasons against the option are placed on the negative pan; this weight can be understood as a requiring weight. This way, the weights for the fund managers' positive and negative reasons are compared to determine the fund managers' position regarding the option under consideration. Furthermore, the outcome of the balancing is refined to better fit the fund management use case. Namely, the option is interpreted as a stock, the $+$ outcome is understood as buying stock, the $-$ outcome is understood as selling stock, and 0 is understood as holding (neither buying nor selling) stock.

In addition to presenting the smart contract implementation, a comparative analysis of computational costs is performed, contrasting the reason-based balancing approach against a benchmark approach that uses abstract argumentation; see the work of Baroni et al. (2018) for more on argumentation theory. The results show significant efficiency gains, suggesting that the method not only improves conceptual clarity but also reduces practical overhead costs.

The remainder of the chapter is organized as follows. Section 2 provides a background overview of IHiBO and the numerical balancing of reasons. Section 3 details how this numerical balancing approach can be embedded within smart contracts for the purpose of fund management. Section 4 presents the results of the benchmark test and compares the performance of the numerical balancing approach with the abstract argumentation approach. Section 5 discusses notable findings, and Section 6 then considers the implementation of Dynamic Dual Scale as presented in the previous chapter. Then, in Section 7, the theoretical aspects of implementing dual-scale detachment in IHiBO are considered. This results in section 8, which elaborates on agents, reasons, and context. Before closing, section 9 discusses some additional considerations. Finally, Section 10 discusses related work, and Section 11 concludes the chapter.

4.2 Background

This section presents an overview of the IHiBO architecture (Yu, Zichichi, Markovich, & Najjar, 2022; Yu, Zichichi, Réka, Amro, et al., 2022; Yu et al., 2021; Zichichi, 2021) and the theory of balancing normative reasons (de Wit, Knoks, & Van der Torre, 2025a; Knoks & Van der Torre, 2023; Knoks et al., 2024).

4.2.1 Intelligent Human-Input-Based Blockchain Oracle (IHiBO): an overview

The Intelligent Human-Input-Based Blockchain Oracle (IHiBO) architecture (Yu, Zichichi, Markovich, & Najjar, 2022; Yu, Zichichi, Réka, Amro, et al., 2022; Yu et al., 2021; Zichichi, 2021) is a computational approach that incorporates argumentation reasoning mechanisms within multi-agent systems. An important aspect is that the human-input-based oracle bridges the gap between blockchain and real-world data by allowing human experts to input information into the blockchain. IHiBO is intended primarily for use in fund management; managers present their arguments regarding the investment plan for stocks. Specifically, IHiBO utilizes multi-agent abstract argumentation frameworks to model decision-making processes, which are then implemented via smart contracts and stored on the blockchain. The integration of argumentation, reasoning and blockchain enhances the transparency and traceability of the decision-making process.

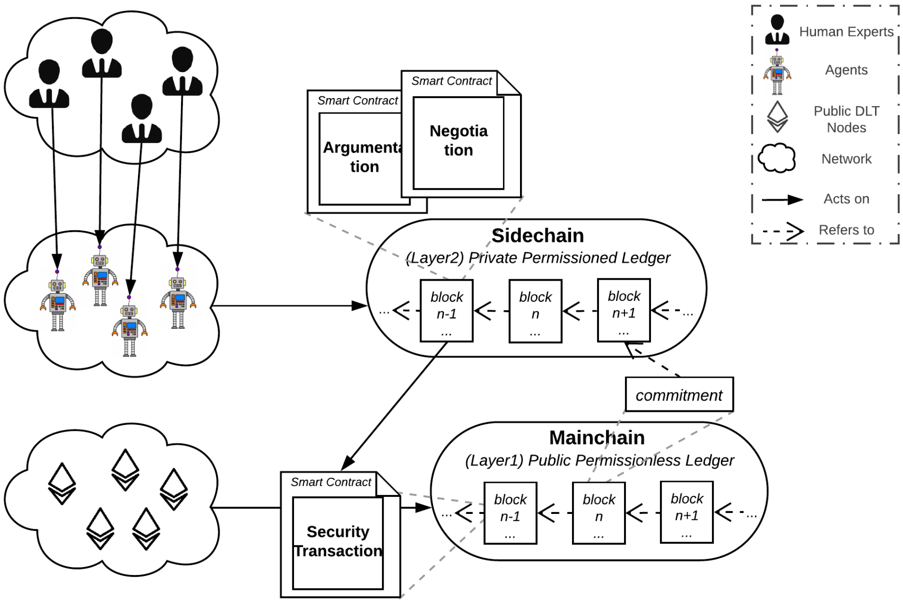


Figure 4.1: The IHiBO Architecture (Yu, 2023)

IHiBO leverages two-layer distributed ledger technology (DLT), as depicted in figure 4.1, to ensure data security and immutability while also maintaining efficiency and scalability. The secondary layer, a private permissioned blockchain accessible only to authorized users, hosts the smart contract for decision-making. This layer maintains privacy and reduces transaction costs, providing a balanced approach to data security and operational efficiency. The main layer is a public, permissionless blockchain, such as Ethereum, where the smart contract executing stock transactions is triggered by the output of the decision-making process. This two-layer design is particularly important in fund management, where decision-making can involve sensitive information. IHiBO's architecture allows for the specification of multiple smart contracts that simulate reasoning. Up to now, only multi-agent abstract argumentation (for collaboration) and negotiation (for adversarial settings) have been encoded in smart contracts. In this work, reason-based balancing is used to model the decision-making process, as will be explained in Section 2.2.

Argumentation Theory used in IHiBO. IHiBO uses abstract argumentation to model and then document portfolio managers' discussions before portfolio decisions are made. It does this by letting the portfolio managers vote on arguments; each vote adds a weight of 1 to the argument. Then, a conflict relation needs to be defined over the arguments to determine which arguments attack which other arguments. After the managers have voted and the attack relation has been specified, a so-called social reduction produces the final abstract argumentation graph. This graph defines a defeat relation on the arguments in which defending arguments are defeated only if they are not stronger than the attacking arguments. The strength of an argument is determined by the number of votes it has received. After the defeat relation has been determined, argumentation semantics is applied to determine which arguments are accepted. Arguments that are not attacked by any accepted argument are accepted. This is called a grounded extension. The accepted arguments then determine what actions the team of portfolio managers is allowed to take. IHiBO uses the preferred extension; the grounded extension is unique, but the preferred extension is not guaranteed to be unique. The preferred extension is the maximal admissible set of arguments (with respect to set inclusion). An admissible set is a set of arguments that defends itself against attacks from outside the set. See Figure 4.2 for an illustration of the procedure for converting a logical structure to an argumentation framework (graph). The graph is used to find argument extensions, such as the grounded or preferred extension. The chosen semantics instantiate a defeasible logic.

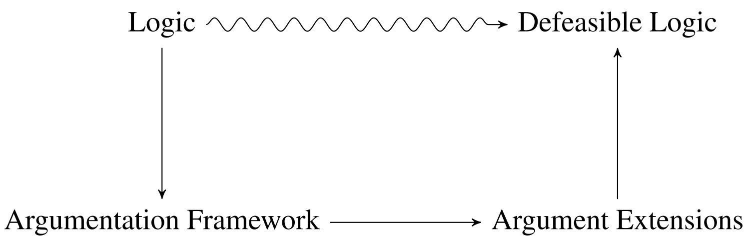


Figure 4.2: Defeasible Logic using Argumentation Frameworks

The IHiBO abstract argumentation smart contract proceeds through multiple steps to generate an output. Initially, fund managers introduce arguments and strengthen arguments; this builds an argumentation framework. They are voting on arguments they acknowledge. Then the attack relation between the arguments is determined. This can be implemented in multiple ways. Fund managers could introduce the attacks, or a third party could introduce them. In structured argumentation, the attack relation is instantiated through the internal logic of the arguments. However, in the abstract approach, that is not automatically possible, and so a different approach is needed in order to specify the attack relation. So far, how this relation is exactly instantiated has not been discussed by Yu et al. (2021).

Definition 25 (Abstract Argumentation Framework)**.** *An abstract argumentation framework (AAF) is a pair* (A, R) *where* A *is a set called arguments, and* $R \subseteq A \times A$ *is a binary relation over* A *called attack.*

An abstract argumentation framework can be extended with a preference relation. The preference relation is defined on the arguments and allows those arguments to be compared. This preference relation can be used to express, for example, the strength of an argument. In IHiBO, the preference is determined by the number of managers who acknowledge an argument.

Definition 26 (Preference-based AAF)**.** *A preference-based abstract argumentation framework (PAF) is a 3-tuple* $(A, R, \succ)$ *where* A *is a set of arguments,* $R \subseteq A \times A$ *is a binary relation called attack, and* $\succ$ *is a partial order (irreflexive and transitive) over* A *called a preference relation.*

A preference-based argumentation framework can be transformed into an abstract argumentation framework. To do this, the attack relation can be transformed into a defeat relation using the preference relation. For this, four so-called preference reductions have been introduced (Amgoud & Vesic, 2014; Van der Torre & Vesic, 2018). These reductions take into account the preference relation and the attack relation to determine whether one argument can defeat another.

Definition 27 (Preference Reduction (PR))**.** *Given a* $PAF = (A, R, \succ)$*, a preference reduction can be performed to reduce* PAF *into* AAF*. Write* $PR_i(PAF) = (A, R')$*, where*

1. $R' = \{(a, b) \in R \mid b \not\succ a\}$

2. $R' = \{(a,b) \in R \mid b \not\succ a\} \cup \{(a,b) \notin R \mid (b,a) \in R, a \succ b\}$

3. $R' = \{(a,b) \in R \mid b \not\succ a\} \cup \{(a,b) \in R \mid (b,a) \notin R\}$

4. $R' = \{(a,b) \in R \mid b \not\succ a\} \cup \{(a,b) \notin R \mid (b,a) \in R, a \succ b\} \cup \{(a,b) \in R \mid (b,a) \notin R\}$

The preference relation can be instantiated in multiple ways. One way is for each agent in a collective to acknowledge the arguments they like, then count how many agents acknowledged each argument.

Definition 28 (Multi-Agent Argumentation Framework)**.** *A multi-agent argumentation framework (MAAF) is a 4-tuple* $(A, R, S, \sqsubset)$ *where just as in an AAF,* A *is a set of arguments,* $R \subseteq A \times A$ *is a binary relation over* A *called attack,* S *is a set of agents or sources, and* $\sqsubset \subseteq A \times S$ *is a binary relation associating arguments with agents. Then* $A_\alpha = \{a \in A \mid a \sqsubset \alpha\}$ *for all arguments that belong to agent* α*, and* $S_a = \{\alpha \in S \mid a \sqsubset \alpha\}$ *for all the agents that have argument* a*.*

When an agent acknowledges an argument, it is understood as one vote for that argument. The preference ordering is then instantiated by the number of votes. Here, the arguments with the most votes are the most preferred, and the arguments with the fewest votes are the least preferred.

Definition 29 (Social Reduction (SR))**.** *Given a* $MAAF = (A, R, S, \sqsubset)$*, a social reduction can be performed to reduce the* $MAAF$ *into a* PAF*. Write* $SR(MAAF) = (A, R, \succ)$ *with* $\succ = \{(a,b) \mid \|S_a\| > \|S_b\|\}$.

One can then transform a multi-agent argumentation framework into an abstract argumentation framework by chaining the two reductions as one reduction in the following way:

Definition 30 (Reduction (Red))**.** *Given a* $MAAF = (A, R, S, \sqsubset)$*,*
write $Red_i(MAAF) = PR_i(SR(MAAF))$.

Abstract Argumentation Semantics. An abstract argument graph consisting of arguments and a defeat relation can be subjected to argumentation semantics. The argumentation semantics chosen and implemented in IHiBO is "preferred" semantics. This semantics labels an argument "IN" if it is not defeated by any other argument, or if all the arguments that defeat it are labeled "OUT". An argument is labeled "OUT" if it is defeated by an argument labeled "IN". An argument is labeled "UNDECIDED" if it cannot be labeled using the given specification.

Definition 31 (Admissibility)**.** *Let (A, R) be an AAF. Then a set $E \subseteq A$ is said to be **conflict-free** if and only if there are no arguments $a, b \in E$ such that $(a, b) \in R$ or $(b, a) \in R$. A set $E \subseteq A$ is said to **defend** an argument $c \in A$ if and only if for all arguments $b \in A$ such that $(b, c) \in R$, there is an argument $a \in E$ such that $(a, b) \in R$. A set $E \subseteq A$ is said to be **admissible** if and only if E is conflict-free and defends all the arguments in E.*

Abstract argumentation started with an article by Dung (1995). The definitions of abstract argumentation and admissibility in this subsection regarding abstract argumentation come from that article. Also the following types of semantics are specified in it.

Definition 32 (Dung Semantics)**.** *A Dung semantics is a function σ that associates an argumentation framework $AAF = (A, R)$ with a set of subsets of A; the elements of $\sigma(AAF)$ are called extensions.*

Dung (1995) continues by introducing multiple extensions, which are listed below.

Definition 33 (Extensions)**.** *Let (A, R) be an AAF. Let $E \subseteq A$ be a:*

- ***complete** extension if and only if it is admissible and it contains all the arguments it defends, i.e., $E = \{a \mid E \ \texttt{defends} \ a\}$.*
- ***grounded** extension if and only if it is the smallest complete extension (for set inclusion).*
- ***preferred** extension if and only if it is the largest complete extension (for set inclusion).*
- ***stable** extension if and only if it is conflict-free and attacks each argument that does not belong to E.*

There are many more semantics that can be defined; however, preferred semantics is a reasonable baseline. The designers of IHiBO gave no reason for choosing this semantics, but the semantics is generally appreciated for maximizing the number of "IN"-labeled arguments. The preferred semantics is non-unique, meaning it can produce multiple labelings. This can be seen as a downside, as each labeling could support different decisions. Therefore, one would need another decision function to determine a choice once the preferred extensions have been identified.

One strength of this approach is that the arguments are abstract and can represent many things as long as the conflicts between them are modeled correctly. However, correctly modeling conflicts is not necessarily easy. The discussion can be limited to one stock or expanded to multiple stocks without changing the setup. As stated previously, one downside of the approach is that the attack (or conflict) relation is difficult to implement. Various methods can be considered; however, this relation strongly influences the outcome, so it is important to have a "good" method for it. Also, arguments that contain decisions or actions are not handled differently than other arguments. This means it is theoretically possible to end up with an extension that contains no decision.

4.2.2 Balancing normative reasons

The definitions introduced by de Wit, Knoks, and Van der Torre (2025a), Knoks and Van der Torre (2023), and Knoks et al. (2024) are used to describe the numerical balancing of reasons. Although Knoks et al. considered three distinct values $(+, -, 0)$, their approach is extended to include five values in Section 4.3. In this setting, a normative reason is understood as a consideration that counts in favor of or against a particular issue—an issue being any matter requiring a decision about whether a certain action should be taken. The notation presented here is closer to the notation of Knoks and Van der Torre (2023), which includes reasons with explicit polarities. The notation represents context as a set of reasons together with an option. In the previous chapter, the context was abstract and merely referenced. In this chapter, this representation is adopted since we first implemented a basic version of balancing. The most basic version is to focus on a single balance scale and so a single option and relevant reasons. This also minimizes the amount of information required for documenting reducing the gas cost of the smart contract. Also, the new smart contract for numerical balancing, IHiBO-NB, focuses on a single option and the set of reasons concerning it, thereby aligning with the preceding description. In this chapter, we briefly touch on the difference between this IHiBO-NB implementation and the dynamic dual scale, and we will discuss this topic further in the next chapter.

Definition 34 (Reasons)**.** *Let $A = G \cup O$ be an infinite set, called the universe of discourse, where G is the set of grounds and O is the set of options. And let $V = \{+, -, 0\}$ be the set of possible values. A reason r is a triple (g, o, v) where $g \in G$ and $o \in O$, where g is the ground, o is the option, and $v \in V$ is the value (or polarity) of the reason.*

For the IHiBO implementation, a deviation from the theory is made to account for the case where only one option is considered. Therefore, a reason here has polarity. As seen in the previous chapter, reasons in general do not occur by themselves but occur as a group in the context of a particular set of options. One can understand such a context-crux pair as a case to be determined. In the practical context of IHiBO, it is sufficient to understand a context as a set of reasons and option; which may at some point be extended to a set of options. This is a simplification of the definitions in the previous chapter.

Definition 35 (Context)**.** *A context c is a pair of the form (R, o) where R is a finite set of reasons and o is an element of O called the option (or issue) of the context.*

Example 6 (Context construction)**.** *A team of fund managers considers the option of buying a particular stock "A" ($o =$"stock A"). To determine whether they want to increase or decrease their stock of asset A, they create a context by gathering multiple grounds: $x_1 =$"Asset A is increasing in value", $x_2 =$"Asset A is decreasing in value", $x_3 =$"Asset A will maintain its value", $x_4 =$"The market is bullish", $x_5 =$"The market is bearish".*

These grounds give rise to the following reasons:

$$r_1 = (x_1, o, +), \quad r_2 = (x_1, o, -), \quad r_3 = (x_2, o, +), \quad r_4 = (x_2, o, -),$$

$$r_5 = (x_3, o, +), \quad r_6 = (x_3, o, -), \quad r_7 = (x_4, o, +), \quad r_8 = (x_4, o, -),$$

$$r_9 = (x_5, o, +), \quad r_{10} = (x_5, o, -)$$

Each fund manager can select a context consisting of those reasons that they consider relevant, for example, $c = (\{r_1, r_5, r_7, r_{10}\}, o)$.

To determine the balance of reasons of different magnitudes in a particular context, the reasons need to be assigned a weight. For this Knoks et al. (2024) defined the notion of a weight function that assigns a weight to each reason in a specific context. In the previous chapter, the dual weight approach was introduced. In this chapter, a version of detachment closer to the one presented by de Wit, Yu, et al. (2025) and Knoks and Van der Torre (2023) is used. This is mainly a different way of packaging certain inputs. Instead of having a set of reasons to choose from, the context is understood as containing the reasons and the option to be evaluated. Also, in the work of de Wit, Knoks, and Van der Torre (2025a), reasons are understood as occurring in the context, even though the context is left abstract.

Definition 36 (Weight Function)**.** *Let C be the set of all contexts; then for each $(R, o) = c \in C$ there is a function w_c that for each $r \in R$ assigns a weight to the reason $w_c(r) \in \mathbb{R}_{\geq 0}$.*

Note that this appears to deviate from the dual-weight approach. However, on closer inspection, it turns out to be close to dual scale comparing o with its negation $\neg o$. Since the reasons now have polarity, justifying weight can be understood as the weight of $+$-polarity reasons, and requiring weight of alternative options as that of $-$- polarity reasons. While these representations are not completely equivalent, they suffice in the IHiBO context since the smart contract implementation will only balance reasons for and against a single option. That is, it will determine the permissibility of only one option, all things considered.

The main relation considered by Knoks et al. (2024) is called a balancing operation, which is a subclass of the relations defined by Knoks and Van der Torre (2023). This relation relates a context to a value. This operation allows an issue to be evaluated in its context. These are the detachment functions from the previous chapter.

Definition 37 (Detachment function)**.** *Let C be the set of all contexts. A* detachment function*, denoted as D, is a functional relation between contexts and values:*

$$D \subseteq C \times V,$$

such that for any $(c, v), (c', v') \in D$, if $c = c'$, then $v = v'$.

As we saw in the previous chapter, a detachment function can be implemented using a weight function together with a balancing procedure. In the work of de Wit, Knoks, and Van der Torre (2025a) and Knoks et al. (2024), a procedure is introduced to assign a value to a context. This procedure is defined so that each context is assigned a value using a specific weight function.

Definition 38 (Instantiation)**.** *A detachment function can be instantiated by a weight function $D_w = D$.*

In the previous chapter, the default operator was the dual scale operation. However, in this chapter, a slightly modified operator will be developed using the additive permission scale. The specification of the permission scale is provided below.

Definition 39 (Additive Permission Scale)**.** *Let D_w be a detachment function, and let $c = (R, o)$ be a context. The detachment function is defined by:*

$$neg(c) = \{r \in R : r = (g, o, -)\}, \tag{4.1}$$

$$pos(c) = \{r \in R : r = (g, o, +)\}. \tag{4.2}$$

Then, the additive operator assigns a value based on the sum of weights:

$$D_w(c = (R, o)) = \begin{cases} + & \text{if } \sum_{r \in pos(c)} w(r) > \sum_{r \in neg(c)} w(r) \\ - & \text{if } \sum_{r \in pos(c)} w(r) < \sum_{r \in neg(c)} w(r) \\ 0 & \text{otherwise} \end{cases}$$

A detachment function using this balancing operator is called an additive detachment function.

Example 7 (Decision Production)**.** *Following Example 1, fund managers use the additive balancing procedure to determine a polarity in a context: $c = (\{r_1, r_5, r_7, r_{10}\}, o)$ with reasons:*

$$r_1 = (g_1, o, +), \quad r_5 = (g_3, o, +), \quad r_7 = (g_4, o, +), \quad r_{10} = (g_5, o, -)$$

The weight function is set to the following weights:

$$w(r_1) = 5, \quad w(r_5) = 6, \quad w(r_7) = 20, \quad w(r_{10}) = 21$$

The sums are calculated as follows:

$$pos = \sum_{r \in pos(c)} w(r) = 5 + 6 + 20 = 31 \quad (4.3)$$

$$neg = \sum_{r \in neg(c)} w(r) = 21 \quad (4.4)$$

Since $pos = 31 > neg = 21$*, the procedure assigns* $D_w(c) = +$*. Thus,* $(c, +) \in D_w$*.*

This does not immediately translate to the definitions of Chapter 3 since the detachment functions defined there yield two values and a more complex contextualization. However, a single permission scale can perform a dynamic dual scale, provided the correct weights are assigned to the pos and neg scales. To see this, recall the dynamic detachment function and the dual-scale detachment function. The dynamic dual scale for a single option detaches $+$ when the option never loses a comparison (i.e., it never gets a $-$), and detaches $-$ when it loses at least one comparison (i.e., it gets a $-$). Therefore, the pairwise comparison of an option o with its negation is about whether there exists an option that defeats o. This is a single comparison in which either the positive weight is outweighed by the negative weight or the justification for o is outweighed by the requiring weight of $\neg o$. The next section shows that these definitions are sufficient to understand the protocol to be implemented for IHiBO.

4.3 Integrating reason-based balancing into fund management

In fund management, decision-making typically occurs in three stages: planning, execution, and feedback (Maginn et al., 2007). Although IHiBO focuses on the execution phase, the underlying reasons for decisions can originate from any stage. The process proceeds as follows: after the smart contract is initialized and the issue (e.g., “Apple stock”) is set, managers propose reasons for increasing or decreasing the stock. Each reason is assigned one of five polarities: $--$ (strong sell), $-$ (weak sell), 0 (hold), $+$ (weak buy), or $++$ (strong buy). These default positions on a stock are adopted because they are common on advice platforms. These advice platforms are a public place where advisors can share their advise on buy or selling a particular stock. As such, advice from such platforms can be directly incorporated into the decision process, generating new advice that another smart contract connected to IHiBO-NB can then act upon. The exact difference between a weak and a strong sell is implementation-dependent. It may correspond to a percentage of stocks or liquidity. Alternatively, a weak buy/sell could be tied to a single unit while a strong sell might involve selling all units of a stock and a strong buy using all available liquidity to purchase them. By encoding the decision as a polarity, fewer options need to be considered. Rather than comparing different quantities of stock as options, a direct decision can be made on whether to buy or sell. Fund managers express their votes on these reasons using weights that reflect their level of confidence. The influence of each vote depends on two parameters: the manager's reputation and their confidence in the given reason. Higher reputation and confidence yield greater weight, ensuring that more trusted opinions have a stronger impact. The weighted reasons are then balanced, using a predefined procedure, in order to reach a decision, and the decision is recorded on the blockchain to ensure transparency and traceability.

To formalize how votes are translated into numerical weights, voting parameters are defined. Reputation captures a manager's trustworthiness and domain expertise, whereas confidence reflects the degree to which a manager believes a given reason. By multiplying reputation and confidence, the vote's magnitude is calculated, determining how much the manager's vote contributes to the total weight of a reason.

Definition 40 (Voting Parameters)**.** *Let I be a set of managers. The reputation of a manager $\alpha \in I$ on the subject of issue o, denoted as $rep(\alpha, o)$, is a non-negative integer. The confidence of the manager α in a reason r regarding issue o, denoted as $conf(\alpha, r, o)$, is also a non-negative integer. The magnitude of a vote, $m(\alpha, r, o)$, is defined as $m(\alpha, r, o) = rep(\alpha, o) \times conf(\alpha, r, o)$.*

A vote is then defined using these parameters and the concept of a reason.

Definition 41 (Voting)**.** *Given a set of managers I, a manager $\alpha \in I$ votes on a reason $r = (g, o, v)$ by specifying their confidence $conf(\alpha, r, o)$ in that reason in relation to issue o. Additionally, a separate entity manages the reputation of every manager in I with regard to issue o , assigning $rep(\alpha, o)$ to each $\alpha \in I$. The weight of reason r in context $c = (R, o)$ is then calculated as $w((r, (R, o))) = \sum_{\alpha \in I} m(\alpha, r, o) = \sum_{\alpha \in I} rep(\alpha, o) \times conf(\alpha, r, o)$.*

Example 8 (Weight Function Initialization)**.** *Consider the context $c = (\{r_1, r_5, r_7, r_{10}\}, o)$. The set* *of reason-context pairs is* $\mathcal{X}_{\{c\}} = \{(r_1, c), (r_5, c), (r_7, c), (r_{10}, c)\}$*. Assume there are three managers:* α*,* β*, and* γ*, with reputations* $rep(\alpha, o) = 1$*,* $rep(\beta, o) = 2$*, and* $rep(\gamma, o) = 3$*, respectively. The managers cast votes with the following confidence levels:*

$$conf(\alpha, r_1, o) = 3, \quad conf(\alpha, r_5, o) = 3, \quad conf(\alpha, r_7, o) = 3$$

$$conf(\alpha, r_{10}, o) = 3, \quad conf(\beta, r_1, o) = 1, \quad conf(\beta, r_5, o) = 2$$

$$conf(\beta, r_7, o) = 1, \quad conf(\beta, r_{10}, o) = 2, \quad conf(\gamma, r_1, o) = 0$$

$$conf(\gamma, r_5, o) = 0, \quad conf(\gamma, r_7, o) = 5, \quad conf(\gamma, r_{10}, o) = 5$$

The resulting contributions are summed to produce the weight of each reason:

$$w((r_1, c)) = 1 \times 3 + 2 \times 1 + 3 \times 0 = 5$$

$$w((r_5, c)) = 1 \times 3 + 2 \times 2 + 3 \times 0 = 6$$

$$w((r_7, c)) = 1 \times 3 + 2 \times 1 + 3 \times 5 = 20$$

$$w((r_{10}, c)) = 1 \times 3 + 2 \times 2 + 3 \times 5 = 21$$

Thus, the weight function for this context is initialized as: $w_{\mathcal{C}=\{c\}} = w$*.*

Figure 4.3 illustrates the entire decision-making process. The managers α, β, and γ vote on the reasons, each one assigning confidence levels to the reasons. As indicated, the notations co_n^α and r_n^α are used to represent confidence and reasons respectively, where $n \in \mathbb{N}$ and $\alpha \in \mathcal{I}$, since not every manager will vote on every reason due to various reasons. Possibly the manager does not know about a reason or disagrees with the reason or is not permitted to use the ground. The reputation of each manager on issue o is denoted by $rep(\alpha, o)$. After aggregating these inputs, the balancing operation is applied to produce $(co^*, (R, o, dec)) = (co^*, r^*)$. The asterisk in co^* distinguishes this *confidence* output from the *confidences* input. The notation is slightly abused by writing $(R, o, dec) = r^*$ to indicate that the output can be viewed as a new reason. This reason output captures the entire outcome of the decision and can be reused as input for subsequent decisions.

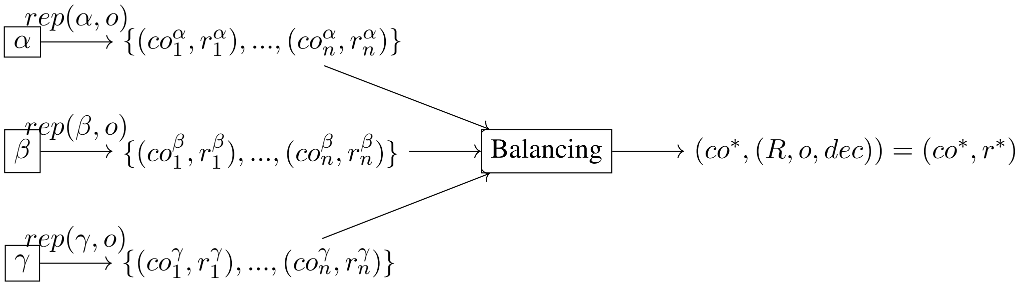


Figure 4.3: The whole process of decision-making regarding fund management

Algorithm 1 shows the adapted additive balancing procedure. After categorizing the reasons by polarity, the corresponding weights are summed. The weights of strongly positive or strongly negative reasons are doubled. Neutral reasons serve as a baseline that must be surpassed. If the aggregate weight of a particular polarity sufficiently outweighs the opposing polarity, the decision adopts that polarity. Otherwise, the procedure yields a neutral decision. The final result includes a confidence measure and the selected polarity, which are then recorded on the blockchain.

Example 9 (Balancing Procedure)**.** *Consider a context* $c = \big(\{r_1, r_5, r_7, r_{10}\}, o\big)$*, together with the following reasons and their weights:*

$$r_1 = (x_1, o, +), \quad r_5 = (x_3, o, +), \quad r_7 = (x_4, o, ++), \quad r_{10} = (x_5, o, --),$$

$$w\big((r_1, c)\big) = 5, \quad w\big((r_5, c)\big) = 6, \quad w\big((r_7, c)\big) = 20, \quad w\big((r_{10}, c)\big) = 21.$$

Categorization. *The reasons are grouped by their polarities as follows.*

$$\begin{aligned}
\textit{positive (+)} &= \{r_1, r_5\} && \textit{with total weight } pos = 5 + 6 = 11,\\
\textit{strongly positive (++)} &= \{r_7\} && \textit{with total weight } ppos = 20,\\
\textit{negative (-)} &= \varnothing && \textit{so } neg = 0,\\
\textit{strongly negative } (--) &= \{r_{10}\} && \textit{with total weight } nneg = 21,\\
\textit{neutral (0)} &= \varnothing && \textit{so } neut = 0.
\end{aligned}$$

Algorithm 1 Decision Procedure

Require: $\|Reasons\| > 0$
Ensure: $Issue$

$neut \leftarrow \sum_{r \in \{r \in Reasons | r=(x,Issue,0)\}} w(r,(Reasons,Issue))$
$pos \leftarrow \sum_{r \in \{r \in Reasons | r=(x,Issue,+)\}} w(r,(Reasons,Issue))$
$ppos \leftarrow \sum_{r \in \{r \in Reasons | r=(x,Issue,++)\}} w(r,(Reasons,Issue))$
$neg \leftarrow \sum_{r \in \{r \in Reasons | r=(x,Issue,-)\}} w(r,(Reasons,Issue))$
$nneg \leftarrow \sum_{r \in \{r \in Reasons | r=(x,Issue,--)\}} w(r,(Reasons,Issue))$
$sum \leftarrow (pos + 2 * ppos) - (neg + 2 * nneg)$
$magn \leftarrow |sum|$
if $!(2 * magn > neut)$ **then**
    $confidence \leftarrow neut$
    $dec \leftarrow 0$
else if $sum > 0$ && $ppos > pos$ **then**
    $confidence \leftarrow ppos$
    $dec \leftarrow ++$
else if $sum > 0$ && $ppos < pos$ **then**
    $confidence \leftarrow pos$
    $dec \leftarrow +$
else if $sum > 0$ && $nneg > neg$ **then**
    $confidence \leftarrow nneg$
    $dec \leftarrow --$
else if $sum > 0$ && $nneg < neg$ **then**
    $confidence \leftarrow neg$
    $dec \leftarrow -$
else
    $confidence \leftarrow 0$
    $dec \leftarrow 0$
end if
emit Output($(confidence,(Reasons,Issue,dec))$)
return $(confidence,(Reasons,Issue,dec))$

Balancing Computation. *Following the additive balancing operator (see Algorithm 1), the net "sum" of reasons is:* sum $= \big(pos + 2 \cdot ppos\big) \;-\; 2 \cdot nneg \;=\; 9$. *Since* sum > 0*, the positive side is favored. Moreover, because* $2 \times$ |sum| $> neut$*, the difference exceeds the neutral baseline. Next, to decide which positive polarity applies,* ppos *is compared with* pos*. Since* ppos $= 20 >$ pos $= 11$*, the operator selects a strongly positive outcome ++, and the confidence measure is* 20.

Outcome. *The procedure yields:*

$$\big(\, \textit{confidence}, \big(\{r_1, r_5, r_7, r_{10}\},\ i,\ {+}{+}\big)\big) \;=\; \big(20, \big(\{r_1, r_5, r_7, r_{10}\},\ i,\ {+}{+}\big)\big).$$

This outcome is recorded on the blockchain, ensuring that the final decision and its confidence measure are both transparent and traceable for subsequent evaluations. Although there is a strongly negative reason (r_{10} with weight 21), the combined effect of two positive reasons plus one strongly positive reason (++) tips the scales toward a strongly positive result. The additive balancing operator doubles the weight for strongly positive and strongly negative reasons. Ultimately, the net difference is 9 in favor of the strongly positive polarity, producing the polarity ++ with a confidence of 20*.*

4.4 Implementation and Evaluation

The objective is to integrate numerical balancing into the IHiBO architecture through a Solidity-based smart contract implementation (see the relevant GitHub pages of de Wit (2024) and Zichichi (2021)). To evaluate this blockchain-based approach, the gas costs incurred by executing transactions on the Ethereum network are measured; these costs reflect the computational cost of executing the smart contract. Although the focus is primarily on the gas cost comparison, a deeper investigation of how fund managers come to an agreement is beyond the scope of this work. However, the goal is to improve investors' trust in managers; therefore, it is not intended to stand in the way of managers' ability to express themselves and influence the process. The idea is that weighted reasons improve managers' expressivity, and that a balancing process can be selected that best suits them.

The current smart contract implementation can be understood as an extension of a balance scale to one that has five polarities. If the model is limited to 2 input polarities, the scale behaves as a normal scale. However, one may also view the implemented scale as a modification of the dual scale's permission scale. In both cases, the options are reinterpreted as issues. The full dual-scale model is less closely aligned with this implementation because comparison with alternative options is not included in that way, so comparativism and contrastivism fail. Dynamic-scale tournaments and full dual-scale was not implemented due to their complexity and computational constraints. Although this topic will be discussed in greater depth later, note again that the dynamic dual scale is relatively close when the 5 polarity scale is limited to two polarities and understood as a comparison between o and its negation $\neg o$. The work presented here is a proof of concept for reason-based balancing on a blockchain and enables comparison with abstract argumentation. Future work should focus on bridging this gap to enable dynamic-scale tournaments using dual-scale detachment.

Experiment Setup. The evaluation follows the benchmark experiment conducted by Yu, Zichichi, Réka, Amro, et al. (2022), which, among other things, measured the gas costs incurred by smart contracts implementing abstract argumentation. In the original experiment, a decision-making process was executed 60 times for four node counts (5, 10, 15, and 20) while varying the probability of creating attack relations (0.33, 0.5, and 0.66). Each run considered three fund managers, with a 25% probability that a manager would vote on a given argument.

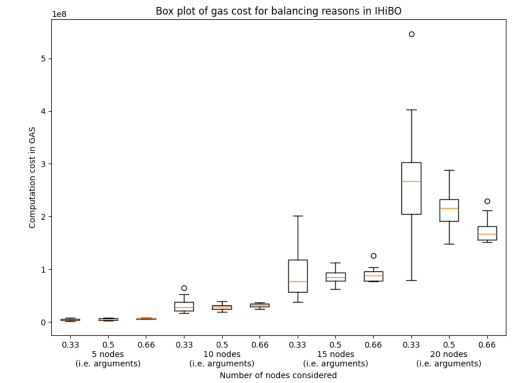
In the evaluation, the newly developed numerical balancing smart contract (NB) was compared to the abstract argumentation smart contract (AA). Since the reason-based balancing framework does not involve attack relations, the attack relation could not be instantiated for NB. Instead, in the experiment, 60 iterations were conducted for each of the same four amount of nodes: 5, 10, 15, and 20; under the NB setting. Each manager had a 65% chance of voting on a reason, and if they chose to vote, they randomly selected one of the five possible polarities: $--$, $-$, 0, $+$, or $++$. For the sake of simplicity, each manager could vote only one polarity per reason; i.e. they could not cast multiple votes for the same base reason (i.e. (g, o)) with different polarities. The managers always voted with confidence of 1, and reputation was also fixed at 1 for all managers, thus preventing these factors from influencing the results. These constraints allowed us to isolate the computational overhead introduced by the NB approach and to make a fair comparison with the AA benchmark.

4.4.1 Evaluation

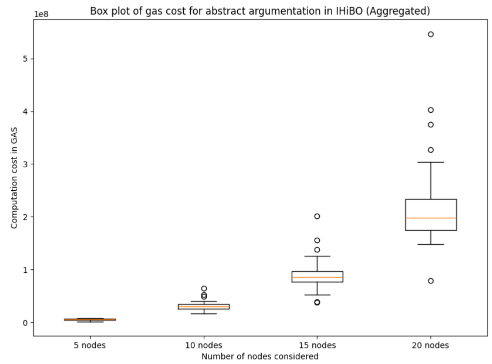
To evaluate the reason-based balancing smart contract, both qualitative and quantitative aspects should be considered. The qualitative comparison examines differences in modeling and their implications, whereas the quantitative comparison focuses on computational resource costs. The experimental results presented by Yu, Zichichi, Réka, Amro, et al. (2022) and de Wit, Yu, et al. (2025) are used to compare NB and AA under similar conditions.

Quantitative comparison

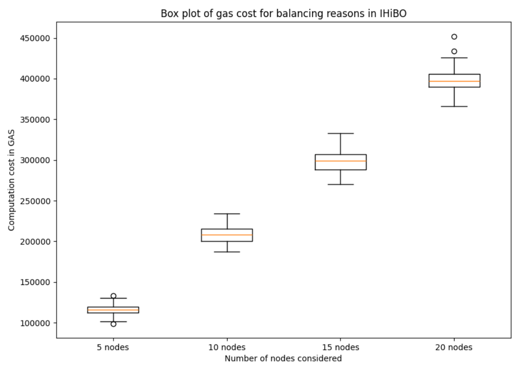
The results of the benchmark experiment are now presented. The AA results were taken from the work of Yu, Zichichi, Réka, Amro, et al. (2022), and the NB results were obtained from the experiment conducted by de Wit, Yu, et al. (2025).



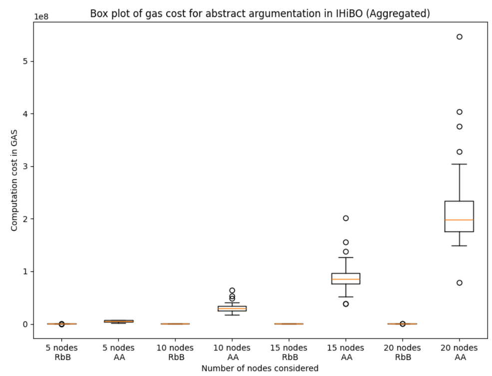
(a) Abstract Argumentation



(b) Abstract Argumentation (aggregated)



(c) Numerical Balancing



(d) Comparison of (4.4b) with (4.4c)

Figure 4.4: Boxplots of gas costs — experimental results

Figure 4.4 shows the gas costs for both AA and NB. Figure 4.4(a) presents the AA benchmark results, where gas costs increase as the number of nodes increase, and the costs vary more when the probability of generating a conflict relation is lower. Figure 4.4(b) displays the aggregated AA results over different probabilities of the attack relation, illustrating that gas costs increase exponentially as the number of nodes increases. Figure 4.4(c), by contrast, shows the benchmark results of NB, where gas costs grow linearly with the number of nodes. Finally, Figure 4.4(d) compares the aggregated AA results with the NB results, showing that NB is less expensive to startup and in growth. NB exhibits a linear growth in gas costs, which is in stark contrast to the exponential growth of AA.

Qualitative comparison

We now consider the qualitative differences between IHiBO-AA and IHiBO-NB. Although confidence and reputation parameters are newly introduced in NB, they could also be introduced to AA, so these parameters do not inherently create a modeling difference. Both models rely on weights to express preference, but they apply them differently. IHiBO-AA counts the managers who acknowledge an argument and uses that number as the argument's weight, whereas IHiBO-NB allows managers to assign a positive integer to each reason. The sum of these integers serves as that reason's weight from the perspective of the collective.

AA uses two data structures — nodes (arguments) and arrows (attack relations) — and employs algorithms such as social reduction and preferred semantics. The final output is a set of arguments that represent a preferred extension. NB, on the other hand, uses a single data structure (nodes as reasons) and an algorithm (the extended additive decision procedure). NB also introduces the concept of an "issue" or "option" that contains a possible choice, which AA does not explicitly include, although it could with some adaptations. NB outputs a new derived reason with a confidence value equal to the total weight of the chosen polarity together with a decision and the full set of reasons referenced in producing that decision. This output can serve directly as input to a future decision-making process. In AA, one would need to iterate over the accepted arguments and feed them back into a new process.

In the NB approach, the nodes are called reasons rather than arguments. In AA, arguments are often represented as $a = (p, c)$, where p is a premise and c is a conclusion; this can easily be extended to include polarity. In NB, each reason $r = (x, y, v)$ includes a ground x, an issue y, and a polarity v. By embedding polarity directly into each reason, NB eliminates the need to specify the conflict relation that AA requires. Instead, the additive balancing operation aggregates and compares weighted reasons to produce the outcome. Since the reasons carry polarity information, fund managers can express more directly how each reason relates to the issue under consideration. In the AA smart contract, managers do not specify conflict relations themselves. Ideally, this would be inferred from the argumentation structure induced by the arguments under consideration, but this is not currently implemented.

Focusing on a single "issue" at a time can limit the ability of NB to account for interdependencies between multiple "issues". For example, when two different stocks are being considered for purchase, buying one may make it more difficult to purchase the other asset due to limited funds. By contrast, AA can produce a set of accepted arguments (an extension), from which a decision is derived on how much of each stock to buy simultaneously. NB must either incorporate the other decision process as reasons, or treat both stocks as constraints, or allow multiple issues to be represented in a single discussion, or use one decision as an input to the reasoning of another issue. Careful structuring may be required to prevent contradictory outcomes given that NB does not check whether the grounds for different reasons are consistent with one another. In other words, if two stocks are considered, then two processes are needed. If only one stock can be purchased, the outputs of these two processes must be reconciled into a single consistent decision. Therefore, managers need to be careful when setting the weights for the polarities to ensure that the processes are aligned.

4.5 Discussion

The AA experiment involves more varied parameters because of the presence of conflict relations, whereas NB abstracts from these relations by incorporating polarity into the reasons themselves. This difference is likely the main factor in NB's reduced computational cost. It exchanges a complex graph analyzing algorithm for simple addition. The result is that AA requires probabilistic generation of conflict relations while NB requires probabilistic generation of polarities. Future research should clarify the exact implications of changing the underlying decision-making mechanism from preferred semantics in AA to additive balancing in NB. Also in this dissertation the idea of finding preferred semantics in the numerical balancing approach is considered (4.7). Some key questions include the following.

1. Can the NB weight function express the same types of relations between reasons as AA between arguments, e.g., conflict?
2. How do AA decisions relate to NB decisions and vice versa?

3. What is the difference in expressivity for fund managers who must provide reasons or arguments supporting their decisions? More fundamentally, what is the relationship between reasons and arguments?

4. What is the difference in trust created by the approaches, given NB's explicit representation of polarity and the issue under consideration?

5. Can the outputs of each approach be reused as input for subsequent decision-making processes (without loss of context)?

These questions suggest avenues for further exploration. For now, the results indicate that the NB implementation offers computational advantages and may provide greater clarity, whereas AA imposes fewer restrictions, albeit at the cost of increased computational complexity. This, however, is limited to comparing abstract argumentation to the implemented 5-value single-scale type model. The dynamic scale still needs to be compared with abstract argumentation and may be at least as expressive as abstract argumentation, and possibly more expressive in terms of modeling capacity. A dynamic dual-scale may be more expressive because it accounts for dual weights, and it is unclear how dual weights relate to abstract argumentation theory.

Argumentation in terms of dual scale. Using the weight-function constraints, it is possible to describe the circumstances under which a weight function implements an attack relation, as in argumentation theory. A function can be written so that users of abstract argumentation can still specify an attack. An attack on an argument can defeat the argument if the attacking argument is "IN" and the defending argument is not preferred over the attacker. However, it is not immediately obvious that this is possible in dual-scale detachment.

In the IHiBO argumentation theory, there are multiple steps. The first step is the social reduction from votes on arguments to a preference relation. The second step is the preference reduction that reduces the attack and preference relations to a defeat relation. The final step is the preferred semantics that maps the argumentation graph to the sets of acceptable arguments. To reinterpret the AA process in NB terms, the final step is performed by the balancing procedure. The first and second steps transform the votes into a representation from which the semantics can be computed. The votes should therefore be chosen so that a dual-scale system allows the balancing procedure to generate preferred semantics. To achieve this, the voting system should replicate the voting influence found in AA while at the same time encode the information of which elements defeat others in the process and how that influences the final outcome. In the dynamic scale system, an option becomes impermissible when its justifying weight is less than the requiring weight of an alternative option.

An attack, as specified in the principle of attack in Chapter 3, can be represented as follows.

Principle 10 (Attack)**.** *A weight function w encodes attacks when for all reasons r the weight is $w(r) = (0, 1)$. One can say that ground g attacks option o when $rw(o, o', (g, o')) = 1$, $jw(o, o', (g, o')) = 0$.*

This type of weight will ensure that the attacked option becomes impermissible when attacked. However, this encoding of attack does not implement the preference relation that (when evaluated under global semantics) allows an argument to survive an attack that cannot be defended locally. The reason-based system lacks any mechanism for defending against reasons since it only contains reasons that relate grounds to options. This limitation can be addressed by introducing a more complex structure in the reasons graph. An option can remain permissible only if its justifying weight is sufficiently high in relation to the requiring weight. A manager's support for the argument can be understood in terms of justifying weight. In the AA framework under consideration, an argument has greater support than its attackers. So, an attack in AA can be understood as corresponding to a requiring weight of 1 for an alternative option in NB. Similarly, support for an argument in AA corresponds to a justifying weight of 1 for that option in NB.

Since the argumentation implementation only represents arguments and does not distinguish between grounds and options, it should be possible to strengthen the position of an option without attacking another argument. This is achieved in the reason-based approach through justifying weight. Then a reason can be assigned to the option with weight $w(r) = (1, 0)$ thereby increasing the weight of the option in comparison with alternatives their requiring weight. This, however, requires careful specification of the dynamic scale's behavior.

Suppose there are at least two options. One of the options is attacked but also receives a supporting vote. This means that the requiring weight against it is 1, and the justifying weight is also 1; therefore, the permission scale will equate to 0. On the dynamic level, if the other values are not negative, then the outcome of the dynamic scale will be positive; however this only occurs if the output is restricted to positive and negative evaluations and 0 output or "weak" permission is not explicitly added. Perhaps that evaluation should in fact be $-$, especially if the alternative option also receives support with value 1, thus evaluating equal weight as forbidden. Furthermore, the attacks need to result in defeat, therefore, if balancing yields the value of 0, the outcome on the dynamic level should be $-$.

In simple cases, this leads to all grounds being treated as unattacked and thus accepted, since they are unattacked if the grounds are separate from the options. The grounds are understood as facts that are taken to be true. The options will be labeled "in" or "out", since these are the only possibilities on a dynamic scale where outcomes are evaluated as either $+$ or $-$. More research is needed, possibly via coding experiments and empirical testing. This can show whether the dynamic scale with the particular weight function yields preferred semantics. Alternatively, it could show whether the two systems are equivalent in terms of expressivity, or whether a function similar to that entailed by the principle describing the attack concept can be found and shown to be satisfactory.

Artificial Neural Network interpretation. The principle that implements the weight constraint for the principle of single-representatives is intended to capture how weights work in artificial neural networks. To fully go from multi-agent detachment to artificial neural networks, two modeling routes can be considered: either the entire artificial neural network is implemented within a detachment framework or the detachment function is implemented at the neuron level by means of IHiBO instances. In the latter case, the artificial neurons are implemented as instances of IHiBO, and a network is built from IHiBO-NB instances. To this end, additional work is needed to understand how to relate the output of IHiBO to the next layer, that of IHiBO-NB instances. Either approach will require additional work to fully show that the mapping between the systems is properly defined. However, the reasons function, as specified for the principle of single-representatives, closely resembles how a single layer of an artificial neural network operates. Also, IHiBO-NB's output format, as currently defined, is close to that of artificial neurons. These properties are sufficient to represent a full artificial neural network.

External Obligations. The purpose of IHiBO is to determine and document which options are collectively permissible or impermissible. However, it may be beyond the collective's power to execute a collectively permitted option. When the collective chooses to execute the permitted option, it may be prevented from doing so by factors outside its control. A prime example in the context of portfolio management is the market, which is governed by many variables arising from agent dynamics and other factors such as resource availability. When a collective attempts to execute a purchase, the purchase may be prevented due to price changes before completion. If there are insufficient funds to continue with the purchase, or if the goods are sold to another party, the chosen option cannot be executed.

Joint actions. In the specified application area of portfolio/fund management, the managers collectively manage a portfolio of funds through a voting procedure. However, in application areas where collaboration is less explicit i.e. implicit collaboration as a coincidence, the multi-agent detachment introduced by IHiBO may still be applicable. In such cases, the output of the dynamic dual scale, given the collective's specified weights, indicates which options the collective permits. However, the execution of any permitted option will still require the collaboration by the collective.

For example, a group of agents may decide to initiate a complex plan in which each agent must perform multiple actions, possibly involving risk. The full option thus consists of multiple actions that each agent must perform. In addition, if any agent fails to perform one of the assigned actions, the complex option will not be completed. This can have very serious consequences, especially when the risk to some agents depends on the completion or non-completion of actions assigned to other agents. Moreover, when agents assigned low-risk actions defect on the plan, the consequences for agents assigned high-risk actions are unfair.

A simple example of this is the traffic light system, which requires strict cooperation among all road users. If someone decides to drive through a red light, it can have dire consequences because other road users assume that everyone will follow the traffic light signals.

4.6 Towards Dynamic Dual Scale

The system presented implements a version of reason-based balancing for financial decision-making using the IHiBO architecture. However, the implementation differs from the dynamic dual-scale system presented in Chapter 3. Below, the differences are examined along with the role of implementation and experimentation within the framework. Finally, the system's limitations are analyzed.

4.6.1 Differences between NB and the Dynamic Dual Scale

The theory of balancing reasons was adapted to cater for a simplified use-case of portfolio management. Drawing on online platforms showing investor advice, multiple types of advice were identified: strong sell, weak sell, hold, weak buy, and strong buy. However, the theory of weighted reasons originally used a 2-polarity scale. To enable voting on additional polarities that reasons can take toward an option, three new polarities were introduced: $--$, $++$, and 0.

A more tangible difference is the interpretation of polarities as options, the absence of dual weights, and the absence of a dynamic scale derived from pairwise comparisons. To some extent, justifying and requiring weights are still present. The $+$ polarity makes an option permissible like justifying weight, and the $-$ polarity makes that option impermissible like requiring weight. What is missing, though, is the pairwise comparison, since the five options are directly compared on a graded scale. However, the polarities are not interpreted as permissible and impermissible but rather as graded buying and selling options, which is quite different.

The options allowed for each stock are reduced to 5 different evaluative actions on that stock, which are interpreted as polarities on a discrete evaluation scale. Even though the size of the option space was reduced, one could still, in principle, buy any amount of stock. The option hold is clear, as is the distinction between buying and selling. However, the precise distinction between a strong sell and a weak sell remains less clear. In portfolio management, the portfolio as a whole is more likely to be considered, with sales and purchases used to reshape it into a different portfolio. As such, a single-stock implementation serves solely as a proof of concept for balancing weighted reasons on the blockchain.

The output is also different. Yes, it states the output polarity for the stock. However, the dynamic dual scale outputs the evaluative status of each possible option. Additionally, the implementation provides an additional output that functions as a confidence metric for the output. That number is generated from the weight of the option weight relative to the weight of the other available options. This is also not part of the dual scale balancing theory.

Agency and voting to find a mutual permitted decision are both important in IHiBO and were not part of the dual scale theory. This can be introduced via definitions in the dual-scale detachment function. To incorporate voting on reasons and weights, multiple factors such as reputation and agency, which do not exist in the theory, are introduced into the dual-scale detachment function, albeit implicitly. Now, specific reasons aligned with different agents can be included in the same detachment process, which was not the case before. Before, a ground was a reason for only one option in a pairwise comparison. Now, the system allows a ground g to be a reason for two options endorsed by two different agents. This does not necessarily affect the outcome — if the ground is a stronger supporting reason for the first option than for the second option, the weight of the ground for the second option is dominated by that for the first option ($jw_{o_1} \geq rw_{o_2}$ and $rw_{o_1} \geq jw_{o_2}$). However, it does introduce a new type of reason — a reason that justifies both options, i.e., $jw_{o_1} > rw_{o_2}$ and $rw_{o_1} < jw_{o_2}$.

4.6.2 Testing through trading in a simulated trading environment

To test dynamic dual-scale detachment, including the multi-agent voting aspect introduced through IHiBO, it is useful to develop a software implementation of the system. This is easier to do in a language outside the blockchain environment such as Python. In Python, agents can be simulated in an agent-based trading environment where they interact with stocks. By adding different agent types, clusters of prototypical agents can be created that evaluate grounds with different weightings. The agents may be instantiated with different investment styles: risk-averse, risk-seeking, and balanced.

This can be done; however, the scope of the experiment is important. The financial market is highly complex, and considering only a subpart means removing variables. Having a simpler system than the real financial market introduces a type of error. Therefore, the initial focus may be placed on a single stock, and real historical data (e.g., Apple stock) can be used for the experiment. However, this would mean that the interaction between the simulated agents' buying and selling decisions does not directly affect the price. Adding this dynamic further complicates the coding of the experimental environment. Another challenge concerns the space of options. To keep things simple, the experiment may simply consider buying, or selling one unit, or holding of the stock. However, in a real financial market, agents can invest all their money in stocks or sell all their existing holdings if they wish. The advantage of this simplification is that the set of options remains finite. This finite structure can be extended by adding a temporal dimension, allowing options to span multiple decision stages, for example first selling and then buying. Nevertheless, at each point in time, the agent will still either buy, sell, or hold stocks. This relates to maximalism versus minimalism concerning options.

Implementing a dynamic scale requires devising a method to determine the weights. The simplest way to assign weights is to use single-proportion weights that remain the same across pairwise comparisons; this reduces the number of parameters required to specify the weight function. The function should also allow a ground to be a reason for two different options in a pairwise comparison. This makes it easier to get started since fewer weights need to be assigned. Then, the set of options to be considered should start with o_1 : "buy 1", o_2 : "sell 1", o_3 : "hold 1", expanding to arbitrary amounts, and later to options that encompass full portfolios. Grounds need to be determined, and financial management software already provides many common financial metrics. Grounds can include, among others: price, available liquidity, moving average, Return on Equity, risk-adjusted return, volatility, standard deviation, Sharpe ratio, Sortino ratio, Calmar ratio, Treynor ratio, moving average crossover, relative strength index, and moving average convergence divergence. Since these are numerical functions, they need to be turned into propositions that can be true or false. For example, "`price` $\leq$ `available-liquidity`" should be true before this ground can be a reason to buy a particular stock. With these additional grounds, setting up a trading strategy can be time-consuming. Moreover, other factors can be taken into account, for example, the stock's sustainability. For this, a sustainability heuristic would be useful: the stocks can be compared using "`sustainability(stock A)` $>$ `sustainability(stock B)`", based on the chosen weights, which could be a deciding factor in the purchase of stock A over stock B.

Once grounds have been specified, they can become reasons supporting options. Here, the main question is whether the ground is a reason to buy, sell, or hold, especially in the case of dual weights and a dynamic scale, where, for any integer x, the option of buying x shares exists. When the purchase of x amount of shares is allowed, the system becomes much more complicated because many options become available, each one requiring its own set of weights. However, once we recognize whether a ground is a reason to buy, hold, or sell, the system becomes relatively simple. The weights determine which option is permitted, so they should indicate when an option is good and when it is bad. Of course, this also depends on the trading agent's style: some investors are risk-seeking, while others are risk-averse. To generate the set of voting manager agents, multiple prototypes can be created and, by adding variation, a large group of slightly different agents can be instantiated.

Once the agents have been instantiated, they can be configured for collective trading. However, each agent's detachment function also produces particular trade proposals. These individual and collective trades may be compared in terms of performance. A reputation mechanism can be implemented by increasing an agent's reputation when their vote contributes to a profitable outcome and decreasing it when their vote contributes to a loss (turning reputation into a reinforcement learning parameter). The difficulty is that what may seem like a good choice at the time may prove to be bad choice once outcomes are observed, and the reputation assignment system should account for this. By default, agents' initial reputation should be set to 1, and it may be advisable to ensure the reputation score is at least 1 to avoid muting a voice. However, if necessary, the reputation score can be set to 0 or, in principle, even to negative values. Experimentation is needed to test this. Furthermore, it may be worth considering imposing a maximum reputation, perhaps 10, so that an agent cannot maintain a long streak of bad choices after having earned a reputation of say a 1000. These parameters can be tuned during the experiment to identify suitable values, look at reinforcement learning techniques. The challenge is how to determine the evaluation's time range. It may be misleading to consider only the decision's immediate effect, since some decisions are good in the long term but not in the short term. Additionally, to keep the reputation scores bounded, the system can simply update reputation scores based on outcomes and then subtract, from all agents, the reputation score of the worst-performing agent less one. This preserves the relative differences in reputation while keeping the lowest reputation score at 1.

A first step was taken toward implementing a trading environment to test the dynamic dual scale and to compare the performance of the collective with that of individual agents. Figure 4.5 shows the results that can be obtained so far. The figure shows three graphs: Market Data and Decisions, Equity

Curves, and Equity After Expenses. The market data represents the value of a particular stock and is randomly generated before the experiment starts, so trading within the system has no influence on the stock value. We consider different types of agents classified in terms of behavior (conservative, balanced, aggressive) as well as a collective agent. Some of their decisions are depicted in the first graph. The conservative agent is less likely to trade, the aggressive agent likes to take risks, and the balanced agent exhibits moderate risk-taking behavior. The collective agent aggregates the weighted inputs of the three agents to generate trading decisions. One issue here is that, when aggregated, conservative and aggressive traders may offset each other, increasing the relative influence of the balanced trader. This means that the collective agent may exhibit tendencies similar to those of the balanced agent. The second graph shows the equity curves, where equity is the portfolio value, defined as: "$\texttt{cash} + \texttt{stock} * \texttt{currentStockValue}$". In this case, the collective primarily follows the aggressive trader because the other two agents stop trading relatively early and therefore have less influence. This also becomes clearer in the next graph, which depicts equity after trading costs. In this graph, each agent's expenses are subtracted from their equity. This shows that the two agents that stopped trading maintain a relatively high equity value, whereas the other two appear to struggle. The idea behind this graph was to identify why they stop trading. Since the market fluctuates, the timing of trades matters. When a stock is purchased at a particular time, a purchase cost is established, and money is spent to acquire the stock. The only way to recover that money is to sell at the same price or at a higher price. It appears that conservative and balanced traders are cost-averse, which seems reasonable. If all traders sold all their stock at the end, the equity shown in the equity graph would represent their remaining funds. To provide further detail about the simulation code, the 4 agents each start with the following portfolio:

```
{‘‘cash’’: 10000, ‘‘stock’’: 100, ‘‘cost\_basis’’: 10000}.
```

The actions they can perform are: buy 1 stock, sell 1 stock, and hold. The following style biases are used:

```
‘‘conservative’’: {‘‘Buy’’: 1/6, ‘‘Sell’’: 1/6, ‘‘Hold’’: 4/6},
‘‘balanced’’: {‘‘Buy’’: 1/3, ‘‘Sell’’: 1/3, ‘‘Hold’’: 1/3},
‘‘aggressive’’: {‘‘Buy’’: 5/12, ‘‘Sell’’: 5/12, ‘‘Hold’’: 1/6},
```

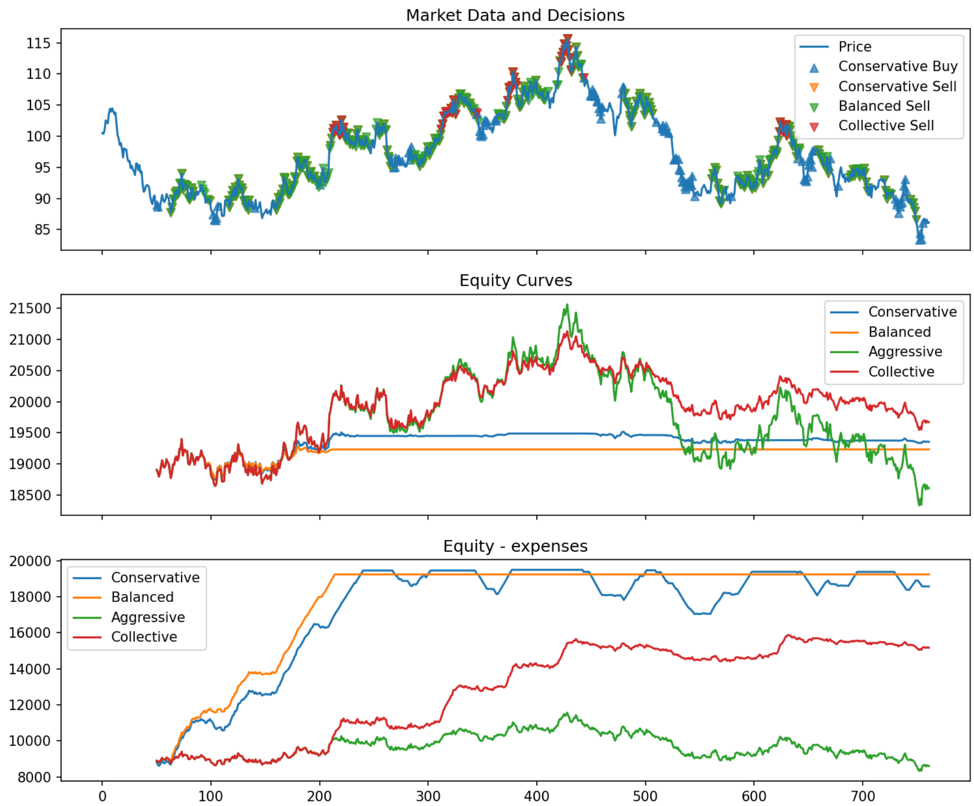


Figure 4.5: Simulated trading experiment

```
‘‘collective’’: {‘‘Buy’’: 0, ‘‘Sell’’: 0, ‘‘Hold’’: 0}.
```

The collective agent has no bias and is constructed solely from the other agents' input weights. Each agent has grounds that can be activated through market analysis. Once a ground is true, it may increase the weight of a particular option in a given comparison. This weight is then also given to the collective agent. It is likely that the less aggressive agents' grounds are not activated and so they don't provide input. If the aggressive agent only provides input to the collective agent, then the collective agent simulates the aggressive trader. The code can be viewed in Appendix B.2. Finally, the experimental timeline was divided into $50 + 365 * 2$ time points: 50 days for bootstrapping the system and initialising the grounds, 2 years for the simulation period. The grounds used are: MA-crossover(Moving Average), RSI-low(Relative Strength Index), RSI-high, MACD-positive(Moving Average Convergence/Divergence), MACD-negative, equity-gain-on-sell, and enough-liquidity-for-buy. The justifying and requiring weights each agent uses are pseudo-random. However, they are based on the style bias assigned to the agent. Each weight is a random number between 0.5 and 2, multiplied by the style bias.

4.6.3 Computational complexity of the dual scale compared with other approaches

The protocol's computational complexity is important. The gas cost tests show that the computational resource cost of the implemented scale model is much lower than that of abstract argumentation. However, how does the implemented balancing scale model compare with the dynamic dual scale in this regard?

If the number of grounds is $n \in \mathbb{N}$ and the number of options is $m \in \mathbb{N}$, then the maximum number of reasons is $n \cdot m = k \in \mathbb{N}$. In the balancing system implemented for IHiBO-NB with $m = 5$, there is a single balancing operation. In the dynamic scale, each option is pairwise compared with every other option, so for a given option, there are $m - 1$ pairwise comparisons. This gives a total of $m(m - 1)$ pairwise comparisons. Each pairwise comparison consists of 2 balance scales, so the total is $2 \cdot m \cdot (m - 1)$. However, if the weight function is symmetric, then the number of pairwise comparisons can be divided by 2 since o_1 vs o_2 and o_2 vs o_1 represent the same pairwise comparison, giving $m \cdot (m - 1)$. If a single proportion of weights is applied, the number of pairwise comparisons can again be divided by 2 since the dual scale reduces to a single scale. If both single proportion and symmetric weights are implemented removal of duplicate pairwise comparisons, then gives $\frac{m(m-1)}{2}$. However, in the implemented system IHiBO-NB, there is a single balancing process with $5n$ possible reasons. In the dynamic dual scale, given $m = 5$, there are $5n$ reasons and associated weights, and under a weight function that is symmetric and single proportion; then instead of 1 balancing process as in IHiBO-NB, there are $\frac{5 \cdot 4}{2} = 10$ pairwise comparisons.

4.7 Some properties of reasons used in the IHiBO balancing system

The implemented reason-based balancing serves as a proof of concept for balancing weighted reasons in IHiBO. Adding numbers is computationally cheaper than analyzing a graph, making the numerical approach feasible. Implementing a dynamic dual-scale is also feasible, though some constraints on the set of options may be required. However, weight allocation may introduce greater complexity compared to a single scale approach. IHiBO-NB can apply the theory introduced in the previous chapter while extending it with multi-agent voting. In this section, theoretical ideas introduced by IHiBO for detachment are specified using notation of detachment functions. The first idea introduced by IHiBO-NB is to combine separate weight functions into a single collective weight function, since each agent votes and has its own weight function.

Definition 42 (Adding weight functions)**.** *Given two weight functions w_1 and w_2 implementing detachment functions D_{w_1} and D_{w_2}, a joint weight function can be created by adding the weight functions, written as $w_* = w_1 + w_2$. Given a reason function R, for each o_1, o_2 and c, and for each r in the domain of w_1 and w_2:*

$$w_*(R, o_1, o_2, c, r) = w_1(R, o_1, o_2, c, r) + w_2(R, o_1, o_2, c, r)$$

Lemma 1. *Given two weight functions w_1 and w_2 and their detachment functions D_{w_1}, D_{w_2}, adding w_1 and w_2 allows the instantiation of another detachment function $D_{w_1+w_2} = D_w$.*

Proof. Although it is possible to combine each entry of the weight function, the resulting detachment function may fail under comparativism because a ground cannot be a reason for two competing options. The merger of two weight functions is therefore not guaranteed to generate a "legal" comparativist weight function.

However, weight functions that do not require comparativism can be found. □

The combination of weight functions is possible; however, the function R is pivotal. If the function R differs between agents, there may be a ground that favors two opposing options in the same pairwise comparisons, i.e., when two options o_1 and o_2 are compared, a ground g may be found such that $w_1(o_1, o_2, (g, o_1)) = (x, y)$ and $w_2(o_1, o_2, (g, o_2)) = (a, b)$. This can be understood as the two agents effectively exhibiting different weight functions. In certain situations, they may have different detachments, which may be partly due to each agent having a different R function. The main result of Alcaraz et al. (2024) is that the normative information is encoded primarily in the graph and not in the weights. When the reason function is the same, the definition of the weight function ensures that at least one of the weights has a non-zero value. In that case, the combination of the weight functions is immediate.

In IHiBO-NB, equality of R functions are not verified, and managers may vote freely on options using any reasons they wish to construct. However, it may be a good idea to enforce the constraint that no voter is associated with an R function under which grounds can support contrary options. In practice, this does not matter, since the weights cancel each other out. Such constraints will incur gas costs, as will adding reasons whose grounds support contrary options.

Definition 43 (IHiBO weight merger)**.** *Given $n \in \mathbb{N}$ and a let I be a set of n voting agents with weight functions $w_1, \ldots, w_n$, reason functions $R_1, \ldots, R_n$, and contexts $c_1, \ldots, c_n$. Denote their reputation factors as $rep_1, \ldots, rep_n$ and for each w we write that $w(R, r) = (0, 0)$ if $r \notin R(o, o')$, then the collective weight function is defined as follows:*

$$w^*(R^*, o_1, o_2, c^*, r) = \sum_{i=1}^{i=n} rep_i \cdot w_i(R_i, o_1, o_2, c_i, r)$$

where $R^* = \bigcup_i R_i$ and $c^* = \{c_1, \ldots, c_n\}$.

As this definition shows, IHiBO merges the weights, but does not merge the contexts. Also, since the reason functions are merged, in a pairwise comparison a ground can favor both options. However, because the weights push in opposite directions, they partially cancel each other out. This introduces the possibility of a reason having justifying weight for both options, or requiring weight for both options, and not being a reason for one or the other exclusively, but both.

Lemma 2 (Weight scaling)**.** *A "legal"(weight function definition abiding) weight function can be scaled to yield a new "legal" weight function.*

Proof. Scaling a weight function fully does not change the detachment function in the case of an additive dual scale. Given a detachment function D and a weight function w such that $D_w = D$, scaling the weight function by any scalar $\lambda \in \mathbb{R}$ results in the same detachment function. We write λw for the scaled weight function and then $D = D_w = D_{\lambda w}$. □

Another way to modify a detachment function is to directly update the detachments via a top-down detachment update.

Definition 44 (Detachment Update)**.** *Given a detachment function D, that detachment function can be changed with a set of update detachments $X = \{x_1 = (R, o_1, o_2, RR, c, v_1, v_2) \ldots, x_n\}$ as follows. For each $(R, o_1, o_2, RR, c, v_1, v_2) = x_1 \in X$, change the detachment function so that $D' = D$ whenever there is no (R, o_1, o_2, RR, c) such that there are v', v'' with $(R, o_1, o_2, RR, c, v', v'') \in X$, and whenever there is $(R, o_1, o_2, RR, c, v_1, v_2) = x \in X$, set $D'(R, o_1, o_2, RR, c) = (v_1, v_2)$.*

Lemma 3 (Guaranteed weight function)**.** *Given a detachment function D, there exists a weight function w such that $D_w = D$ under dual scale detachment.*

Proof. In the set RR of the specified comparison, the reasons always have at least one non-zero weight. This limits the possible output valuations. If there are only reasons for o_1 and not for the alternative o_2 then the following valuations are possible: $(+, 0)$, $(0, -)$, $(+, -)$, depending on whether their is justifying weight, requiring weight or both. The other valuations are not possible, since they would imply either that there is a weight that presumes a reason for o_2 or that the reasons have no weight.

Therefore, the proof fails, since there exists a detachment function D for which there is an input (R, o_1, o_2, RR, c) such that a particular valuation (v_1, v_2) cannot be achieved by a weight function w with $D = D_w$.

If there is at least one reason for each option, then there exists a weight function that generates each of the nine possible deontic output states. □

A deeper alignment between argumentation and detachment is possible. Argumentation uses a set of arguments and defines a directed graph over them. The reason function can be interpreted as a graph on elements; however, its domain and range are, by definition, separated. If they are not separated, a directed graph is obtained from the same class as abstract argumentation graphs.

Definition 45 (Reasons Graph)**.** *A reasons graph is a reasons function where the domain and range are the same set of elements A, called arguments (i.e, $A = G \cup O$).*

$$R : A \times A \rightarrow \mathcal{P}(A \times A)$$

The directed edges in an argumentation framework are interpreted as attacks, so within the weight function, they can only be requiring weights, not justifying weights. Since there tends to be now difference between attacks they should have the same weight say 1 requiring weight.

Definition 46 (Argumentation Weighting)**.** *A reason can be understood as an attack in abstract argumentation if it has polarity $-$ or without polarity has weight $w(r) = (0, 1)$ for an alternative. The requiring weight of* 1*, is placed on the requiring side of the permission scale of the option under consideration. Thus, if the reason is "in", the option is valued at $-$.*

Alternatively, this can be modeled without polarity; this requires multiple reasons, one for each alternative option using the same ground, with weights $w(r) = (0, 1)$. This also causes the option to lose every pairwise comparison. The ground "attacks" that particular option.

To apply basic abstract argumentation semantics, the status of the attacking arguments must be taken into account. An argument that is attacked by an unattacked argument cannot defeat the arguments it attacks. For this to work, the reasons graph itself needs to be used as the context of the weight. The influence of the defeated arguments must be nullified, the ground becomes "out" or 0, and this is only possible by it becoming ineligible for being a member of the set RR. Another possibility is to introduce a binary parameter analogous to reputation, which can be set to 0 or 1 depending on whether the argument is defeated.

Conjecture 1 (Reason-based preferred Argumentation Semantics)**.** *For each comparison, unattacked arguments will be evaluated* 0 *and thus labeled as permitted at the dynamic level. In that pairwise comparison, an argument that is attacked by an argument that is permitted has to at least overcome a requiring weight of 1; otherwise, it is assigned the value* $-$ *and thus labeled impermissible at the dynamic level. There is no justifying weight in normal AA as there are only attacks.*

Similarly, we can consider other classes of functions, for example, perceptron-style artificial neural network. These are also directed graphs with a weight function defined on them, together with an aggregation function at each node that determines whether the input exceeds the threshold condition. A single layer is simply a detachment function as specified previously.

Conjecture 2 (Convolutional layer)**.** *A convolutional layer consists of two sets of elements, inputs G and outputs O, where each input element is linked to each output element. Then, detachment is defined structurally as dual-scale with dynamic scale weighting. However, the weight function is subject to specific constraints, since the dynamic scale requires instantiation of multiple weights for each comparison. That is, the influence of an input* $g \in G$ *on an output* $o \in O$ *can be represented with a single number* $x \in \mathbb{R}$*, such that:*

$$D(R, RR, o, c) = sgn(\sum_{r \in \{(g,o) \in RR | g \in G\}} w((g, o))).$$

Similar to the principle of single representatives of the previous chapter.

Lemma 4 (Single number influence)**.** *The deontic influence of a ground* $g \in G$ *on an option* $o \in O$ *can be represented by a single number* $x \in \mathbb{R}$*.*

Proof. $D(o) = +$ if o never loses a pairwise comparison, i.e., if $\forall o' \in O \backslash \{o\}$ it is the case that $D(o, o') \neq (-, v)$.

Recall:

$$D(o, o') = (-, v)$$

if and only if $\sum_{(g,o)=r \in RR} jw((g, o)) - \sum_{(g,o')=r' \in RR} rw((g, o')) < 0$.

Assume $jw((g, o), o, o') = jw((g, o), o, o'')$ for all o', o'' and $rw((g, o'), o, o') = rw((g, o''), o, o'')$.

Then for $D(o, o') = (v_1, v_2)$ and $D(o, o'') = (v_3, v_4)$ it is true that $v_1 = v_3$.

Hence, we may write $w((g, o)) = jw((g, o)) - rw((g, o'))$. □

The detachment functions, as specified, can capture many systems. They differ in their specification of the reasons function R and its relation to the context c. It might also be possible to use the set of reasons functions SR, to specify the weight function in a particular way, i.e., by setting particular constraints.

Furthermore, as Knoks and Van der Torre (2023) showed, it is also possible to instantiate a consequence relation using detachment. This construction can be extended to the dynamic scale as follows.

Definition 47 (Dynamic-Dual-Scale Detachment-based consequence)**.** *For a detachment function D we write $(R, RR, c) \vdash o$ if and only if $D^*(R, RR, c, o) = +$.*

An interesting question is the correspondence between classical logical axioms and the principles satisfied by the weight functions that implement D^*. Maybe we may find a way to implement classical logics through this way for dynamic dual scale.

4.8 Expanding and contracting

Using the detachment specification and the multi-agent expansion, "IHiBO-DET" (implementing detachment relations) can accommodate multiple types of contraction and expansion. Content inside IHiBO-DET can be viewed as a frame or picture that can be expanded and contracted. As an example, consider a photo presented in the news that is said to lead to certain conclusions. Someone may claim that the photo is misleading because if it had a wider frame, the additional contextual content would undermine the supposed conclusions. Both sides make particular claims about what is not in the photo. The detachment theory allows reasons to be added to or removed from the set of reasons already considered. Expansion and contraction can occur at different levels. The context under consideration can be a sub-context or super-context of another context. Under IHiBO-NB, the collective of participating agents can be expanded or contracted.

Agents. IHiBO-DET allows agents to form a collective detachment on the options. Two types of interaction scenarios may be considered: one in which the agents agree on the context beforehand, and one in which the agents do not agree on the context. When the context is agreed upon beforehand, the exact weight assigned to each reason may vary across agents due to differences in their inputs. When they do not explicitly agree on the context beforehand, some agents may have different perspectives on the situation, so they may consider different reasons. From a collective perspective, the assembled information may be inconsistent. As the number of participating agents increases, the amount of inconsistency may increase due to the diversity of agent perspectives. Larger groups may thus contain "more" inconsistency.

Reasons. The detachment function specification in the previous chapter allows a form of *ceteris paribus* reasoning. Where reasons are available within a given context, they may or may not be taken into consideration for weighing. The context is given and thereby sets the weights of the reasons for an agents' perspective. This allows reasons to be removed and added without changing the weights, since the context is kept fixed.

IHiBO-DET cannot selectively remove specific reasons from consideration. IHiBO-DET takes the reasons, options, and weights and determines which options are permissible. A new feature could be added that allows IHiBO-DET to have some degree of agency in selecting reasons. This could be achieved, for example, via a consistency-of-reasons criterion, or by allowing IHiBO-DET (or IHiBO in general) to remove reasons until at least one permissible course of action remains.

Context. The context can be modeled as a structural graph, as developed by Streit et al. (2023). The graph will have subgraphs and supergraphs that also serve as contexts. To date, this avenue has not been investigated, but could become more relevant when looking at agents that with different perspectives on the situation. For example, in such a graph-based representation, IHiBO may limit itself to finding a common ground among agents, considering only weights from reasons that all agents acknowledge and finding a context that all agents agree on.

Options. If there is no information exchange among the agents, they cannot communicate and agree on the contents of the following sets: grounds, options, reasons, context. This means that since grounds and options are propositions, they may overlap.

Also, according to the definitions of reasons, an agent interprets a ground in a pairwise comparison as a reason for only one option. However, since IHiBO merges input from multiple agents, this assumption cannot be maintained. As mentioned earlier, it may be that, in a pairwise comparison of two options, two different agents take the ground as a reason for opposing options. In many cases, this will not be noticeable as the weight of one reason may outweigh that of the other reason, thereby effectively canceling out the other reason's weight. In other cases, a ground may serve as a reason that justifies both options or makes both options required.

This is sensible. Consider, for example, the case of buying a product. The product has a price, and the purchaser must be able to afford, so having sufficient funds justifies purchasing the product. If there are two products available and a choice must be made between them, having sufficient money to buy either justifies both options. Recall that the dynamic dual scale is about understanding which options are permitted, not which option is best.

4.9 Some additional considerations

Several additional topics are also relevant to this discussion.

Reputation. The reputation factor introduces a multiplicative component to the additive system. Reputation setting is an important aspect, but it is not fully articulated in this dissertation. Reputation can be understood in multiple ways. It appears to function as a learning parameter. It resembles the Elo score in competitive games. It can be read as a trust factor, or a credit-scoring system. This makes the reputation factor both very useful and disputable. Reputation could have operational roles beyond its function as a weighting mechanism in collective decision-making. Reputation could also be used to verify that an external agent associated with a particular address is a trusted source. Receiving input from an unknown address associated with a good reputation may be valuable in other contexts as well. This is particularly the case if that reputation value is reliable for other users, even if they do not know the address. It can allow users to trust inputs provided by external users with a good reputation.

Documentation detail. The level of documentation detail is another critical component. The benefit of using weights is that they reduce computational cost and quantify the influence of a particular piece of information (a ground). Therefore, the proposition or information described in the ground, together with the associated weight, should be sufficient to express why a particular option is considered good or bad. If less information is provided, making the ground too ambiguous, different people may interpret it in different ways, making it difficult to arrive at a satisfactory explanation. The voters, therefore, should consider carefully not only their weights but also the grounds they provide. As an alternative to providing only a minimal ground, the ground can reference a larger, more specific document to provide more detail.

Financial deployment. The goal of IHiBO is to build trust by documenting portfolio managers' reasoning before they make decisions. This way, if investors wish to consult the reasons portfolio managers use to arrive at particular decisions, they can obtain that information. IHiBO is not strictly necessary for this purpose, as a standard non-distributed information system could be used instead, and would likely be less expensive. Nevertheless, IHiBO, being an Ethereum smart contract, offers advantages.

Portfolio management is already a highly digitalized sector. Portfolio managers are required to use many software systems, partly because the market moves quickly. Computer systems are used to view the market and to analyze stock indices, and these services are often combined into a single software system. A portfolio manager can place trades manually or use parameters to let the system manage trades on its own. In either case, any change to a portfolio can be documented with additional contextual information about the trade.

IHiBO enables this process on the blockchain. This means it can interact with other smart contracts on the blockchain and access data on the blockchain. Furthermore, documentation is strengthened by the blockchain's immutability. Portfolio managers are guaranteed that already documented information cannot be altered. Thus, if a portfolio manager makes hasty decisions and subsequently seeks to cover them up, it would be very difficult to do so. If the portfolio management corporation is attacked by hackers, the information is less vulnerable since it is on the blockchain. However, this security depends on the exact implementation. If the corporation records hashes on the blockchain but leaves the actual information in its database, this remains a vulnerability. Moreover, companies often need an off-site backup of their systems. One key benefit of smart contracts is that they are specified in advance, and that investors can see which contract is used. This makes the pipeline more verifiable and thus trustworthy.

There are also downsides to using blockchain technology. The main problem is that blockchain is not universally accepted as legal. Some blockchain-related actions such as exchanging fiat currency for cryptocurrency have been made illegal in certain countries. The execution of the IHiBO-NB balancing smart contract does not, by default, take laws into account. Portfolio managers will need to ensure that the laws are either well represented by reasons and weights or that an additional smart contract is included that filters out illegal options before execution. This becomes especially important when a dynamic dual scale is implemented for IHiBO, as it allows managers to submit options unconstrained to the smart contract.

Beyond adversarial and collaborative settings. A smart contract can also implement a form of negotiation using numerical balancing. The managers of two adversarial parties can specify their reasons for permissible trade options. Where the options concern the specifics of the trade (i.e., quantity, cost, etc.) if no permissible trade is found, the managers can add more reasons or revise their weights. If a single permissible option is found, the managers can proceed to execute the permitted trade. If multiple permissible trades are possible, they must first agree on a single option before finalizing the trade.

If IHiBO can be used in both adversarial and collaborative settings, it may also be possible to use the system in mixed settings. Mixed cases involve multiple managers whose collaboration is not fixed a priori. Such situations are typical in politics, where parties may oppose each other on one topic and collaborate on another.

Furthermore, the IHiBO-NB smart contract could potentially be used as a blockchain consensus mechanism. Common consensus mechanisms are Proof-of-Work (PoW), Proof-of-Stake (PoS), and Proof-of-Authority (PoA). These mechanisms determine which block is accepted based on work, stake, and authority respectively. These three concepts can all be seen as reasons for accepting one block over another. Balancing reasons may then be understood as a generalization of all three consensus mechanisms.

A recursive/inductive approach. It may be possible to derive weights recursively using the output specified in IHiBO to increase the weight of particular reasons. The ground cases consist of single reasons, each with a weight of 1. Then, as these cases are combined into more complex reasons, the associated weight may increase to larger natural numbers.

Say there is a reason r_0 in favor of option o_1. In a context with only this reason, and assuming a single proportion symmetric weight function, comparing o_1 to the other options will make o_1 permissible and the other options impermissible. This is the base case, i.e., single reason contexts. Depending on the feedback, this outcome can serve as either a positive or negative reason to choose the option again.

So, $((r_0, o_1, +), o_1, v)$ expresses the feedback case where $v \in \{-, 0, +\}$. If $v = +$, the feedback indicates that the previous valuation was good; if $v = -$, the valuation led to a bad outcome, which is a reason to choose an alternative option.

4.10 Related work

This work integrates the theory of normative reasons with blockchain technology to address decision-making in multi-agent systems, with a focus on trust fund management. Several existing research areas are relevant to the proposed approach.

First, normative reasoning frameworks have gained attention for formalizing how agents reason about actions in the presence of competing reasons, often balancing pros and cons. In particular, the foundational work of Knoks et al. (2024) introduced a structured model for numerical balancing of reasons in which actions are evaluated based on weighted pro and con reasons. This work has been extended to enable practical applications, such as portfolio management, in which the aggregation of reasons informs critical financial decisions.

In the context of blockchain-based decision-making, the work described in this thesis aligns with the broader push to leverage blockchain's decentralized and immutable properties to improve transparency and accountability in multi-agent systems (Manolache et al., 2022). Previous studies have explored various ways of integrating argumentation and decision-making frameworks into blockchain environments (Yu, Zichichi, Markovich, & Najjar, 2022; Yu, Zichichi, Réka, Amro, et al., 2022; Yu et al., 2021; Zichichi, 2021). The use of abstract agent argumentation frameworks in decision-making processes has been applied to several blockchain-based protocols, providing a formal structure for evaluating conflicting arguments in decentralized systems. The approach builds on these ideas by focusing specifically on a reason-based balancing model rather than abstract argumentation, demonstrating a novel application of balancing normative reasons in fund management through smart contracts.

Finally, comparing the abstract argumentation approach (Yu et al., 2021) with the reason-based balancing model (de Wit, Knoks, & Van der Torre, 2025a; Knoks & Van der Torre, 2023; Knoks et al., 2024), as implemented in IHiBO (de Wit, Yu, et al., 2025), reveals notable differences in both computational efficiency and the underlying reasoning methodology. Although abstract argumentation typically involves resolving conflicts between arguments, numerical balancing simplifies the decision process by aggregating reasons with varying weights and polarities. This leads to significant reductions in computational costs, as demonstrated by the comparative analysis in the evaluation section.

The contribution of this thesis also intersects with explainable AI (XAI) research in which systems are designed to provide transparency into their decision-making processes. Argumentation-based systems, such as IHiBO, align with the goals of XAI (Vassiliades et al., 2021) by enabling decisions to be explicitly documented and explained before execution. The ability to trace the reasoning behind each decision is particularly valuable in high-stakes areas such as financial management, where accountability is important. This is especially relevant for automated trading approaches, as they have caused flash crashes in the past (Kirilenko et al., 2017).

4.11 Conclusion

This chapter evaluated the practical integration of reason-based detachment into the IHiBO framework and compared its performance with abstract argumentation. The analysis focused on computational efficiency, usability, and conceptual clarity. Experimental results demonstrated that the additive detachment operator exhibits linear growth in gas cost, in contrast to the exponential growth observed in argumentation-based smart contracts. This confirms that numerical balancing is more scalable for deployment on blockchains.

Beyond performance, the evaluation highlighted trade-offs in expressiveness and transparency with abstract argumentation. For future work, an equivalence result would be helpful to determine whether the two can be expressed in terms of one other and, if so, under what conditions. The restriction to set options allows for greater focus in a decision procedure than in abstract argumentation, which may output sets that do not specify how to proceed. Also, by specifying the issue and introducing polarity, reasons can be related to the issue under the specified polarity. This allows managers to specify how they want the information they provide to be interpreted in relation to that issue. Previously, in IHiBO-AA (abstract argumentation), there was no solution for defining the conflict relation between arguments through agent voting. This change gives managers more control over their inputs.

Although the implemented model effectively captures polarity and weight interactions, dynamic-scale tournaments and full dual-scale semantics remain future work. Nevertheless, the current implementation is generally sufficient, since traders often focus on a single stock at a time. The traders may implement the strong versions of polarity in ways that suit them; for example, a strong sell may be interpreted as selling all stock. Another issue is that dynamic scale is expensive because it requires balancing all options against one another. Reducing it to single balancing cuts costs; however, it is not clear whether expressivity is lost, and, if so, whether it can be recovered. Further research may compare dynamic scale expressivity and cost to those of the currently implemented NB and AA versions. Overall, the choices lead to a well-rounded implementation.

The use-case analysis illustrated how the framework supports auditable decision-making in domains such as portfolio management and consumer choice. The addition of reputation, confidence, and polarity increased expressivity for fund managers. Managers were previously unable to specify how they wanted their arguments interpreted in IHiBO-AA. In IHiBO-NB, they can now express how they wish their argument to be interpreted and how much influence it exerts on the outcome.

In general, the evaluation validates the feasibility of embedding normative reasoning in decentralized systems, paving the way for more robust, transparent, and regulation-compliant, low-cost decision pipelines.

Chapter 5

Discussion of Perspectives

This chapter critically examines and seeks to further clarify the theoretical framework introduced in Chapter 3 and its implementation in Chapter 4. The focus is on assessing whether the implementation faithfully reflects the principles and detachment models defined earlier, and on identifying conceptual and practical caveats. Key questions addressed in this chapter include:

- Does the implementation fully capture the (dynamic) dual-scale detachment theory, or does it introduce simplifications?
- Where do theoretical constructions break down in practice (e.g., weight normalization, contrastivism, dynamic scale)?
- Which inconsistencies arise when merging multiple weight functions in a multi-agent setting?
- What implications does IHiBO have for the future directions of detachment in a multi-agent voting context?

The discussion concerns analytical rather than experimental work, highlighting gaps between theory and implementation, such as the absence of dynamic-scale tournaments and reliance on a permission scale rather than full dual-scale semantics. It also examines vulnerabilities in the voting process, including adversarial weight inflation and epistemic asymmetries between agents. It then examines whether the principle-based approach can be extended to other system dimensions. These findings inform recommendations for refining the framework. These findings will guide future work on improving theoretical fidelity and robustness in practical deployments.

5.1 Introduction

After rendering a scale model in mathematics and implementing a version in IHiBO, some issues remain unaddressed. Although the balance scale detachment smart contract appears to be less expensive than the argumentation smart contract, it remains unclear how well IHiBO performs with this detachment. This is especially the case given the new voting setup, where fund managers have more parameters to specify in their inputs. These new parameters increase the dimensionality of the input space relative to the abstract argumentation implementation, thereby requiring more expertise from fund managers when formulating their inputs. How should the reputation be set? What weights should fund managers use? What happens when a fund manager does not know about an option? How does this system relate to the law, and should laws be represented on the scales as reasons? Another issue is the level of documentation required when using the smart contract, especially regarding reasons. A reason may have extensive context, but if all of it is stored, the cost of documentation increases. A benefit of the weighted reasons approach is that it documents the reasons and their weight. Even if some information is not documented, its significance can still be represented in the weight. The main caveat in the theory is that different agents assign weights to reasons differently. Although this flexibility is the main benefit of IHiBO, it creates a comparability problem regarding the relative meaning and influence of those weights in determining the outcome. Simply combining the weights may be too ad hoc and could yield undesirable results. A crucial part of the solution to this problem is documenting what each manager votes. This creates accountability for the smart contract's output and gives managers an incentive to vote to avoid, or at least minimize, risks when assigning weights.

The layout of this chapter is as follows. Section 2 looks at the gap between the two previous chapters and the literature. Section 3 takes a closer look at the system developed considering the following perspectives: the input, the detachment process, the output, and the use case. Section 4 recaps the main points, including caveats, limitations, and alignment with or perspectives on other approaches. Section 5 discusses additional topics related to the subject matter that require further elaboration than what has been provided so far. Section 6 is the conclusion.

5.2 Connections of the Contribution

The two previous chapters developed a principle-based approach to dual-scale detachment and implemented a numerical balancing smart contract for IHiBO. This section examines the differences across multiple dimensions. First, the difference between the (dynamic) dual scale and the numerical balancing implemented as a smart contract is analysed. Then, the formal approach to balancing weighted reasons presented by Streit (2024) is compared with the approaches presented in this dissertation. The abstract argumentation implementation in IHiBO is also contrasted with the numerical balancing implementation. Then, the legal aspect is situated in the context of the DILLAN project (Digitalisation Law and Innovation). Finally, the impact of dual weights on other areas of the literature is examined.

5.2.1 Dynamic dual scale and IHiBO implementation

As already mentioned, there is a difference between what was implemented as a smart contract for IHiBO and the dynamic dual scale. So far, the theory behind the smart contract has been called the numerical balancing smart contract. The first difference between these two is conceptual. The dynamic dual scale is grounded in philosophy and adopts an ethics perspective. The smart contract implementation simulates agreement between fund managers and focuses on integrating different perspectives. More technically, in the former there is a one weight function, whereas in the latter multiple weight functions are combined.

Chapter 3 definitions. The specification of the dynamic dual scale, given in Chapter 3, is as follows:

Definition 48 (Additive Dual Scale Weighing (of Options))**.** *Let V be the set $\{+, -, 0\}$, called* (scale) values*. An additive dual scale weigh model is a* weighing function $D_w : SR \times O \times O \times \mathcal{P}(G \times O) \times C \to V \times V$ *that assigns to each pair of options o_1 and o_2: a subset of reasons $RR \subseteq R(o_1, o_2)$ that are relevant to this pair, a context c, a weighting function w, and a pair of scale values (v_1, v_2)*

as follows:

$$v_1 = \begin{cases} + & \textit{if } \Sigma_{x \in JW_{o_1}} x > \Sigma_{y \in RW_{o_2}} y \\ - & \textit{if } \Sigma_{x \in JW_{o_1}} x < \Sigma_{y \in RW_{o_2}} y \\ 0 & \textit{otherwise} \end{cases}$$

$$v_2 = \begin{cases} + & \textit{if } \Sigma_{x \in JW_{o_2}} x > \Sigma_{y \in RW_{o_1}} y \\ - & \textit{if } \Sigma_{x \in JW_{o_2}} x < \Sigma_{y \in RW_{o_1}} y \\ 0 & \textit{otherwise} \end{cases}$$

where $JW_o = \{jw((g,o)) \mid (g,o) \in RR\}$*, and* $RW_o = \{rw((g,o)) \mid (g,o) \in RR\}$*. When the options, the set of relevant reasons and/or the context are important for the weighting function* w *under discussion, write* $D_w(R, o_1, o_2, RR, c) = (v_1, v_2)$ *or* $D_w(R, o_1, o_2, RR, c, v_1, v_2)$*. When these are clear from the surrounding text, they can be written as* $D_w(RR, v_1, v_2)$ *or* $D_w(RR) = (v_1, v_2)$*.*

Then, using a dual scale dynamic scale was specified in the following way: Definition 49.

Definition 49 (Dynamic Scale Detachment)**.** *Let the values* $V = \{+, -\}$ *be positive and negative, or permissible and impermissible. Then, given a set of options O, a context* $c \in C$ *and a weighing function* D_w *associated with it, a dynamic detachment function* D^* *assigns values to each option o in O as follows:*$D^*(R, O, RR, c)(o) = +$ *if* $D_w(R, o, o', RR, c) = (+, v)$ *or* $D_w(R, o, o', RR, c) = (0, v)$*, for all* $o' \in O$ *such that* $o' \neq o$*;* $D^*(o) = -$ *otherwise.*

A dynamic scale allows us to say, given a set of options, which of those options are permissible in the context, and which are not. A dual scale compares only two options, but a dynamic dual scale is a pairwise comparison of all options against all others. When there are three options, six comparisons are made: o_1 vs o_2, o_1 vs o_3, o_2 vs o_1, o_2 vs o_3, o_3 vs o_1, o_3 vs o_2. If an option is found to be non permissible in any pairwise comparison, that option becomes impermissible all things considered.

Chapter 4 specification. The numerical balancing implemented in Chapter 4 combines the balancing approach of Chapter 3 with ideas from portfolio management. Taking inspiration from a trading platform, the use case chosen was single-stock trading and it incorporated five types of advice corresponding to five types of actions. These were: $++$ (strong buy), $+$ (weak buy), 0 (hold), $-$ (weak sell), $--$ (strong sell). The managers have a reputation and can vote with weighted reasons that have a polarity; the polarity indicates which option the reason favors. Below is the algorithm 2 introduced in Chapter 4.

Algorithm 2 Decision Procedure

Require: $\|Reasons\| > 0$
Ensure: $Issue$

$neut \leftarrow \sum_{r \in \{r \in Reasons | r=(x,Issue,0)\}} f_w(r, (Reasons, Issue))$
$pos \leftarrow \sum_{r \in \{r \in Reasons | r=(x,Issue,+)\}} f_w(r, (Reasons, Issue))$
$ppos \leftarrow \sum_{r \in \{r \in Reasons | r=(x,Issue,++)\}} f_w(r, (Reasons, Issue))$
$neg \leftarrow \sum_{r \in \{r \in Reasons | r=(x,Issue,-)\}} f_w(r, (Reasons, Issue))$
$nneg \leftarrow \sum_{r \in \{r \in Reasons | r=(x,Issue,--)\}} f_w(r, (Reasons, Issue))$
$sum \leftarrow (pos + 2 * ppos) - (neg + 2 * nneg)$
$magn \leftarrow |sum|$
if !$(2 * magn > neut)$ **then**
    $confidence \leftarrow neut$
    $dec \leftarrow 0$
else if $sum > 0$ && $ppos > pos$ **then**
    $confidence \leftarrow ppos$
    $dec \leftarrow ++$
else if $sum > 0$ && $ppos < pos$ **then**
    $confidence \leftarrow pos$
    $dec \leftarrow +$
else if $sum > 0$ && $nneg > neg$ **then**
    $confidence \leftarrow nneg$
    $dec \leftarrow --$
else if $sum > 0$ && $nneg < neg$ **then**
    $confidence \leftarrow neg$
    $dec \leftarrow -$
else
    $confidence \leftarrow 0$
    $dec \leftarrow 0$
end if
emit Output$((confidence, (Reasons, Issue, dec)))$
return $(confidence, (Reasons, Issue, dec))$

The difference. The difference is that the IHiBO implementation is only specified for one option, and it uses polarities, not only $+$ and $-$ but also $++$, 0, and $--$. If the NB IHiBO implementation is limited to $+$ and $-$, it becomes extremely close to the permission scale of dual scale detachment.

In that case, $+$ can be understood as justifying weight for option o and $-$ as requiring weight for the alternative. However, the NB implementation in IHiBO may be viewed as a dynamic scale producing an all-things-considered output regarding a particular option (o or $\neg o$. This single scale should output whether the option is permitted, all things considered. The practical dimension of IHiBO allows reasons to be missing. Also, the output is interpreted differently. It is not understood as permission or acceptability but as a decision regarding the stock; i.e., buy (strong/weak), sell (strong/weak), or hold.

A single balancing operation is less expensive than many balancing operations, therefore, this approach is beneficial for on-chain execution. However, the details of mapping the correct weights to $+$ and $-$ need to be worked out. If one sets out to replace the dynamic dual scale with a single permission scale to determine the permission of a single option, this is a pairwise comparison of o with $\neg o$, where $\neg o$ denotes the set of alternative options. The system in Chapter 3 states that the weight of reasons can differ across pairwise comparisons. At the moment, it is not proven that the information conveyed by this system can be translated into a single normative balance, i.e., a binary comparison between option o and $\neg o$.

5.2.2 Streit's structural approach and dynamic dual scale

In the dissertation of Streit (2024), a first-order representation is developed as a structured approach to context, inspired by the conception of balancing within the reason-based approach to deontic logic. This subsection clarifies the relation between the dynamic dual scale and Streit's structural approach.

Streit specifies reasons as triples $(premise, conclusion, role)$. This specification allows them to be a subclass of the type "normative consideration" where reasons have the role "reason+" or "reason-", i.e., positive or negative reason. The weight of reasons is defined structurally via a directed graph, and each normative consideration in the graph is assigned a single weight. The elements of the graph can have a single directed arrow to another element, and the leaves of the graph connect to reasons; the reasons relate to options, also called actions. The graph creates a structure that implements the final weight of the reasons, and the weight function assigns weights to each element. This graph may also be understood as implementing the weighting of reasons for the weight function specified in Chapter 3 of this dissertation. However, the most interesting difference is in how the balancing function yields the deontic status of the options.

In his dissertation, Streit calculates the normative balance for each option by aggregating the relevant weights and subtracting negative values from positive ones. Paraphrasing his definitions, we have:

$$\delta_o = \sum_{r \in \{(p,c,role)=r \in Reasons \mid c=o \wedge role=reason+\}} w(r) - \sum_{r \in \{(p,c,role)=r \in Reasons \mid c=o \wedge role=reason-\}} w(r)$$

This number may be negative or positive. However, this is not the end result. Streit uses this number to identify a set of options with the best normative balance and deems those options permissible, while the others are deemed impermissible:

$$\texttt{permitted-options} = \{c_i \in \mathit{options} \mid \delta_{c_i} = \max(\delta_{c_1}, \delta_{c_2}, \ldots, \delta_{c_n})\}$$

This should give the set of c_i such that the corresponding δ_{c_i} has the maximum value in the set $\{\delta_{c_j}\}_{j=1}^{n}$. If there is only one option in the set of permitted options, then that option is obligatory.

Comparison. The computation of the normative balance δ is essentially the same computation made in the permission component of the dual scale for the pairwise comparison between option o and its negation $\neg o$. However, because this number is negative on the dual scale, the option is impermissible. In Streit's system, an option with a negative δ can be permissible when there are no alternatives with a greater δ.

This strongly relates to the discussion on the existence of dilemmas. A normative dilemma occurs when no options are permitted, which would occur when the dynamic dual scale results in every δ being negative. Streit's system states that there is always a permitted option. The system in Chapter 3 allows the modeling of situations with no permitted options.

In the system of Chapter 3, in order for an option to be permitted, it must have a neutral or positive normative balance (δ) in each pairwise comparison. Streit uses the ordering implied by the δ's to order the options and only considers those with the highest δ values as permitted. This difference is worthy of discussion. Does Streit's system imply that normative balance implies a type of optimality? The system of Chapter 3 will, in most cases, provide more permitted options. However, after that, there can be several ways to find an option that can be considered optimal within the space of permitted options. If normative dilemmas exist, there will be options with a negative normative balance, and Streit's system provides out-of-the-box a way to address them. The system in Chapter 3 does not provide a way to handle dilemmas. One may argue whether, in the case of a normative dilemma, there should always be options considered permitted or that when there are no permitted options one is obligated to commit a forbidden(illegal) action. If dilemmas do not exist, then when the system in Chapter 3 detects a dilemma, it means that something is missing or wrong with the information under consideration.

5.2.3 AA- and NB-Smart-Contract

The portfolio management (also known as fund management) use case is a complex scenario to model. Financial managers control the content of a portfolio. Normally, it holds liquid assets (i.e., money), which can be used to buy assets in many forms: stocks, bonds, securities, options, real estate, cryptocurrency, gold, and more. The assets can be sold to obtain liquidity, and liquidity can also be obtained in a specific currency. The decisions regarding the portfolio's content are guided by the investor policy statement (IPS). In that document, investors and managers agree on a policy that sets the portfolio's direction. This involves setting the acceptable risk level, the initial portfolio content, the type of content, and other relevant factors before investing.

When evaluating the portfolio's content, the managers will need to assess the quality of each asset. This depends on the current price of each asset and its estimated future price. The price of an asset is determined by the market. If market participants buy and sell an asset at a particular price, then that is its price at that moment. Here, one can picture a stock price chart. However, different assets work differently. For example, real estate does not change hands often and tends to fluctuate less in value.

The question of what the best portfolio is at any time is, therefore, rather complex. IHiBO was developed to document fund managers' reasoning behind the decisions they make regarding the portfolio. Two smart contracts were created by Yu et al. (2021), one implementing negotiation moves to model a negotiation between two adversarial parties, and the other implementing abstract argumentation to model the discussion of a team of fund managers. That work was further presented in greater detail in Yu's (2023) dissertation.

In Chapter 4, a new smart contract was developed and compared to the abstract argumentation smart contract. The new smart contract implements ideas from Chapter 3 and uses a balance scale to compare the weight of the relevant reasons in order to reach a decision.

We now compare these approaches again. They both make specific assumptions about fund managers' reasoning. To complicate matters, the reasoning and decision making process of fund managers is not entirely clear; this has not been studied. In fact, today, much of portfolio management is done digitally, with managers simply setting some parameters in the software.

Abstract argumentation smart contract. This theory of abstract argumentation is very flexible and allows any type of information to be embedded as an argument. Moreover, any method can be used to set a conflict relation between the arguments. This implies that a great deal can be modeled with this approach. There is, however, one limitation. The theory does not set defaults for the use-case domain, even though such defaults may be desirable, unless a way is found to set those parameters in the graph.

The expressivity of this approach means that it should be possible, in principle, to specify every nuance of portfolio management in the abstract argument graph. Once a graph is obtained, semantics can be applied, and the one implemented for IHiBO is preferred semantics. This seems like a good choice, but it is not explained why this semantics was chosen. Also, this semantics need not yield just one result. After the semantics is applied, there may be two options left, of which the managers presumably choose one, but IHiBO does not know which one.

The main problem is how to obtain the abstract argument graph. To obtain arguments, the smart contract uses voting. The managers submit their arguments, and when a manager votes for one argument, that argument gains a strength point. Strength points are used to support the idea that when many managers acknowledge an argument, it cannot be defeated by an argument that fewer managers acknowledge.

However, the current implementation does not provide a procedure for generating the conflict relation between arguments. There is a function that allows a conflict relation between two arguments to be submitted, but this is not an official part of the manager voting procedure. The managers, therefore, do not have control over how the arguments they acknowledge are interpreted.

Comparison. Two comparisons can be made: first, the AA theory with the theory in Chapter 3; second, the AA theory with the implementation in Chapter 4. Here, the comparison of the AA theory with the theory implemented in Chapter 4 is the focus, since Chapters 3 and 2 have been compared above.

The new smart contract implements a single decision outcome, arbitrary positive-integer-weighted reasons, manager reputation, and voting with polarity. These are all advantages over the abstract argumentation theory. At the same time, the new smart contract was found to be less expensive to execute. This was to be expected, since the algorithm does not need to cycle through a graph to compare arguments via the defeat relation.

Nevertheless, the implementation does make some concessions to obtain these advantages. To the best of my knowledge, the inclusion of weighted reasons that are voted on is an improvement because it gives managers more control over what they are voting on. However, as will be apparent in this chapter, this increase in expressivity has some drawbacks when aggregating from multiple sources(agents in IHiBO context fund managers). In particular different agents may use weights differently creating a Babylonian confusion of tongues of sorts.

The balancing output is a single choice, which is good; however, the output domain is severely limited. Instead of arbitrary options, the balancing occurs in relation to a single stock and allows only 5 choices: strong buy, weak buy, hold, weak sell, and strong sell. These options are fair; they can be implemented such that "strong" is set relative to a particular amount of liquidity or stock, for example, strong sell refers to selling all of that stock, while weak buy and sell refers to buying/selling a single share of that stock. Moreover, the limitation to a single stock is sensible, as in today's digital markets, many investments take place through baskets of stocks, such as index-tracking instruments and ETFs (exchange-traded funds), which can be treated as a single stock. However, this is still a limitation; a dynamic dual scale can accommodate arbitrary options, including comparisons of full portfolio makeups. To some extent, the inclusion of options as explicit elements is also an improvement. The options in the argumentation theory are not made explicit, so, in principle, it is not guaranteed that an option is included in any of the output sets of the argumentation semantics.

5.2.4 The DILLAN connection

This research is part of the DILLAN (Digitalisation, Law and Innovation) project. While many such projects focus on law, this one takes a computer science perspective. The project was designed to connect law with AI by using logical methods to build greater trust in complex systems that involve legal aspects. This subsection highlights how Chapter 3 and Chapter 4 relate to this idea. The most tangible connection to the legal domain is the contractual relationship between the investor and the fiduciary in Chapter 4. The balancing that occurs in both chapters is similar to that of the legal domain where different pieces of information need to be weighed against one another to reach a decision. The system in Chapter 3 may have legal value, as it could provide guarantees for black-box systems.

IHiBO. The Intelligent Human-input-based Blockchain Oracle (Yu et al., 2021) fits this purpose. The protocol was developed to improve the trust relationship created through a contract between a principal and a fiduciary (Yu et al., 2021). In the case of fund and portfolio management, the principal is the investor and the fiduciary is the manager. The fiduciary thus has a legal duty to be loyal to the principal's goals, and the principal must be able to trust the fiduciary to act in the principal's best interests. A fiduciary by contract has a legal obligation to give the principal insight into their actions upon request. In other words, the principal has the legal right of oversight over the fiduciary's actions.

IHiBO aims to improve this relationship through distributed ledger technology (DLT). More precise, it uses a DLT called Ethereum. Ethereum allows the specification of smart contracts (code/algorithms) using the Solidity programming language. Before execution, users of the smart contract can read the code and accept its use before committing to anything. IHiBO supports the documentation of information considered during decision processes. This allows both the fiduciaries and the principals to inspect the code and understand the documentation process. By documenting the information considered by the fiduciaries, the principals can review the information that led to decisions. Therefore, the smart contract should increase the principals' trust in the fiduciaries. This idea may be understood as AI in the loop.

Beyond fund management, the application can be used for any situation where records need to be kept of processes that can be simulated by balancing. One can imagine this as a model that is updated with inputs. The input-output relation is changed by the weight adjustments introduced by the updates. If the system's functioning changes such that the outcomes become unacceptable, the series of updates may be analyzed using IHiBO's documentation. Using such documentation of the update sequence is still largely conceptual. It is not clear how to use this documentation to determine which updates were "wrong". Future research may also seek a better understanding of what documentation the legal field requires. At the moment, IHiBO may document: grounds, options, weights, balancing process, time, voters' addresses, who voted for which grounds, who voted for which options, and who voted for which weights. Such future research may be understood in the context of regulating AI systems. The relevant question is: what permitted an AI system to perform a particular action?

Balancing. Balance scales are present in both chapters. In Chapter 3, single-scale and dual-scale approaches are studied, while in Chapter 4, a balancing-reasons model is implemented. The idea of balancing, however, also has a strong connection to law. To some extent, one can say that judges balance the reasons presented to come to a judgment. However, other concepts, such as checks and balances, are present across the legal domain. Using knowledge representation methods to develop precise representations of what happens in such processes may improve legal decision-making. Describing how decisions are made in legal processes may improve decision-making in automated systems. At the same time, developing a representation for such reasoning may lead to improved notation in decision processes. Documenting how much weight a particular reason has may improve the reasoner's assessment of the influence of the reason on the decision process; i.e., the influence of the reason is quantified. This may help the decision-maker to compare the outcomes of different cases involving similar reasons, or to see how the attached weight influences the process, thus making decision boundaries and the weight of reasons across contexts more tangible.

An example in the literature of a formal representation of balancing is provided by Sobek and Montag (2018). Here, proportionality is addressed and described formally. The work presents a weighted approach that assigns three weights to a legal principle. The approach balances the intensity of interference, the importance, and the reliability of empirical assumptions to determine which principle should be given priority. The three factors are multiplied to obtain the aggregate for each colliding principle, and then the two aggregates are compared by dividing one by the other. The output is 1 if the two aggregates are equal, less than 1 if the first principle is more important, greater than

1 if the other principle is more important. The Chapter 3 approach can model this by delegating the setting of each principle's aggregate weight to the contextual weight function. Then the two principles are pitted against one other as competing options, and the aggregates can be compared. However, the output will not be a gradient indicating the extent to which one weight aggregate exceeds the other.

Detachment Functions. Used in Chapter 3, these offer a systematic way to implement and compare methods for connecting a domain of inputs with a domain of outputs, called grounds and options, respectively. The connection of detachment functions with logic has already been established by Knoks and Van der Torre (2023). Throughout this dissertation, connections have been made with artificial neural networks and abstract argumentation. Considerable steps can still be taken to obtain a clearer picture of the relationship between artificial neural networks, abstract argumentation, and their expression as a detachment function or an ensemble of detachment functions; this may, in time, even extend to LLMs, which are a type of artificial neural network. This may foster a deeper understanding of the consequences of these systems' foundations. It may, in turn, allow researchers to study black-box systems as a detachment function that determines which principles the system is guaranteed to satisfy. This fosters a foundational understanding of the input-output relations that the black box system can construct, giving the legal domain an entry point for concrete regulation of such systems.

An abstract argumentation semantics takes as input an abstract argument graph, which is a set of arguments together with a defeat relation. And as output, it generates a set of extensions, i.e., a set of sets of arguments. An option may either represent a set of arguments or an individual argument. However, in the latter case, it may be difficult to express multiple sets of arguments as output since the semantics assigns a truth value to individual arguments rather than to sets of arguments. The reasons may be understood as encoding the defeat relation together with the weight function. The grounds will most likely replace arguments. This will take work to make precise. The first step is to implement the semantics as a detachment function without a weight function.

An artificial neural network takes an input vector and generates an output vector, and, in general, performs numerical computation to do so. At the end, the output vector may be interpreted as an element of a particular domain, for example, a word. However, the input and output tend to be a single vector. It makes sense to see the output vector as a single option. However, the input vector may make more sense as a set of grounds considered true in which the numbers in the vector represent the weights attached to the reasons. This is not the tricky part; what becomes more complicated is organizing the layers of the artificial neural network within the detachment function and applying possible different operations on weights and collections of weights. A different approach is to understand each artificial neuron as a detachment function of its own. More work is required here. For artificial neural networks, a weight function is needed; one may also start by simply specifying the relation between the inputs and outputs $(\text{input}_1, \text{output}_1)$.

The detachment functions have proven to be a promising approach for expressing and studying relations between an input domain and an output domain. The approach connects the inputs and outputs using graphs and numerical computation. Particular new principles of these functions may be discovered and expressed, some guaranteed to apply and others not. The different systems can then be compared directly. It would be valuable to be able to specify other law-relevant complex systems—such as LLMs—as detachment functions. This would provide an avenue for obtaining guarantees and understanding a wide variety of functions, including their behavior, from white-box to black-box.

5.2.5 Dual weights and the literature

Dual weights may have a more far-reaching impact than ethical balancing models. It may be that dual weights could be applied in more areas of mathematics. This may help explain previous problems when using mathematical models. Single-proportion weights are often sufficient; however, there are cases where the single proportiono assumption leads to model errors. Normative reasoning puzzles show that such cases exist in normative reasoning. Normally, the single-proportion assumption does not lead to problems directly; however, in light of these borderline cases, the model cannot represent what is happening. Since other areas of mathematics have not been studied, it is not possible to draw direct conclusions about them. And many areas of mathematics will likely not be affected, such as geometry, which is very different. Also, mathematicians have methodologies to ensure they can study what they are focusing on without necessarily being constrained by this distinction.

To understand when dual weights may become relevant, it is necessary to understand when a single scale occurs. Since the dual scale has been introduced as an alternative to a single scale, removing an underlying assumption, it can express a single scale through the implementation of single-proportion weights. The single scale occurs when weights, as numbers, are used to compare elements, and the numbers are all included in the same total order; i.e., each number stands in a $<$ relation to every other number. This means it is possible to directly compare the numbers representing the elements' weights. An example of where such comparisons occur is in game theory. In game theory, strategy profiles are given a utility value with respect to each agent and their counter-strategies. This is a single-scale approach, and implementing a dual-scale approach may be relevant. The implementation of justifying utility and requiring utility may lead to different types of strategies. Instead of dominant strategies, strategies may be found that the agent has sufficient justification to choose and other strategies for which it does not have sufficient justification.

Quantification. The whole theory of attaching weights to reasons is based on the idea that importance can be quantified as weights. In this dissertation, Chapter 3 expresses the idea of allowing reasons to be assigned a range of possible weights since, in general, it is not guaranteed that one can attach exact numbers as weights to reasons. Conceptually, it is already more feasible to speak merely of sufficient weight, or that the justifying weight is greater than the required weight against it. This does not imply a commitment to setting an exact weight. The book "The Weight of Reason" by Tucker (2025) is not about defending the view that quantifying the weight of reason is possible or correct. Instead, it seeks to address how to weigh these reasons correctly when setting out on this endeavor.

5.3 A closer look at the pipeline

This section discusses the constructed pipeline. Although the smart contract implemented for IHiBO uses numerical balancing of weighted reasons, it is merely a proof of concept. Full implementation of the detachment function methodology and other balancing functions remains future work. However, the full implementation may be discussed conceptually. Multiple factors are at play, and some nuts and bolts can be modified in particular ways. The pipeline has four parts: the input space, the detachment process, the output space, and the use case. Understanding the functioning of these parts allows for further extension, modification, and application of the system.

5.3.1 A closer look at the input

The input for IHiBO-NB is votes; more precisely, they are weighted reasons from portfolio managers along with their reputation factor. Reputation can be established in multiple ways; essentially, it reflects the degree of trust placed in the manager, either by the group or by a third party. Depending on the implementation, the number of options considered in the smart contract instance can be just two or more. Options can be pre-specified or determined during the voting process by examining the reasons managers supported. The latter can be understood as finding the normative (or "deontic") status bottom-up. The detachments are derived from the weights.

A different voting methodology is to view what IHiBO does as a type of merging of detachment functions through weights. So instead of aggregating the weights, one might also look into combining the detachments directly. This approach to determining the deontic status is top-down. After the detachments have been determined, a weight function can be found. The weights are derived from the detachments.

Voting weights

When users vote on reasons with weights, the question arises: why do the users assign these weights?

Multiple aspects should be considered. The weights that come from an agent implement a detachment function and are therefore chosen in a particular way. In addition, each agent has a different implementation of how context influences weights, and may also differ in which contextual factors they perceive to be present and relevant at the moment of voting. Furthermore, each agent may consider different options due to their epistemic horizon. Lastly, one may consider adding polarity to the input space. This can implement the idea of a reason against an option and may have certain benefits.

Depending on the situation, the weights assigned by users may also be adversarial in nature. In that case, voters are inclined to inflate their voted weights, since this will increase their chances of getting the outcome they want. However, in the fund manager use case, this is less likely since the managers bear responsibility for the outcome and collaborate to achieve a good outcome. In some situations, documentation does not create a sufficient incentive, and reputation needs to have a stronger influence. Or alternative solutions may be considered, such as a maximum total weight an agent is allowed to assign per reason or per process.

The goal of the IHiBO-NB process is to merge the different views to obtain an optimal perspective on what is permitted. The collective view carries more weight and may take into account more reasons and options than any individual would on their own. As a result, the collective view may be less risky. The opposite is also possible, since too many conflicting opinions may result in inaction.

Weight distribution. Even if agents are voting with collaboration in mind, the weights they share can cause problems. An agent's votes, by themselves, yield a coherent set of detachments that the agent considers appropriate, since they are consistent with the chosen weights. However, another agent may use very different standards to determine how the weights are distributed. This means that a "light" reason for one agent could be a relatively "heavy" reason for another. This can be understood as a Babylonian confusion of tongues—agents may be unable to combine weight functions effectively without additional feedback.

Solutions can be devised; for example, choose the smallest weights possible. However, this would imply that someone with only one reason would have to assign particularly low weights. This might make the reason be classified as a light reason, even though the agent regards the reason as very important. In a case involving only one reason, a justifying weight of 1 may be sufficient to make the option permissible. Alternatively, in a case involving many reasons, the same reason may require a justifying weight of 1000 to outweigh all opposing reasons. The reason may be identical in both cases; only the justifying weight differs. It is 1000 instead of 1 because that is what is required for the reason to justify the option in that context.

The weights may also be normalized to cover the same range (e.g., by dividing each weight by the sum of all weights or using softmax). However, this will turn light reasons into heavy reasons if a user does not vote for many reasons. It will also remove details and reduce the impact of heavier reasons. Nevertheless, the order of weights within a weighting function remains intact. Moreover, the weights across functions lie within the same range.

Alternatively, a reference value might be chosen, for example 100 as the reference value "normal" weight. However, the question is what each agent regards as a normal weighted reason. If the point of the weight was to solve this, then choosing a reference value might just undermine the purpose.

The problem of agreement can be "outsourced" to voting weights. Preemptively setting the normal weight bypasses the natural formation of the normal-weight reason as an emergent effect.

Another approach is to give each voter a maximum voting weight, say, 100. If the voter has only one reason and thinks it is extremely important, they can assign it the full 100 weight. If they regard it as not very important, they can still give it a weight of 1 out of 100. This places strict limits on the weights users can use to vote while preserving some degree of freedom of expression. At the same time, it does not really solve the problem but merely regulates it somewhat. To really see the influence of these ideas, they should be tested experimentally.

An interesting feature of IHiBO is that it does implement this to some extent. Specifically, the variables that contain the weights in the smart contract are integers. On the Ethereum blockchain, storage is allocated for integer variables at creation time; the integers used in IHiBO are of type int256. This means they use 256 bits and can therefore represent numbers from 0 to $2^{256} - 1 \approx 1.158 \times 10^{77}$, inclusive. Thus, there is a maximum weight that can be accumulated; however, it is a very high ceiling.

Mathematically, there also exist different sizes of infinity. In theory, this may mean assigning a weight to one reason that is so much larger than the weights possible to assign to other reasons such that the first reason cannot be overruled.

Reputation value feedback could solve the weight-interaction problem; however, it would take time to settle on specific values. This would also give power to those who define how the reputation is set. The reputation mechanism can reduce the influence of voters whose votes lead to "bad" permissions and increase the influence of voters whose votes lead to "good" permissions. This shows how what constitutes a good and a bad permission is determined. Moreover, one can set reputation to be not only agent-specific, but also topic-specific. A finance expert may not know about particular laws, while a law expert does.

Another solution may be to define a specific set of parameters that each manager must include in their votes before casting them. This means they have to accept particular reasons, weights, and detachments in accordance with a pre-established contract between the voters concerning particular features of the system. For example, the contract may seek to ensure that a particular action is always permitted or forbidden. This solution also reduces inconsistencies in proportional weighting by using weights that implement particular detachments, which then become reference parameters for the other weights in the system.

Individual and shared context. Each agent has a context-dependent weighting function. In addition, each agent perceives a particular context and assigns weights based on that context. At any one time, multiple agents may perceive different contexts for various reasons. It might even be that the perceived contexts are not consistent with one another. However, since the agents share only weights, this is not an immediate problem. Nevertheless, the weights should be able to resolve conflicts. The chosen weights determine if a permission is detached or not; therefore, if one agent indicates through weights that an option is not permitted while another agent indicates through weights that it is permitted, then the aggregate of voted weights determines whether that option is considered permitted or not. It may be that they perceive different contexts, and that if they were to agree on which things are true, they might each select a context — still possibly different from one another — that would permit the same options.

Unknown options. The set of options that agents consider possible in a given situation may also differ. This can also be due to various reasons. For instance, they may not be very imaginative, they may exclude certain reasons, or they may simply not be aware of certain options. In a dynamic dual scale, if they do not consider an option, then they may not be able to specify weights relating to that option. In the most unbiased setting, the weight of a reason for an option can only be set when an alternative option is also considered. Therefore, if an agent does not consider an option, that agent cannot set weights for reasons when that option is considered as an alternative. This could be a curse, but also a blessing. This means that if agents are unaware of something, they will not influence its status. However, if some agents would have been against the option had they considered it, the results could be problematic. In addition, they cannot justify an option against an alternative if they are unaware of that alternative.

Because of this, it may be very important to decide whether to make the content of an IHiBO instance common knowledge before output is generated. However, when the interaction is adversarial, it may not be possible to share the content of the IHiBO instance due to privacy concerns; agents may prefer not to disclose certain information to all participants. Even though the current IHiBO implementation does not implement a dynamic dual scale, the current implementation does solve this problem to some extent by focusing on the permission scale of a selected option and allowing managers to assign justifying (+) weight and opposing requiring (−) weight relative to that option.

Polarity. Polarity is a notion that has found its way into IHiBO-NB. This notion is also used in (Knoks & Van der Torre, 2023; Knoks et al., 2024) but was not used in (de Wit, Knoks, & Van der Torre, 2025a) and Chapter 3. The idea is that the voter explicitly specifies how the reason relates to the option. A connection to the dual scale can be established depending on the weighting function used. If a reason with polarity has a single weight, the weight simply goes into the justifying pan of the specified option's permission scale if +, and into the requiring pan if −. However, with dual weights, more information is needed to specify exactly which pan the weight should be placed in under which comparison. In that case, a − polarity simply makes the reason into a reason that supports the alternative, the justifying weight justifies the permissibility of the alternative, and the requiring weight requires the impermissibility of the specified option.

The polarity in IHiBO-NB specifies discrete levels of polarity for a reason with respect to an option, i.e., ++ (strongly in favor) and + (in favor), 0 (neutral), − (against) and −− (strongly against). These polarity levels were interpreted as options themselves; they can also be used in detachment as polarity in reasons for options. If the reason is against the option, then the reason can participate in each pairwise comparison of that option; however, it will work in favor of the impermissibility of the option. The discrete levels can distinguish between weaker and stronger reasons. For example, +, compared to ++, represents a weak reason in favor. The 0-polarity reason was introduced; this polarity places greater weight on detaching from a polarity other than 0, requiring greater confidence to avoid detaching 0. This can be understood as vying for additional certainty. The IHiBO implementation allows for ++, +, 0, −, −−, but more can be added. If users do not use ++, 0, and −−, the system will function as a normal additive balance scale but with reasons against an option as an additional feature. To exclude reasons against an option, one should remove the − polarities.

Sharing Detachments

A different approach is to combine detachments directly; this is a top-down approach. It was not studied in the principle-based analysis or implemented in IHiBO-NB. However, it may have some benefits. For example, it may be good to derive detachment directly from laws and enforce them on the weighting function to conform to those detachments. Merging detachments directly, however, runs counter to the design of IHiBO-NB, which derives its strength from delegating the detachment merger to the weight aggregation process. If managers disagree on the status of a particular detachment, then the resolution of this conflict is determined by the weights assigned through voting. Nevertheless, particular cases may be better solved by considering the agents' detachments, i.e., which detachments they agree on and which they disagree on.

Unwanted Options. After the voting process, each agent has a weighting function defined by a shared specification. Each of these weighting functions can be used to create detachments. Therefore, one can also compare the detachments separately. It is possible that the detachments derived from each agent's votes would render an option impermissible, but when combined into a single system, the option becomes permissible.

Example 10. $O = \{o, o_1, o_2\}$, $R = \{(g, o), (g_1, o_1), (g_2, o_2)\}$

$D_{w_1}(R, o, o_1, \{(g, o), (g_1, o_1)\}, c_1) = (-, v_1)$

$\sum_{r \in RR_o} jw_1(R, o, o_1, c_1, r) = jw_1(o, o_1, (g, o)) = 1$

$\sum_{r \in RR_{o_1}} rw_1(R, o, o_1, c_1, r) = rw_1(o, o_1, (g_1, o_1)) = 2$

$D_{w_1}(R, o, o_2, \{(g, o), (g_2, o_2)\}, c_1) = (+, v_2)$

$\sum_{r \in RR_o} jw_1(R, o, o_2, c_1, r) = jw_1(o, o_2, (g, o)) = 2$

$\sum_{r \in RR_{o_2}} rw_1(R, o, o_2, c_1, r) = rw_1(o, o_2, (g_2, o_2)) = 1$

According to the weighting function of the agent with w_1, option o is impermissible relative to option o_1 and is permissible relative to the option o_2. The dynamic scale will determine that o is impermissible.

$D_{w_2}(R, o, o_1, \{(g, o), (g_1, o_1)\}, c_2) = (-, v_3)$
$\sum_{r \in RR_o} jw_2(R, o, o_1, c_2, r) = jw_2(o, o_1, (g, o)) = 2$
$\sum_{r \in RR_{o_1}} rw_2(R, o, o_1, c_2, r) = rw_2(o, o_1, (g_1, o_1)) = 1$

$D_{w_2}(R, o, o_2, \{(g, o), (g_2, o_2)\}, c_2) = (+, v_4)$
$\sum_{r \in RR_o} jw_2(R, o, o_2, c_2, r) = jw_2(o, o_2, (g, o)) = 1$
$\sum_{r \in RR_{o_2}} rw_2(R, o, o_2, c_2, r) = rw_2(o, o_2, (g_2, o_2)) = 2$

According to the weighting function of agent 2*, the option* o *is permissible relative to option* o_1 *and impermissible relative to option* o_2*. The dynamic scale will determine that* o *is impermissible.*

$D_{w_1+w_2}(R, o, o_1, \{(g, o), (g_1, o_1)\}, c^*) = (0, v_5)$
$\sum_{r \in RR_o} jw_1(R, o, o_1, c_1, r) + \sum_{r \in RR_o} jw_2(R, o, o_1, c_2, r) = jw_1(o, o_1, (g, o)) + jw_2(o, o_1, (g, o)) = 1 + 2 = 3$
$\sum_{r \in RR_{o_1}} rw_1(R, o, o_1, c_1, r) + \sum_{r \in RR_{o_1}} rw_2(R, o, o_1, c_2, r) = rw_1(o, o_1, (g_1, o_1)) + rw_2(o, o_1, (g_1, o_1)) = 2 + 1 = 3$

$D_{w_1+w_2}(R, o, o_2, \{(g, o), (g_2, o_2)\}, c^*) = (0, v_6)$
$\sum_{r \in RR_o} jw_1(R, o, o_2, c_1, r) + \sum_{r \in RR_o} jw_2(R, o, o_2, c_2, r) = jw_1(o, o_2, (g, o)) + jw_2(o, o_2, (g, o)) = 2 + 1 = 3$
$\sum_{r \in RR_{o_2}} rw_1(R, o, o_2, c_1, r) + \sum_{r \in RR_{o_2}} rw_2(R, o, o_2, c_2, r) = rw_1(o, o_2, (g_2, o_2)) + rw_2(o, o_2, (g_2, o_2)) = 1 + 2 = 3$

When the weight functions of agent 1 and agent 2 are combined, the option o *is permissible relative to option* o_1 *and permissible relative to option* o_2*. The dynamic scale will determine that* o *is permissible.*

This setting shows that although agent 1 and agent 2 individually consider option o *impermissible, they collectively permit it.*

This result can be interpreted in multiple ways. It may be beneficial, but it may also be problematic. If, in the shared system, nothing else is permissible, it might be worth finding at least one permissible option, even if no manager would permit it individually. This is undesirable if the managers do not favor the option, but due to only interacting through weights, are unable to clarify that the option is not good. Then they may collectively commit themselves to an option none of them actually wants. It is possible to address this in the code, i.e., make options impermissible when each manager separately would consider impermissible, so that they cannot be detached as permissible.

At present, it is not clear whether the opposite is also possible i.e., the managers would all independently permit the option; collectively, they would forbid it. It may be sufficient to simply switch the weights for justifying and requiring. However, the dynamic scale exhibits asymmetric behavior with respect to permissible and impermissible values due to its handling of 0 values, and it primarily focuses on the permission scale.

Top down restrictions. Outside the IHiBO-NB process, there may also be rules and regulations that impose restrictions on detachments. In this case, some detachments may have to be imposed on the options, preventing some of them from participating in the weighting process. If the top-down restrictions are expressed in terms of weighted reasons within the balancing process, then they may not align with the assumed detachment. This is not currently implemented in IHiBO-NB. However, in a full dynamic dual-scale IHiBO contract, this can be added without difficulty.

Localization of detachments. The final topic to be discussed here is the epistemic horizon of voters. Each voter assigns weights to the reasons they consider relevant, given the options they perceive in their context. However, the reasons and options considered by each voting manager may differ greatly. It could be that the managers' sets of options are pairwise disjoint or that they completely overlap. If they are disjoint, and if the reasons against ($-$ polarity) are not implemented, the result is that the detachments of each voting manager are effectively patched together into a single system without direct interaction. The reasons against, i.e., $-$ polarity, include reasons of the form $(g, \neg o)$, which allows one to express opposition to o even without knowing which specific alternative options are available.

One can also consider this process from the opposite direction, they can choose detachments and then find weights that align with that. A manager may implement one or multiple detachments from another agent in an area where they do not yet have any detachments. The idea here is that options and contexts can be new to an agent. In this case, they may simply incorporate specific detachments from other agents that cover those options or contexts into their detachment function. This can happen in a way that does not alter the detachments already present in their detachment function. However, when further updates are made, new options and contexts may influence the already present detachments.

For the output, it does matter which options are considered. It is natural to assume that there are relations among the options. Consider, for example, buying a quantity of a stock. Three quantities, A, B, and C, will be compared in size. So, if $A < B < C$, it is perhaps counterintuitive if A and C are allowed but not B. However, this cannot necessarily be enforced through the input. One can let their weight function express this, or add a particular functionality to IHiBO-NB to account for it. However, in ideal conditions, IHiBO-NB is not given additional work because of the issue of GAS costs. Therefore, letting IHiBO-NB stick to merely solving the permission scale of a particular option in this sense is a way to avoid this issue.

5.3.2 A closer look at the detachment process

As shown in the previous chapters, given weights or a detachment relation, one can be derived from the other via a balancing operator.

Crux and Context

Inside an IHiBO-NB instance, reasons are assigned weights. The reasons are there because a fund manager voted for them. The options are also there because they have been made available in the system. Alternatively, an IHiBO-NB instance can be created with a dynamic dual scale and a default option, for example, “do nothing”.

Context, Options and Grounds (cogs). The reasons inside IHiBO-NB can be referred to as reasons because they are reasons in the strict sense; however, it may be sensible to call them something else, for example, cogs, since they incorporate context. One can even say that they are not only

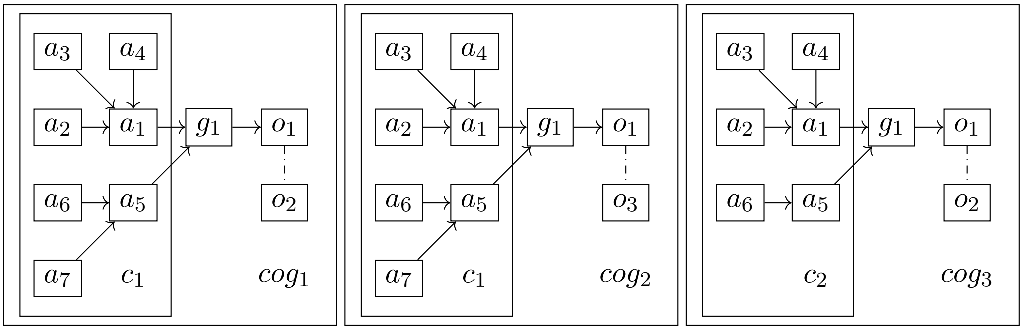


Figure 5.1: A structural depiction of cogs

context-dependent but also reasons expanded by contextual features. These contextual elements are undercutters, attenuators, and amplifiers as structurally described in (Streit, 2024; Streit et al., 2023). Furthermore, the term "cog" emphasizes that it is a cog in the system, which it is (the word also occurs in cognitive science to represent a kind of cognitive unit).

What is important here is that the element has a particular contextual background. This means that there can be two reasons (g, o) with different contextualizations. Depending on the amount of context that is shared, the two "cogs" could be considered equivalent. However, the agent votes based on the context it perceives and assigns weights that reflect that context. Part of the context is hidden because it does not pass through the voting process and is thus represented only by the weight. We can also understand a cog as being dependent on the alternative options under consideration. So two cogs can represent the same reason and have the same contextualization; however, they have different weights due to different alternative options. In summary, a cog is a carrier of weight, and all carriers of weight can be seen as cogs.

Figure 5.1 shows three cogs. We see the same reason in each cog namely (g_1, o_1). However cog_1 and cog_2 have a cog that participates in a different pairwise comparison. Namely cog_1 has a reason for o_1 but is in a pairwise comparison with o_2, and cog_2 has a reason for o_1 but is in a pairwise comparison with o_3. The third cog cog_3 has a reason for o_1 in comparison with o_2 however is different from cog_1 because its context is c_2 while the context of cog_1 is c_1. Here the context is depicted by elements a. Ideally we can see these elements as reasons supporting the later elements thus accomplishing the weight of the reason in the cog.

Issue, and multi-agent deliberation. When combining the cogs of all these voting agents, a problem is encountered that was introduced in the previous subsection, namely: do all the agents know all the options? The IHiBO-NB dynamic dual-scale instance begins to determine the deontic status of options available in a particular scenario even though participating agents may not consider all of them. In addition, setting the reputation score can become complicated if the options are not known in advance. It may therefore be sensible to regard the IHiBO-NB dynamic dual scale instance as relating to an "issue". Here, an issue is a scenario in which one or more options are considered, and each agent has a particular reputation or trust factor. The issue may be a set containing a single option or many options to allow for greater flexibility.

As noted earlier, some reasons or cogs may be inconsistent when combined. For example, some agents may consider a ground to be a reason for an option, while others consider it a reason against that option. This depends on the agent's underlying consistentency criteria. The benefit of cogs and numerical balancing is that consistency is not required; the balance scale will find a detachment value regardless. This highlights a complex problem in which reputation specification, crux (i.e., options), context, and domain all play a part. These design choices can cause the detachment process in the IHiBO-NB to render one option permissible rather than another. Therefore, whether IHiBO-NB considers single options or sets of options at a time in a decision process is an important design choice.

Operators

The other part inside an IHiBO-NB instance is the balancing process. The balancing function, together with the weighting function implements a detachment function. The balancing function can be understood as consisting of two parts: the aggregator and the activator. In a dual scale, the aggregator is the addition function, and the activator is the sign function. There is also a comparison function; however, this part is not subject to modification, as the only way to compare numbers is by using $<$, $>$, and $=$.

Aggregators. In the chapters on principle-based analysis, multiple aggregators were introduced: addition, multiplication, and max. Variations on these aggregators were also introduced: relative weight satisficing and absolute weight satisficing. In addition, scalars can be introduced to multiply the weight of different kinds of reasons by different scalar values.

Activators. The activator used throughout this dissertation is the sign function:

$$sgn(x) = \begin{cases} -1, & \text{if } x < 0 \\ 0, & \text{if } x = 0 \\ 1, & \text{if } x > 0 \end{cases}$$

In the application domain of determining the deontic status of an option, only the above values are useful, therefore this function is the most sensible choice. Nevertheless, it is possible to use a different function. Common activation functions include the logistic function, ReLU (Rectified Linear Unit), the Gaussian function, and arctangent function. Changing the activation function will have consequences for interpretation and requires additional explanation. The sign function has only three outputs, but some alternative activation functions have a continuous output domain, for example, $[-1, 1]$. In that case, the meaning of a detached value such as -0.5 should be explained and interpreted.

5.3.3 A closer look at IHiBO-NB output

Trust is generated on the output side of IHiBO-NB through documentation. Managers should also trust the process and therefore be able to select a process they trust. IHiBO can be used to share input information among participants, thereby improving the process (i.e., creating common knowledge of reasons and/or options through information sharing). Or alternatively not share and keep information private, when users don't trust each other. The main output of IHiBO-NB is a kind of receipt of the detachment process that at minimum specifies what is permitted and what is not. Of course, this also depends on the input format and the type of detachment chosen.

Receipts

In a supermarket or a restaurant, whenever people make a purchase, they receive a receipt for the transaction. The receipt lists the items purchased and their individual prices, with the total listed below the items. Some additional information can be found such as the staff name or the date and time of purchase. The output of IHiBO-NB can be viewed to some extent as a type of receipt of its internal process. However, what is written on the receipt could potentially reveal confidential information, so users should carefully consider what kind of output they want. The documentation can contain each vote and may possibly, but not necessarily, include the voters' identities. This information can be used to generate each individual's weight function and detachments.

Votes. Writing the individual votes of each fund manager and their identity on the receipt is prima facie a good idea. The receipt then shows exactly the contribution of each fund manager. If a fund manager deliberately chooses high weights, this will appear on the receipt. Also, if a fund manager has been given a trust factor, it would be good to document this, since it could explain the magnitude of the weights. However, with this information comes privacy concerns. The reasons may need to be inaccessible due to trade secrets. In addition fund managers should, among other things, have the freedom to vote without worrying that their votes will be visible to everyone.

Voting in IHiBO-NB differs from how voting usually works in judgment aggregation (Grossi & Pigozzi, 2022). In judgment aggregation, agents vote on judgments or propositions representing options. In contrast, IHiBO-NB employs dual-weight reasons voting, where reasons contribute to the permission status of voters rather than agents voting directly for a particular option.

Reasons. Documenting the reasons is important, since the ground specifies the information used to support reasons, and the option specifies the action under consideration. Furthermore, reasons in IHiBO-NB are more than a ground and an option. They also come with context, and it needs to be understood how much of the context is "forgotten". To some extent, there will be a cutoff point for context, and the voted weights will represent the background information. As such, some fund managers will vote for the same reason, even if they contextualize that reason differently. The level of detail also needs to be chosen. A starting point is to document only the reasons and leave out the context. However, this may not be good enough. Alternatively, it may be possible to provide more specific information about the chosen grounds by using pointers to other documents on the blockchain.

Detachments. The most important part of the receipt is the detachments. Without them, there is no real purpose in running the IHiBO-NB process, since downstream systems will require information about which options are permitted and which are not. If multiple balancing procedures are implemented, information about the chosen balancing operator can also be added.

Here again, the idea of external detachments should be noted. If the external system in which the managers operate forbids an option, they will have no choice but to choose options other than those that system forbids. In fund management, if there is insufficient liquidity to execute a purchase, the external system will not accept such a trade.

Smart contract design and pipeline

The smart contract in IHiBO-NB that determines which options are permitted can become part of a pipeline of multiple smart contracts. Downstream, there may be smart contracts that automatically start trading by determining optimal strategies given the permissions. The important point is that the detachment smart contract fits into a larger code ecosystem. Furthermore, the detachment process can be used in contexts other than collaborative fund management decision-making.

Private, public content. The information provided to IHiBO-NB is documented in a particular way, and some information may be lost as a result of this process. The information stored inside IHiBO-NB can be made accessible as part of a smart contract design. More explicitly, functions and data inside the smart contract can be labeled public or private. A public function can be accessed by anyone with access to the smart contract, while a private function cannot. It is also possible to create a list of addresses that are allowed to use specific functions. This means that some information can be shared among the participants in the process, while other information remains private. In circumstances where the parties do not want to share information, it is possible to choose a contract with more restrictions on the amount of information accessible to the participants. However, in collaborative circumstances, it might be preferable to grant all participants access to all information in the contract so that everyone is aware of all the reasons and options considered. If users want to use the system in more adversarial contexts, the functionality of masking information may make it easier for parties to present their reasons without revealing private information. This may improve the process of finding collective detachments in more adversarial cases.

Output as new input? Since a ground can be a reference to a document, the output of IHiBO-NB can also be used as a ground for a new reason for particular options in a subsequent IHiBO-NB instance. However, it is not possible to use the output of an IHiBO-NB process that is still ongoing. From the perspective of the philosophical background theory, this form of reuse is not coherent. In ethical theory, contexts are not typically partitioned, and outcomes are tied to particular scenarios. What is ethical in one scenario does not automatically generate new reasons in another. In ethics, the omniscient perspective is taken. Either something is ethical, or it is not. Reasons are ethical and not typically derived from previous outcomes. However, in the fund management scenario, due to the multi-agent context, incomplete information, and different (potentially inconsistent) contextualizations of reasons, such types of reasons are plausible and resembles jurisprudential or precedent-type reasoning.

This is already a step away from the theory. In the fund management use case, it may also be valuable to refer to previous outcomes of an IHiBO-NB process. However, it may also be possible to let IHiBO-NB function as a participant in another IHiBO-NB instance. This could simply be for memory purposes, to keep particular content—reasons and options—in a follow-up instance. Perhaps other reasons can be found for doing it. In the fund management use case, the output receipt needs to have the same structure as a vote, that is, a ground-option pair together with associated weights. In general, it is not straightforward to automatically interpret this as a vote in another IHiBO-NB process. A detachment output is of the form: $((R, o), v)$. This can simply be written as: (R, o, v), meaning that the set of reasons counts towards valuation v for option o. Then, since it is associated with option o, it may be reused through some mechanism as a reason in a new IHiBO-NB balancing process concerning option o. In IHiBO-NB, the output takes the form of a weight magnitude that represents the net contribution for each polarity. This way, the output of a new IHiBO-NB instance can be reused as a reason with a weight derived from the previous voting process.

5.3.4 A closer look at the use-case

Financial fund management, also known as portfolio management, is the main use case for IHiBO-NB. IHiBO-NB can partly function as, or be understood as, a receipt system. In principle, the use case entails collaboration among the users (managers); however, an adversarial setting is also possible. Furthermore, the idea of a meta-reason has emerged; this is a reason that incorporates the outcome of IHiBO-NB as ground for an option. In this case, IHiBO-NB gains some level of agency as it is treated as a voter or one of the managers who is contributing to the process, interacting with the new IHiBO-NB process through reuse of its output. Lastly, since the approach concerns balancing reasons in this section, the portfolio management use case is considered in the context of balancing profitability with sustainability.

The idea of robo-judge is not considered, even if that would be reasonable. Robo-judges is a sensitive topic, and the balancing discussed here is not directly grounded in judges' reasoning. Even if it were, a direct comparison with reasoning in court cases would be complex. Work in this direction does exist; for example, a short article (Sobek & Montag, 2018) investigating a computational approach to balancing in proportionality analysis. Proportionality is a sensitive and complex subject in its own right. In this dissertation, the no explicit contribution is made to these topics.

Financial fund/portfolio management

When IHiBO was introduced, it was described as an application designed to improve the relationship between fund managers and their investors. The way IHiBO accomplishes this is by documenting the information fund managers take into account when making decisions. This improves the trust relationship with investors by giving them a guarantee that this information is retrievable and by clarifying how it is documented. This is also good for fund managers, as they have a contractual obligation to provide this information, and automation therefore relieves them of the burden of doing so manually. However, in more practical terms, how does this process work?

In the current version of IHiBO-NB, the first thing to note is that the options can be interpreted in different ways. It may be simply buy, sell, or hold 1 unit of a particular stock. Alternatively, an option could be about a specific number of shares of a stock. However, in a dynamic dual scale, the options could also specify the entire content of the portfolio; in that case, choosing a permitted option would, in principle, mean that the entire current portfolio is exchanged for the portfolio specified in the option. Of course, it is neither necessary nor efficient to sell elements that are present in both portfolios. For this example, it is easiest to understand options as potential complete portfolios (maximalism).

Secondly, although managers make the choice, they must take into account relevant laws. Neither the managers nor the investors can influence or contravene laws. Laws are a given, and they take precedence over any consideration of whether options are permitted by managers. Managers can either make the laws part of the IHiBO-NB process or choose not to introduce options that are not permitted by law. It may sometimes be appropriate to consider not adhering to some laws; it is better to potentially pay a fine than lose the entire portfolio. But typically, laws are reasons for permitting particular options and forbidding others. Managers, or an appointed legal expert manager, can identify weights that align well with legal reasons. A balance can be found such that laws are generally complied with but leeway is given in particularly dire circumstances.

Third, the cost of executing options is somewhat difficult to estimate, as stock prices go up and down; an option may be financially feasible at one time but not at another. In practical terms, if an option is considered permitted at some time and then chosen for execution, but there is not sufficient liquidity at the time of trading, then the trade cannot be executed. Weight-wise, this can also be difficult to represent, because the justifying weight for having sufficient funds and the requiring weight for needing sufficient funds are put on the scales alongside other reasons. If there are many other reasons, and their aggregate constitutes sufficient justifying weight, the justification threshold is reached. To ensure that does not happen when liquidity is insufficient, the required weight for needing sufficient funds and the justifying weight for having sufficient funds should be equal but also high enough that the only way to exceed the requiring weight is by having sufficient liquidity.

The book "Managing Investment Portfolios" by Maginn et al. (2007) offers a detailed account of the portfolio management process. This process starts with the Investment Statement Policy (IPS). This is a document that fund managers draft for a new investor, clearly setting out the objectives and constraints of that investor. The steps of the process are the following.

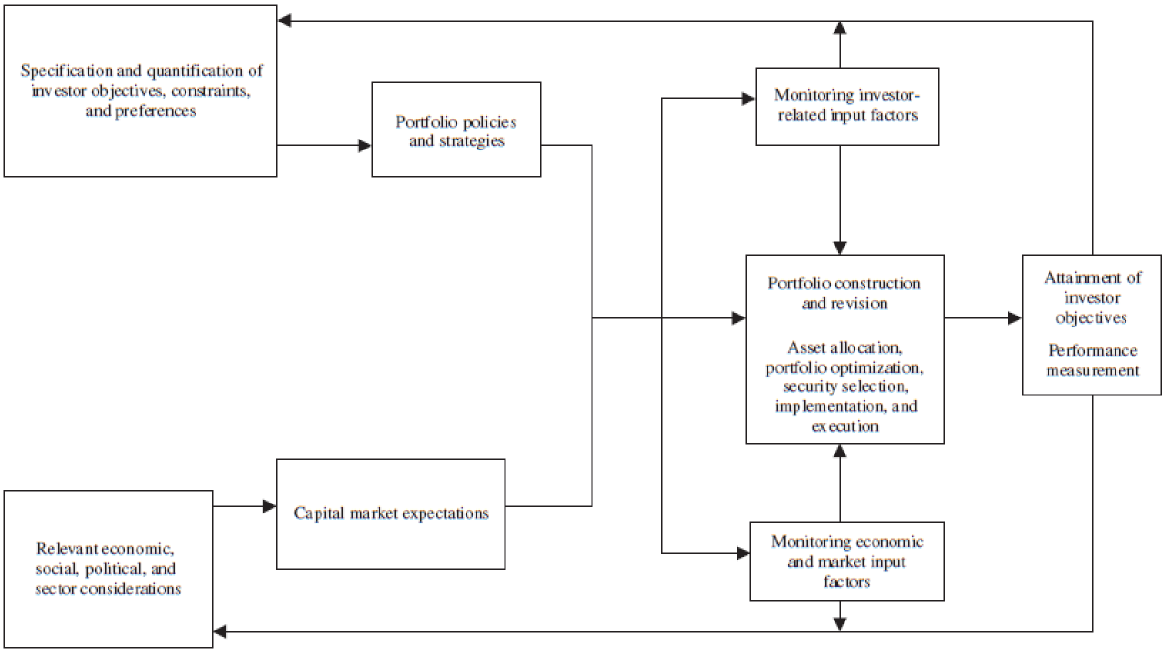


Figure 5.2: The portfolio management process

- The planning step
- The execution step
- The feedback step

The planning step involves creating the IPS but mainly consists of selecting stocks that the investors are interested in and that are in line with the rest of the IPS (investor policy statement). In the execution step, portfolio decisions are implemented, primarily through portfolio construction and modification. Then there is the feedback step: the portfolio's content performance is evaluated, and the feedback is acted upon. Feedback is used to rebalance the portfolio in response to changes and to keep it optimal and aligned with the IPS. This is important because not only does the market change, but it may also react to managers' decisions. Figure 5.2 from "Managing Investment Portfolios" by Maginn et al. (2007) shows a more detailed diagram of the process.

Today, many sub-processes of the portfolio management process are digital. The markets are digitally organized, and metrics are computed by computers at every available time point (tick) of the market. The digitalization and automation of decision-making in markets have been ongoing for a long time. Automated trading, or algorithmic trading as it is often called, even led to the so-called flash crash of 6 May 2010 (Paulin et al., 2019). A system like IHiBO(-NB) could have documented more of what happened during that flash crash, giving judges and investors a clearer image of the sequence of events later on.

To stay competitive, managers need to be able to trade on every tick of the market. The only way to do this is to automate the decision-making process. However, as the flash crash shows, this is not without risks. Therefore, fund managers will need to build more trust with investors and regulators. For regulators, it has also become more complicated to obtain the necessary information to know what is really happening. This is due to the sheer amount of data involved. It also reflects the diminishing information advantage of banks in a changing financial landscape with the emergence of bitcoin, among other factors.[1]

As digitalization and automation increase, sources of information may also become digital. Therefore, a set of virtual "fund managers" may be more accurate as the input source. However, "real" fund managers should still be the ones who determine what information is present. Even if there is a subscription to a service that provides packages of reasons and weight that are used as input to IHiBO(-NB), there will still be a fund manager who decided to allow input from this source into the IHiBO(-NB) instance. The IHiBO(-NB) instance might be password-protected; this may make it possible for someone to break into the system.

Multiple portfolios may be brought forward as possible options through the voting process, say 10 in this example: $o_1, o_2, \ldots, o_{10}$. This process of generating options should also be automated if managers want to be able to trade at every market tick (time point). Then, the voting process gives weighted reasons. These reasons are documented by IHiBO-NB together with the weights, potentially but not necessarily including who votes for which weights. In finance, many metrics can be used to assess a portfolio's performance. Then, the different portfolios can be compared using these metrics. If each option is evaluated using the same metric and the metric scales the weight, then the comparisons will factor in performance on that metric, with a multiplier indicating the metric's importance. However, this also introduces choices about how grounds factor into the weights. Is the reason a justifying reason, a requiring reason, or both? If it is a requiring reason, then it can be

[1] https://www.linkedin.com/pulse/banking-thin-ice-major-threats-banks-already-mftwf

understood as attacking alternative options. What if the metric becomes unreasonably high, and what if the metric can be negative? Managers will need to understand how these metrics, as grounds, are expressed in terms of the weights of reasons.

Once IHiBO determines which options are permitted, another system can determine the optimal path forward given the permitted options and initiate execution.

Setting a weight. Consider the setting of a weight for a reason. Say there is a ground that measures the ROI (return on investment) of a stock, and that ground is used as a reason to buy a particular amount of that stock. The formula is as follows: $ROI = (NetProfit/CostofInvestment) \times 100$. If the ROI is positive, i.e., $ROI > 0$, then the investment has earned more than it cost. A negative ROI, i.e., $ROI < 0$, means that the investment lost money since the net profit is negative. Thus, a positive ROI can be used as justification for investing in the stock, while a negative ROI creates a requirement not to invest in the stock.

$$r_1 = (\text{``}ROI > 0'', o); \qquad w(r_1) = (1, 0)$$

$$r_2 = (\text{``}ROI < 0'', \neg o); \qquad w(r_2) = (0, 1)$$

Alternatively, one may want the justification or requirement to scale with the ROI. In that case, the following weights can be set. Note that the reasons only have weight if the ground is satisfied.

$$w(r_1) = (ROI, 0) \qquad w(r_2) = (0, ROI)$$

However, attention should be given to the cost of the stock. If cost is also given weight and must be counterbalanced by liquidity, then a variable weight may become sufficiently large to offset the stock's cost.

$$r_3 = (cost = 500, \neg o) \qquad w(r_3) = (0, 1000)$$

$$r_4 = (liquidity > cost, o) \qquad w(r_4) = (1000, 0)$$

Given the example weights above, a point may be reached where $ROI \geq 1000$, which would counterbalance the requiring weight generated by the liquidity constraint.

The receipt system

A major part of the discussion so far has been understanding IHiBO(-NB) as a receipt system. This reading means that IHiBO(-NB) is a system that generates documentation of the information considered before a decision is made. It is similar to people's experience when buying groceries at the supermarket. When they pay, they accept the trade and receive a receipt. However, in the case of supermarkets, there is only one possible transation, since the price of groceries have already been determined. The receipt shows how the price of each item contributes to the final total price. Moreover, there is additional information on the receipt such as discounts, the time of payment, supermarket details, and perhaps the cashier at the front register.

Although the receipt provides much information, some things are not explained. The receipt does not explain why the price of pasta is as shown. Similarly, IHiBO can document weights but will not, in general, explain why the weights are as stated because only the voters themselves can explain why they voted as they did.

If the IHiBO(-NB) documentation is expanded and IHiBO provides documentation for each trade, a clearer picture can emerge. Returning to the supermarket analogy, this picture would explain the price of pasta by showing multiple receipts, namely, the receipt of the grain producer, the receipt of the factory, the receipt of the petrol cost associated with transporting the pasta to the supermarket, and perhaps a few additional receipts.

Conflict resolution

Consensus mechanisms occur in many situations and have been studied broadly. Blockchain is underpinned by consensus mechanisms that validate new blocks. In politics, consensus takes center stage, especially in democracies. However, the reason-based balancing model is relatively new, so voting on reasons has not been studied deeply. One may consider voting in relation to game theory; however, this tends to be about voting on options directly.

One can understand what IHiBO-NB implements as a consensus mechanism, though it may also be understood as conflict resolution. Conflict resolution works by letting stakeholders or managers give weight to the reasons they consider important. Using the weights for the reasons, it expresses which options they think should be permissible and impermissible. The parties aim to reach a consensus on which actions are jointly permitted. When parties meet for this purpose, the parties can be collaborative, as in the case of a team of fund managers. They can also be adversarial, for example, when two teams of fund managers trade with each other. However, it is also possible that the parties are neither clearly collaborative nor in opposition. In cases where the collective of agents has multiple orientations, it may be regarded as beyond adversarial and collaborative.

Collaborative cases. In collaborative cases, the goal is to pool the expertise of each manager/source to ensure the best outcomes for the group. Voting weights and reasons in this case are about ensuring that information important to the group is made available and that options that the group should not want to pursue are forbidden. The documentation may reveal that the weights assigned by a particular agent led to bad outcomes for the group. It is therefore important that agents vote with weights that are not unnecessarily large so that they can be counterbalanced by other important reasons. However, if they strongly believe an option is bad, they should cast a clear vote against it with larger weights. Meanwhile, they should make sure that their votes permit options they stand behind. Using smaller weights is often sensible to avoid making it needlessly difficult for other reasons to outweigh them. However, the weight should not be so small that they can be easily outweighed by other possibly less significant reasons in the given context. To mitigate risks, the group may agree that certain options should be filtered through contract design. This can be done because the agents are explicitly collaborating with each other.

Adversarial cases. In adversarial cases, the agents or groups of agents have opposing goals. However, even in this case, there is still likely to be some level of collaboration. Although each group wants the best deal for itself, they are still collaborating to some extent in order to find a trade that both parties can accept. In this situation, it would be abnormal to completely delegate the trade to voting weights; traders would not grant permission for a trade by trusting the weights fully. Voters vote by weight, so, in principle, they will not know what the contract permits beforehand. It may be that one party inflates the weight of their vote and overrules the boundaries of the trade set by the other party. However, depending on whether information can be extracted from the contract, the sets of options available to different managers may not be the same. This means that some voted reasons for some options will not relate to comparisons with other options, therefore some options may not be "attackable" by users inflating weights on their preferred option.

The contract will specify, given the weights and reasons voted upon, the options introduced that the voters have permitted. However, the actual trade will occur outside the IHiBO-NB instance via a separate contract. It may be that this contract will enforce the strict boundaries of one of the parties, with the result that the parties will still not find a trade even though some trades were permitted.

Beyond adversarial and collaborative cases. When the agents participating in the voting process are neither explicitly adversarial nor collaborative, this can be seen as agents interacting in a shared space [2] and committing to determining what can be collectively permitted. Thought experiments like the "Veil of Ignorance" come to mind. However, in this case, unlike in the "Veil of Ignorance", voters know their own position. What they do not have is access to the internal state of the consensus mechanism. Voters will need to represent their position as well as possible. They may even need to use triage (the practice of prioritizing critical cases), i.e., they should ask themselves "What can we let go of and what can't we let go of?". Triage often took place during the COVID pandemic when medical staff had to decide who should receive urgent medical care and whose care should be postponed.

Meta reasons

The examples provided show multiple types of content that could be in an IHiBO-NB contract. The point of the documentation is that the information that plays a role is referenced in the contract as a reason, and that its influence is quantified using weights. However, there may also be considerations that only come into play when considering potential dependencies. "If it were the case that A follows when C is executed, then reason r is against C". This type of reasoning goes beyond the idea that the grounds in the reasons are facts; they reference potential outcomes. If managers think such considerations play a role, they should document that.

In addition, the examples show that the number of times IHiBO can be instantiated will continue

[2] https://en.wikipedia.org/wiki/Shared_space

to grow. This is particularly true in the fund management use case, where the contract might be called at every market tick, potentially multiple times per second. However, the theory assumes that only a finite number of reasons are provided as input. The case of infinite reasons could also be studied.

Portfolio management: profitability and sustainability

Multiple dimensions of the system have been discussed. Some of these illustrate particular aspects of the developed system more clearly than the portfolio management use case could. Portfolio managers are shown to be agents who must express what they are attached to and to what degree in order to minimize risk and optimize value, and let detachment determine, through balancing, which options are permitted. The initial idea for IHiBO is that the oracle documents the elements considered; however, as more of the process becomes automated over time, execution may also be delegated to an optimizing algorithm. This usage opens up the possibility of making decisions at every tick of the market.

Fund managers can use various metrics to assess an asset's profitability. However, they may also want to consider the asset's sustainability. Sustainability may be interpreted in more than one way, including not only eco-friendliness but also long-term rather than short-term profitability. In addition, managers need to understand the market in ways that go beyond metrics. For example, they may need to understand that possible unexpected events could pose a risk to a particular stock. Managers will need to translate this into weighted reasons for the options so that the sequence of actions aligns with their investment agreement.

Portfolio management is about reducing risk and growing a portfolio's value. Therefore, the goals of profitability and sustainability may be in direct conflict. Consider two managers: one who optimizes for profitability and another who optimizes for sustainability. Both may find sets of options they accept, i.e., O_p and O_s, where p denotes profitable and s denotes sustainable. A simple solution, without balancing weighted reasons, would be to look at the intersection $O_p \cap O_s$ of the two sets. If both portfolio managers agree that the option is permissible, it is permissible. However, it may also be that the intersection is empty: $O_p \cap O_s = \emptyset$. If there is no option available as the next action, this is a problem, since doing nothing is also an option, and if that option is not in the intersection, it is not permitted. A solution might be to simply enforce that doing nothing is always allowed. However, if the portfolio is losing value, this may be the worst option.

The weighted reasons approach may provide sufficient leeway in this case, such that an option is found that both agents together can permit. Maybe only one of the managers accepts the option, or maybe neither one does. However, given that they require at least one option to move forward, they may need to trust not only the voted reasons and their associated weights but also the specification of the detachment process. Since all options can lead to a loss of value, they must be honest when specifying their weighted reasons. The reasons and their weights are documented, and this documentation is what creates accountability. Investors' trust is grounded in the Investment Policy Statement. In that document, the nature and scope of the investment agreement are set out. Managers primarily need to demonstrate compliance with the goals set out in the IPS. Investors know there is risk involved in investing; however, they ask managers to invest on their behalf. Investors would not be happy if value were lost because managers were unable to agree on the way forward, though choosing an option in cases where all options are projected to lose portfolio value is not, in itself, inexcusable. In such cases, the weighted reasons approach may be able to find a way forward.

However, what happens if the balancing-weighted-reasons approach also fails to find a permissible option? Since portfolio managers want to find the least bad option, they may consider an iterated voting process that stops when acceptable permissible options are found. When considering an iterated process, they may additionally impose the restriction that an option should not be accepted if on of the agents does not consider it permissible.

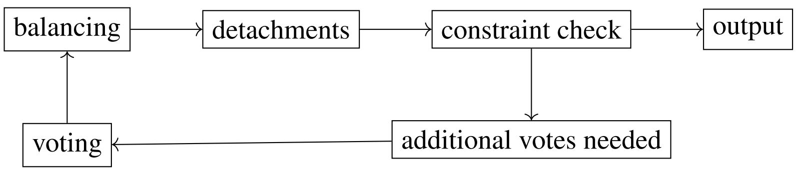


Figure 5.3: The iterative voting process

The managers start with the voting step shown in figure 5.3. Then, through balancing detachments, collective and individual detachments are found. These are then used in a constraint check. Multiple principles can be checked here, but an important one is that at least one permissible option should exist. The second sensible principle is that each individual should accept or reject the proposed permitted options on their own. If the principles used in the constraint check are satisfied, the permissible options can be output as viable. If one of the principles is not satisfied, an additional round of voting is needed. The managers vote again, adding new weights to the existing ones. To avoid a stalemate, depending on what information managers want to share, they can make the voted weights, reasons, and options common knowledge. This can also be done in the first round of voting before calculating the detachments. However, it may be good not to make everything common knowledge to limit the influence voters have on one another. The process continues until sufficient alignment among managers is achieved, ensuring that the output principles are satisfied. The process terminates, and the viable options are output as permissible options.

5.4 Recap, perspectives, alignment, and limitations

This section provides a recap of multiple points, compares perspectives, examines how different parts align, and notes the limitations.

5.4.1 Balancing

The dynamic dual scale is not the same as the numerical balancing model implemented in IHiBO-NB. However, to get the same output as a dynamic dual scale in IHiBO-NB, only a few balance scales are needed. It is not the case that IHiBO-NB explicitly implements these balance scales. However, IHiBO-NB implements only a single balance scale as a sub-implementation when users abstain from using $++$, $--$, and 0. The dynamic dual scale is, fundamentally, a system of balance scales. The foundation needed to build a dynamic dual-scale is in place. In addition, either code is needed to account for more options, or IHiBO needs to be instantiated multiple times. However, this requires significant additional administration. What IHiBO more accurately implements is a single scale, which is used in a dynamic dual scale as a permission scale. Recall that each option on the dynamic dual scale has a permission scale and a commitment scale for each pairwise comparison. Permission for an option depends on not losing any pairwise comparisons on the permission scale. Thus, given an option ϕ, the question is whether ϕ is permitted or not permitted. Considering this question on a dynamic dual scale is as simple as asking whether there is a pairwise comparison in which ϕ is not permitted. If ϕ is not permitted in a pairwise comparison, then this means that there is an alternative that requires ϕ not to occur. If a manager understands that this option exists, they will vote assign negative-polarity weights such that ϕ will not become permitted. In this way, the IHiBO-NB implementation can achieve the expressivity of a dynamic dual scale for each option.

5.4.2 Voting and "proportionality"

Even though the voting that IHiBO-NB introduces is a great addition, it also introduces a major problem. This problem may be called proportionality. Proportionality per se is not the problem; it is that the proportions of weight within a weight function encode particular detachments. The proportions need not have any meaning beyond the weight function encoding the detachments. However, when two weight functions are combined, the relative proportions between them determine the combined weight function and consequently also the detachment function.

A croissant on its own may appear to be of a "normal" size in a picture. However, when a reference item is placed in the picture next to the croissant, it may turn out that the croissant is not of "normal" size.

In the same way, the weights used in a weight function have meaning inside the system that is disjoint from the meaning given to the weights outside of the weight function. At least theoretically, the meaning cannot be guaranteed. Since weights determine the detachment, this requires attention. When the weights are combined, the underlying number system (e.g. the real numbers) and addition provide the method for finding the new weight function. However, this may cause the detachments encoded in each weight function to change into a different set of detachments. To some extent, this is desirable and the goal of aggregating; however, in this case, the conflict resolution is delegated to addition.

Normalization. The voting process, together with changes in reputation, gives managers feedback on how their input is taken into account in an iterative voting procedure. This process may be understood as normalization, in first instance in a procedural rather than numerical sense (however, the numerical sense may not be very distant). Normalization is commonly the process by which particular behaviors become more prevalent and then become the norm. As the process iterates, some detachments become stabilized due to the repeated input provided.

5.4.3 The reasons function

The principles are now mainly applied to the detachment function. However, there are more areas where principles can be applied. A start has been made on principles for weight functions, i.e., single-proportion weights and uniform weights. And some ideas for option principles have been introduced, namely maximalism and minimalism. Also, the reasons function may be subject to principles. Further principles may also be specified for the set of grounds.

The reasons function at the moment constrains the reason to be from the set $G \times O$ where $G \cap O = \emptyset$. However, this constraint may be relaxed such that $G \cap O \neq \emptyset$; in other words, there are elements that are both ground and option. When the reasons function allows this, graphs of the shape in figure 5.4 may be constructed. In this graph, there are four "pure" grounds, one "pure" option, and an element a_1 that is both a ground and an option. If the context detaches a_1 to $+$ at the dynamic level, then it can be a ground for option o_1.

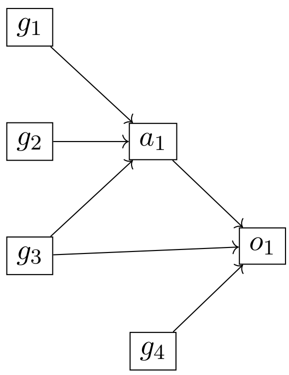


Figure 5.4: A hybrid ground-option

More complex systems can be envisaged, for example, those where $G = O$ is the case, see figure 5.5.

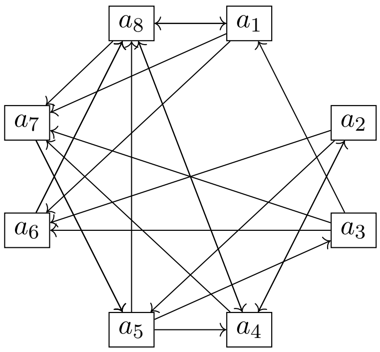


Figure 5.5: A more complex reasoning system than figure 5.4

The question of how to evaluate the deontic status of each element has not been addressed in detail. At the moment, one has to somehow use a dynamic dual scale. At first, specific elements may be considered true by default. Then the consequences of those elements that make the reasons true can be calculated, leading to particular options that have the $+$ value. However, since this procedure has not yet been attempted, it is unclear whether the dynamic dual scale can be applied to these elements without additional work.

To take it a step further, there may be some explicit elements that are solely grounds and some that are solely options. That is, there is additionally a set A of elements labeled a that are not exactly options, in the sense that they do not have an explicit action associated with them, but they are not exactly grounds either, since their acceptance is not a given. This is denoted as $A = G \cap O \neq \emptyset$.

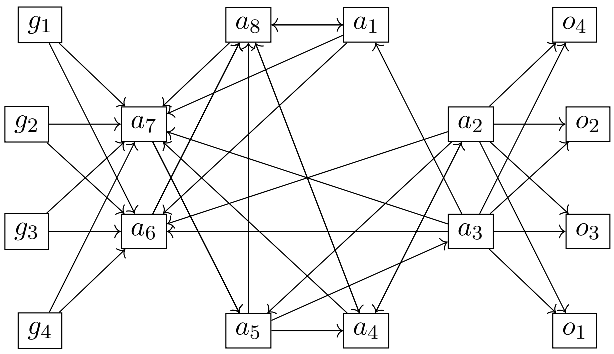


Figure 5.6: Grounds and options together with internal elements

Figure 5.6 shows a graph in which grounds, options, and "in-between" elements are used, creating an almost artificial neural network-type graph. However, this graph could also support abstract argumentation semantics. The specifics of this approach need to be tested and proven, specifically the expressivity and how the dynamic dual scale operates exactly on this graph.

The XOR gate. To make this more concrete, an XOR gate can be implemented. An XOR-gate is a function that takes two inputs, both of which are either true or false, and outputs true when one of them is true and the other is false. This function is often employed in the context of artificial neural networks and shows the usefulness of hidden layers. The XOR gate requires a hidden layer to be expressible.

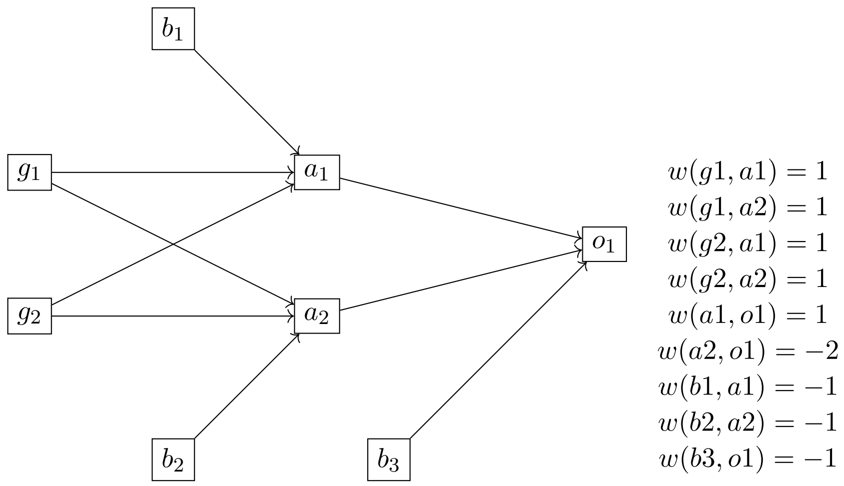


Figure 5.7: The XOR-gate in the artificial neural network approach

The gate should output o_1 as true for $(g_1 \vee g_2) \wedge \neg(g_1 \wedge g_2)$. Node a_1 expresses $g_1 \vee g_2$ and node a_2 expresses $(g_1 \wedge g_2)$. The bias nodes b_1 and b_2 shift the evaluation point from above or equal to 0 to a different point. The biases make it so that for a_1, one g needs to be true to be bigger than or equal to 0. And for a_2, the bias is -2, making it so that both g's need to be true. Then a_1 has weight 1 and a_2 has weight -1; the bias is also -1. This means that if a_1 is true, then o_1 is true; and if a_1 and a_2 are both true, then, due to the bias, the total is -1, so o_1 is false. In this case, the evaluation value of each node depends on the balance scale where the element is weighed against its negation. The reasons that play a role are those that can be constructed from introduced grounds and, via intermediate elements, are connected to the option being evaluated. The weights are specified on the right-hand side.

This example shows that much of the dynamic dual scale's functionality is not necessary to express artificial neural networks. That is, artificial neural networks are a subclass. What is not obvious is exactly how the intermediate nodes a_i fit into the reason-based balancing detachment function paradigm. It is also possible to understand IHiBO-NB as an artificial neural network node, as both are abstract implementations of agency, illustrating how the work in IHiBO contributes to modeling this problem.

When g_1 and g_2 are true, the following evaluation of D_w is found for a_1. $D_w(c, R, a_1, \neg a_1, RR) = D_w(a_1, \{(g_1, a_1), (g_2, a_1), (b_1, a_1)\}) = sgn(w(g_1, a_1)+w(g_2, a_1)+w(b_1, a_1)) = sgn(1+1-1) = 1$ Here, the second scale is omitted. This again illustrates how limited the parameter space required is. A single scale is sufficient to model the value given to $\neg a_1$. Negative values are also used in the computation and in the weights. These indicate the weight in the requiring scale of the permission scale for a_1. This can be understood as a weight that requires the alternative. It also shows how the context can be modeled with only reasons.

To interpret this in terms of formal argumentation, a few adjustments are needed. Justifying weight is not a standard part of argumentation; requiring weight appears to be the most appropriate way to model conflict. In this case, the requiring weights are given by the negative weights.

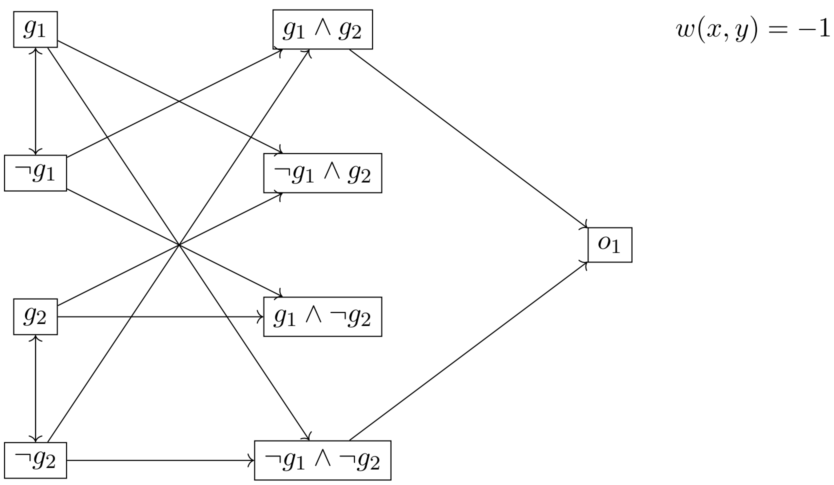


Figure 5.8: The XOR-gate in the argumentation approach

Each edge has a weight of -1. If a node's evaluation being ≥ 0 means that the node is true, then one attack is sufficient to make the node false. The grounds are incompatible with their negation, so truth values should be set before evaluation. The middle-layer arguments are also in conflict with each other, but the conflict relation has not been drawn here to avoid complicating the picture. The output option can only be made false when both g nodes are true, or both g nodes are false. The conjunctions can be made false when the two g nodes have incompatible status. We find $D_w(c, R, x, \neg x, RR) = sgn(y) = v$ either where $y = 0$, in which case we find $v = 0$ which maps to $+$, or where $y < 0$, in which case we find $v = -1$ which maps to $-$. Restricting the weights to only a requiring weight of 1 for the alternative is very limiting. Therefore a lot of the expressivity of the dual scale formalism is unused.

To express this directly as a detachment function, we may consider multiple contexts: $C = \{c_1 = \emptyset, c_2 = \{g_1\}, c_3 = \{g_2\}, c_4 = \{g_1, g_2\}\}$, with the set of options $O = \{o_1, o_2 = \neg o_1\}$, and the set of grounds $G = \{g_1, g_2\}$. From this, the set of constructible reasons is $\{r_1 = (g_1, o_1), r_2 = (g_2, o_1), r_3 = (g_1, o_2), r_4 = (g_2, o_2)\}$. The detachment function then considers the comparison of o_1 with its negation o_2 and is required to satisfy the following relations: $c_1 \vdash \neg o_1$, $c_2 \vdash o_1$, $c_3 \vdash o_1$, and $c_4 \vdash \neg o_1$. In this case, a reasons function needs to be set such that relevant reasons can be taken into consideration, and weights need to be set such that the specified expressivity is encoded. The question then is: are there suitable weights, reason sets R and relevant-reason sets RR such that $D_w(c_1, R_1, o_1, o_2, RR_1) = (-, +)$, $D_w(c_2, R_2, o_1, o_2, RR_2) = (+, -)$, $D_w(c_3, R_3, o_1, o_2, RR_3) = (+, -)$, and $D_w(c_4, R_4, o_1, o_2, RR_4) = (-, +)$. The results from the principle-based analysis indicate that such weights should exist provided that RR_i is non-empty for each i and contains at least one reason that is aligned with the resulting deontic status. Some weights and sets may be chosen to obtain the desired outcomes; however, this may be somewhat unsatisfying. Potentially, reasons such as $(\{g_1, g_2\}, o)$ are also required to encode this in a more satisfying way. If this is not wanted, one has to commit to finding weights for the reasons while choosing, for each context, the corresponding sets of reasons. This is difficult, especially for c_1, because there are no reasons present. An artificial reason may be constructed that provides the bias to reject o_1: $r_5 = (fake - ground, o_2)$, $w(c_1, o_1, o_2, r_5) = (1, 1)$; this is sufficient if r_5 is added to context c_1. Similarly, c_4 yields $(-, +)$ since reasons r_3 and r_4 play a role. They can simply be given the weights $w(c_4, o_1, o_2, r_3) = (1, 1)$ and $w(c_4, o_1, o_2, r_4) = (1, 1)$. Since both are reasons for o_2, they confirm the output valuation $(-, +)$, which is what was desired. For the other two contexts, the projected output is $(+, -)$. In both cases, there is one reason for o_1 and nothing else. To obtain o_1 in contexts c_2 and c_3, the weights must be assigned as $w(c_2, o_1, o_2, r_2) = (1, 1)$, $w(c_3, o_1, o_2, r_3) = (1, 1)$ respectively.

The aim of this exposition is not necessarily to show that one system is superior to the other. The aim is to show how each of these three systems can simulate the XOR gate and how these implementations can be expressed in terms of detachment. The starting point is to understand artificial neural networks and abstract argumentation frameworks within a dynamic dual scale. This allows comparison between the approaches, i.e. which parts of the dynamic dual scale they use and which they do not. Furthermore, a dynamic dual scale may show or explain why a particular approach lacks expressivity, as it may reveal that unused parts indicate such a deficit, in the same way that the single proportion assumption implies absence of dilemmas.

5.4.4 Portfolios

The detachment function can be viewed as a portfolio of detachments. This idea applies portfolio management to detachment functions, seeing the detachment function and its parts as something that can be managed and updated to be viable. This makes detachment something that managers evaluate based on value and decide to include or exclude. This also means that some detachments will be compared, and including one detachment may imply the exclusion of another.

5.4.5 Information cut-off point

The information cut-off point is an important concept in the decision-making documentation process. The grounds depict a piece of information that is understood as a reason for one or more options. The weight of the reason in a particular comparison and context determines its influence on the deontic status of the relevant option. What is documented are the weight and the information in the ground. However, the information may be limited, and may not answer all possible questions or explain every detail. The impact of the ground, therefore, is delegated to the weight, which is a function of the context; the complete context is not recorded.

5.4.6 Top-down and bottom-up approaches

Top-down and bottom-up approaches exist in many variations. These concepts have also been mentioned in this dissertation. One way in which it was introduced is that a detachment function can be updated by specifying valuations for a particular input. This can be understood as top-down when the detachment function must implement particular outputs in some way. The bottom-up approach can be found in the weight function giving rise to a detachment function. The detachment function is specified bottom-up because detachments follow from the weight function.

In social and legal theory, these terms are often used in relation to to laws and social behavior. Bottom-up norms are those that arise from the emergent properties of people interacting with one another. Top-down norms can be understood as laws and regulations that the state implements to address particular problems. Sometimes these coincide, and a natural norm is enshrined in the law.

In IHiBO-NB, the voting process leads to a normative judgment regarding an option. The norm regarding the option is the result of weighted inputs (reasons). The weighted reasons come from agents and, bottom-up, implement a set of norms and a detachment function. The reputation scores scale these weights, and thus the importance given to those agents' normative systems. If an agent's reputation score is large enough, it trivializes all other input. In this case, the agent's input serves as a top-down instantiation of its norms within the IHiBO-NB process. More often, no individual voter will have a sufficiently strong reputation to dominate the outcome. Therefore, the different normative sources will combine in a way that delivers a normative judgment that may align with that of some and potentially the majority of voters, but primarily with the weights assigned.

5.5 Additional elaborations

Some topics that merit discussion or reconsideration have not been discussed at all, or only in other contexts, This section addresses those topics.

5.5.1 Triple scale

There is a theory in (Tucker, 2025) that extends the dual-scale model to a triple-scale model that uses triple weights. The first weight is the justifying weight, the second is the requiring weight, and the third is the commending weight. The commending weight is intended to indicate that one permissible option is more commendable than another permissible option. This means that the commending weight can be used to choose a preferred option among equal options, i.e., among the permissible options. However, to the best of my knowledge, an exact balancing theory for this approach has not yet been developed. I imagine it would work by ranking the permitted options from most commendable to least commendable. Furthermore, on the face of it, this commending weight is not guaranteed to break a tie. Imagine two permissible options with an equal commending weight.

5.5.2 Online operations

An IHiBO-NB instance, as currently conceptualized, is used only once. When the voting process is completed, an output is generated, and the contract is terminated. If required again, a new instance of IHiBO-NB is initiated, used, and then terminated. However, it may also be possible to start an IHiBO-NB instance and let it run perpetually. In this case, users ask it to generate an output at a given time point and not shut down afterwards. This means that the system will have a longer history of votes and outputs. An additional functionality will be needed to clean the data structures inside the IHiBO-NB instance due to the available data structures being limited in size. To add weights to a sum, the sum needs to remain minimal. Otherwise, it will hit the maximum, and information will be lost. The benefit of this perpetual idea is that the contract may process its own output in the balancing process. The IHiBO-NB instance may then update the permissions at each new time point based on the changes made by the inputs. The system then actively balances the changes with the previous inputs. In this case, a good idea would be to ensure that "do nothing" is always available as an option to shut the system down. It is not clear if it such an option should always be a permitted option.

This highlights that balancing can be about not only what is important at a given time but also the conflicts between long-term and short-term goals. These priorities are shaped by adjusting weights based on feedback from evaluated actions. However, online learning systems are risky because they cannot be tested during operation as the weights change. In machine learning, it is common to train a model, then use it while it is no longer able to change its weights. Moreover, machine learning teaches us about vanishing gradients, in which some updates become very small and lose their effective impact due to the layers in the system. Another key point is that as updates are made to a weight function, some expressivity may be lost due to repeated updates that do not reinforce particular features, and as other weights change, this expressivity is further reduced.

5.5.3 More general detachment functions

The specified detachment functions implement a few constraints that can be weakened to obtain more general systems. The first constraint is that the set of grounds and the set of options are separate. If this constraint is weakened, the reasons function specifies a graph on so-called grounds and options, which together we can call hypotheticals. Second, if such a reason function is allowed, it may also be that different functions are used for aggregation, i.e., multiplication instead of addition. This allows the detachment functions to capture a more general class of computation graphs. Different aggregation functions are used at different nodes.

If a ground or option is an intermediate node in a graph, the interpretation changes. One may instead call the nodes of the graph arguments. A ground or option that functions as an intermediate node can then be understood as an argument representing a hypothesis. An argument is a proposition—something that is true or false—and can therefore be treated as a permissible or impermissible hypothesis given the rest of the graph.

However, this idea is not necessarily philosophically acceptable. It means that the outcome of a detachment process is used as the basis for a new option. It seems to say that a set of reasons for an option and the outcome of that option together become a reason for or against another option. There is no literature to support this idea. Mathematically, it may be more interesting to consider the case where the reason function is represented as a graph instead of a function. The principle of functionality, when applied to a graph, means that the graph behaves like a function, making the function a special case. This will also require additional computational work to determine exactly how the value of each intermediate node is determined during the node evaluation process induced by the graph.

5.5.4 Voting of weights

Let us go through the process of choosing weights for the reasons, or more explicitly, the cogs in a given decision problem. The first factor to consider is the premises of the situation underlying the joint-decision problem. Is the situation collaborative or adversarial, or perhaps something in between? Is the context common knowledge or private information? Are there restrictions in the form of principles applied to the weights that can be voted upon? What kind of balancing operator is used? The entire process should be well understood because the outcome is determined by the weighing process.

Once the process is clear, the reasons the agents wish to present and the weights they assign to them can be considered. First, the reasons are obtained. This involves several steps: gathering relevant grounds, gathering the options to be considered and, finally, constructing the reasons, i.e., establishing which ground is a reason for which option(s).

Before assigning weights, the multi-agent setting should also be considered. Other agents will also vote on the weights and present reasons. One agent may specify grounds and options that another agent may not. However, it is also possible that an agent perceives a ground, a reason, or an option but chooses not to include it in the balancing process. In addition to the weights, this part of the process determines which grounds, options, and reasons will end up on the "receipt" of the process. Of course, depending on the exact privacy setting, it may also be recorded who voted for which reasons, and what weight this carried in the process.

To determine which weights to assign, agents should deliberate which positions they are attached to. If an agent considers one reason to be more important than another, then the weights should express this. A good place to start is to rank all the reasons from the weightiest to the lightest reasons. It is also possible for reasons to be explicitly given the same weight. Ranking is intended to ensure that a reason that is considered more important is assigned a larger weight than a reason that is considered less important, while ensuring that equally important reasons have the same weight. Such a ranking would be useful, but this provides only a partial picture, since a reason does not simply receive a weight independently of the surrounding conditions. After that also the proportion of the weights relatively to each other should be determined, maybe a reason should have 3 times the weight of another. Where voting allows dual weights, i.e., justifying and requiring weights, agents need to consider whether to assign two separate weights or use single-weight proportions. Then, the weights are used in pairwise comparisons, so agents need to take into account the various alternative options under consideration in the assignment of weights. Here, it also becomes important how much knowledge agents have of these alternative options, and whether they are put in a position that allows them to consider unknown options i.e., options proposed by other agents that they are not aware of. The other agents may introduce new, unknown, or even inaccessible options (to some agents).

To properly assign weights, it is first necessary to construct the cogs from the reasons. The cogs are the contextualized reasons (reasons expanded by contextual features) for each pairwise comparison. These are the elements that have their weights explained by the context and alternative options. Depending on the number of options and reasons, the number of cogs can increase significantly compared to the number of original reasons.

Before an agent choose which weights to assign and voting them into the system, two additional elements can be introduced. The first is to develop further constraints in terms of a calculus that specifies constraints on the proportional and structural relationships between the weights assigned to different reasons. For example, the weight of reason A is at least twice the weight of reason B. Or, the weight of reason C is equal to the combined weight of reason D and E. The second element is to enable agents to explicitly specify the deontic status, i.e. which detachments they want to encode in the weight, they wish to assign to each option, i.e., which options they would permit and which ones they would forbid. This can be a useful tool for determining the eventual weights and how they relate to the deontic statuses they entail, particularly where the agent has a very strong opinion on certain options, i.e., they wish to veto an option.

Before submitting their votes, agents need to choose their weights. They do this under the understanding that once the votes are in, they will have no further opportunity to influence the outcome. The weights they select are the only influence agents have as the balancing procedure takes over. To understand the range of possible outcomes, consider two extremes. First, the assigned weights are relatively small, so removing them from the collective vote would not affect the outcome. One adverse consequence could be that an agent is held responsible for not adequately signaling that an option was highly risky. Alternatively, the weights assigned by one agent could be much larger than those of the other agents such that removing all the other agents' weights would yield the same outcome. Here, the risk is that a single agent effectively becomes responsible for a bad choice. To choose the weights, agents need to strike a balance between assigning sufficiently large weights that they are influential when needed and sufficiently low weights that they are overruled when needed. This can be understood as a triage process. The weights indicate the extent to which preferences are subject to being overridden.

Having a veto over an option may be difficult in this framework. Typically, the weights determine the deontic status of the options under consideration. However, an option that receives more positive than negative weights can be rendered impermissible by a sufficiently strong reason for an alternative. A strange possibility is that an agent might even create a made-up option just to kill another option on the dynamic level; but this will also be recorded. It is better to locate the veto outside the procedure. For example, in a legal compliance (balancing) process, agents should not go against the law. The laws exist outside the balancing process and ensure that illegal options are not accepted when executed.

5.5.5 Principles and the principal agent

By implementing principles as constraints on the voting, guarantees can be put in place for the detachment process. However, additional principle-based analysis should be performed on the multi-agent detachment used in IHiBO-NB. In addition, the reasons function may be extended to a graph where the current reasons function becomes a special case given by specified principles: a functional graph. This gives way to a flexible system where principles can be applied to select a particular collective detachment process. This provides guarantees on the detachment relation and trust is based on those guarantees. Beyond this, the documentation allows voters to assign sensible weights to ensure accountability. Moreover, reputation scores can be used by agents to adjust the magnitudes of input weights based on their performance history in an iterated version of the voting process. Principles may be found governing option sets, including minimalism and/or maximalism, and possibly other structural aspects of the system, thereby allowing for the construction of a system with deliberate properties.

Once more refined principles are found and understood, it may become possible to choose very specific principles to instantiate a highly specific detachment function. This approach can then be extended to defining and specifying the behavior of artificial agents in general. Thus by choosing principles, an approach is found to ensure that the principal can trust the agents based on these principles. In the most extreme case, we may reduce the amount of learning required by the system and instead use principles to specify agent behavior. This remains a distant goal.

5.6 Conclusion

This chapter discussed various topics related to the system constructed for this dissertation, starting with connections to the literature, and followed by a discussion of input structures, detachment processes, and the financial use case. The discussion highlighted how voting weights, reputation factors, and polarity influence the aggregation of reasons in multi-agent settings. Several caveats were identified, including normalization challenges, adversarial voting risks, and the complexity of merging heterogeneous weight functions.

The use-case studies demonstrated the framework's flexibility across contexts, from collaborative decision-making scenarios such as financial portfolio management to adversarial scenarios such as trade negotiations. These examples illustrated how reason-based detachment can provide transparent, auditable decisions while accommodating various contextual factors.

Overall, the discussion confirms that the framework is conceptually robust and adaptable in practice, though its effectiveness depends on careful governance of weights and context representation. Future work should focus on refining dynamic-scale mechanisms, improving interoperability with legal constraints, and extending the system for continuous, real-time operation.

Chapter 6

Conclusion

This dissertation looks at developments in artificial intelligence, focusing primarily on decision-making processes. In relation to this, three pieces of literature were introduced: the weight of reason (Tucker, 2025), detachment functions (Knoks & Van der Torre, 2023), and IHiBO (Yu et al., 2021). The goal of this dissertation was to synthesize them in order to contribute to trustworthy decision-making processes in the context of any type of AI systems. The weight of reason explains how reasons should be balanced if one sets out to attribute weights to reasons and then use those weights to derive the deontic status of options. The literature of detachment functions provides a framework for describing and studying such balancing operations formally. And IHiBO offers an application architecture that enables the application of numerical balancing of reasons to a use case (portfolio management). Thus, it was possible to implement numerical balancing of reasons for IHiBO. Using the framework and analysis developed, the relationship between the IHiBO-NB implemented and the dynamic dual scale was elucidated. Although it appears, at first glance, that IHiBO-NB does not implement a dynamic dual scale, closer investigation shows that it is able to simulate one if the managers give good input. The relation becomes clear when IHiBO-NB is understood as comparing an option with its negation. Since IHiBO-NB also includes a multi-agent aspect, it is not possible to maintain multiple dynamic dual-scale properties, such as comparativism and coherent contexts. The implementation shows promise with comparatively low computational cost and improved input precision. However, it also leaves considerable room for future work. Notably, there is the work of implementing the full and unbiased type of balancing of the dynamic dual scale. Then, there is the expansion of the options (or actions) considered at the same time, from a limited to an arbitrary finite set of options. And last but not least, there is the expansion of the formal comparative analysis, which could be extended to include a greater variety of systems and principles.

This dissertation concludes by summarizing its contributions: (1) a principle-based analysis of dual-scale detachment functions yielding results on formal expressiveness, (2) the development of a reason-based balancing function in the IHiBO blockchain oracle yielding a quantitative and qualitative comparison against abstract argumentation, and (3) discussion of the computational and conceptual advantages, caveats, and the combination of dynamic dual scale detachment with multi-agent voting. The dissertation emphasizes the potential of normative reasoning frameworks to enhance transparency and trust in AI-driven decision-making. Future research directions include expanding principle-based analysis, refining dynamic-scale models, integrating external obligations, and applying the framework to domains beyond finance.

The research presented in this dissertation addresses the challenge of trustworthy decision-making in multi-agent systems by combining formal normative reasoning with blockchain technology. A dual-scale detachment framework for weighted reasons was described and analyzed, and a smart contract implementation was tested, demonstrating computational efficiency compared to abstract argumentation. Principle-based analysis revealed key distinctions between detachment models, highlighting the expressive power of dual-scale reasoning and its ability to capture complex normative patterns. The integration of confidence and reputation parameters further enhances transparency and accountability in collaborative decision-making.

Beyond fund management, the proposed framework offers a foundation for applications that require auditable reasoning processes, such as regulatory compliance and the regulation of AI systems. Future work should explore richer aggregation operators, dynamic contexts, and interoperability between reason-based and argumentation-based systems. Extending the approach to handle interdependent options and long-term decision cycles could broaden its practical impact. By bridging formal theory and blockchain implementation, this dissertation contributes to the development of transparent, scalable, and normatively grounded decision-support systems.

6.1 Research Questions

In the introduction of this book, I raised the following question:

RQ. How can dual-scale detachment improve IHiBO's input/output expressivity, transparency, and on-chain feasibility?

I now provide the answer to this question, based on the results of this dissertation. However, to answer this question, three additional sub-questions were raised. Let us first reconsider those.

1. How can the dual-scale approach be analyzed as a detachment function?
2. How does on-chain instantiation of dual-scale detachment compare to the existing IHiBO-AA(Abstract Argumentation) smart contract?
3. What are the limitations of dual scale detachment in IHiBO?

Regarding Question 1, yes, a dual scale can be represented by a detachment function. However, the detachment function formalism needed some modification. Additional input parameters were required to ensure that the weights of reasons remain constant when reasons were removed or added to the evaluation of the balance scale. This was achieved by a) introducing the context in abstract terms in order to specify the weights of reasons in that context and b) using a reasons function to specify which reasons exist in that context. From there, the output was extended to output two values: one for the permission scale and one for the commitment scale. Additionally, the dynamic scale was specified using the detachment functions and a pairwise-comparison tournament. Using the modified detachment functions, a principle-based analysis was carried out. The principle-based analysis was not extended to allow direct comparison between dual-scale and abstract argumentation.

Regarding Question 2, the theory of balancing weighted reasons was used to develop an IHiBO smart contract. The IHiBO-NB smart contract is comparable in application area to IHiBO-AA. The current implementation of IHiBO-NB can, however, be viewed as a sub-problem of the problems that IHiBO-AA solves. Nevertheless, IHiBO-NB is substantially cheaper to execute comparable circumstances given that the managers can translate their position from abstract argumentation to weighted reasons. The output of IHiBO-NB is specified to generate a single action for a single stock (e.g., an ETF), whereas IHiBO-AA can output either multiple actions or no action for one stock, many stocks, or even full portfolios. IHiBO-NB also extends the input space with reputation, confidence, and polarity. Reputation (and confidence) can be implemented for IHiBO-AA; however, this already introduces minor issues due to its relation to polarity. One element missing in IHiBO-AA is the ability for managers to vote on the attack relation between arguments. IHiBO-NB implements reasons with polarity, allowing managers to vote in this case on only the weight of arguments or reasons with the intended polarity. This means that if they add weight through voting for the interpretation of an argument in order to use it for a particular action, then that weight cannot be used through the same argument to support a different interpretation. This could be further developed by splitting the weight into a reason weight and a polarity weight, making the resulting vote weight also subject to other interpretations.

The last question concerns the conceptualization of full (dynamic) dual-scale detachment implemented in IHiBO. A full comparison between abstract argumentation and (dynamic) dual scale remains to be done; at the moment, it is unclear whether one is expressively superior to the other or whether they are equivalent. The low cost of IHiBO-NB is promising, suggesting that full dynamic dual-scale detachment will be a low-cost implementation of a system with reasoning capability similar to AA while also offering managers additional input features. Further discussion shows that extending the internals of reason-based detachment to incorporate more complex reasons, reasons including their context, could enable it to specify AA and artificial neural networks directly. Also, a weakness was identified in the additional input space for managers. Namely, each manager can individually reject the permissibility of an option while they collectively support its permission. Additional ideas are considered to address such cases if users dislike such potential outcomes.

Given the answers to the sub-questions, the research question of this dissertation can be answered. To reiterate: How can dual-scale detachment improve IHiBO's input/output expressivity, transparency, and on-chain feasibility? The weighted-reason-based approach provides an answer to this question. It can improve each of the three features expressed in the question. By allowing managers to vote on reasons, they gain control over how their input is translated into outputs. This, combined with confidence and reputation, allows them greater expressiveness in how they see what is before them. This, in turn, improves the transparency that IHiBO can offer the principal or investor as they can more clearly see how a manager sees the influence of a ground on each particular option. Also, by using numbers, the Ethereum virtual machine requires much less computational resources than graph-based computation. This improves the on-chain feasibility of this numerical approach.

6.2 Overview

In chapter 2, the detachment function approach of (de Wit, Knoks, & Van der Torre, 2025a; Knoks & Van der Torre, 2023; Knoks et al., 2024) was presented and modified with a principle-based analysis. The dual-scale system analyzed is based on the model presented by Tucker (2025). The system is highly expressive, allowing other systems to be modeled by imposing principles such as single-proportion weights. Inspired by input-output logic (Makinson & Van der Torre, 2000), it uses tuples to model the normative status relationship between formulas. The principle-based analysis is consistent with Tucker (2025) observation that dual scales are more expressive than single scales, especially for dilemmas. Reason-based approaches to normative reasoning, originating with Horty (2007, 2012), represent a growing area of research. The analysis presented in the literature is limited in two ways. First, it does not cover or compare all the detachment functions defined or derivable from the definitions introduced. Secondly, more principles could be included in the analysis, for example, principles that are the precise opposite of those already introduced, such as anti-monotony as the opposite of monotony. This is important because it is only possible to prove that the principle is satisfied or rejected; however, rejecting a principle is not proof that the negation of the principle is always true. There may also be counterexamples to the negation of the principle. In this case, neither principle is fully characterized, and counterexamples can be constructed for each principle. Simply put, if we look at sets of real numbers, some fail the "only positive numbers" principle, but this does not mean they satisfy the "only negative numbers" principle because some subsets fail both.

In Chapter 3, a detachment function was applied in an explicit multi-agent context, using the IHiBO (Intelligent Human-input-based Blockchain Oracle) use case (de Wit, Yu, et al., 2025; Yu & Gabbay, 2022; Yu, Zichichi, Markovich, & Najjar, 2022; Yu, Zichichi, Réka, Amro, et al., 2022; Yu et al., 2021). The implementation of the IHiBO reason-based balancing model involves multiple new perspectives and is distinct from the dual-scale model introduced in Chapter 2. The chapter compares abstract argumentation theory with research on the numerical balancing of weighted reasons. A comparison is made, but it is not complete. A result on expressivity has not been established between abstract argumentation and Dynamic Dual Scale Detachment. Ideally, it should be determined whether they are expressively equivalent or one approach is more expressive than the other. Once that has been established, there should be another complexity-cost analysis showing which approach is less costly to execute. Even as parameters are added to the dynamic dual scale, it is likely to remain less costly because summing numbers is computationally less expensive than performing graph-based comparisons, as in AA semantics. Additionally, there are parallels between the IHiBO dual-scale detachment process and the operation of artificial neurons in artificial neural networks. However, this observation is preliminary as its usage has not been clarified. Nevertheless, it allows us to interpret the voted weights as weight updates, analogous to updates in an artificial neural network during the learning stage. Then, IHiBO documents are different on the blockchain and are expressed as a sequence of updates to the artificial neural network or another model. If the model makes an error, IHiBO records could be used to trace the sequence of events that led to the error.

In chapter 4, the resulting system and theory were discussed, mostly through observations of the system developed, with further observations. The whole system is flexible, enabling a *ceteris paribus* style of analysis by allowing the reasons used in a detachment to vary while keeping all other things the same. However, it also allows one to expand or narrow the context of a situation, although this was not studied explicitly. IHiBO also enables the idea of extending and contracting a group of agents, which was also not studied but could be introduced to the definitions of detachment. Furthermore, the inner workings of a detachment function can be simplified or complicated by introducing information sharing or masking. The consequences of sharing or masking information were highlighted; in particular, the choices available can facilitate collaborative rather than adversarial decision-making. The principle-based approach facilitates the initialization of a detachment relation and enables more detailed analysis. Voting was also studied, with a particular focus on the problems that arise when combining two or more weight functions. While such operations pose considerable foundational and conceptual problems, their impact can diminish over a sequence of decisions as parameters such as reputation and confidence stabilize. The use case was examined in depth, particularly with respect to the dynamic dual scale. It is a simulation of a financial market where agents use the dynamic dual scale. The challenges of employing this approach are discussed. Overall, the documentation system allows, if desired, the regulation of access to information and the inclusion of additional information.

Although the detachment functions introduced in this dissertation have not been formally proven to express abstract argumentation or artificial neural networks, their potential applications in these fields were demonstrated. The implementation of a proof-of-concept reason-based detachment function in IHiBO demonstrates that detachment functions can serve as a computational architecture for an agent (where IHiBO is the agent). In the final analysis, the future of this approach appears promising, but it depends on its adoption by other researchers; it could serve as a bridge between the sub-symbolic and symbolic approaches to artificial intelligence and multi-agent systems. The framework of detachment functions could therefore serve as a general approach to conducting principle-based analysis, making it possible to compare multiple different approaches that are, at the moment, difficult to compare with each other formally.

6.3 Future Work

Several directions for future work have been identified. The principle-based analysis in this case, specified to study the detachment relation, may be applied to other sets of detachment functions: the reasons graph, the space of options, the space of grounds, operator principles, and principles for context. This extension allows for more fine grained detail of understanding detachment. For example, understanding how things change when the options are not maximal but minimal.

The context can also be studied further, especially the inheritance relation on contexts. At the moment, the weights can be set almost completely freely. The only restriction is that a relevant reason should have a non-zero weight. However, context and weight may follow inheritance relations where a subset of the context is already specified by the superset of it. So, removing particular reasons in the context that modify relevant reasons influences the weight of those relevant reasons in a deterministic way, instead of allowing the weights to change to any other weight.

The framework may be studied in the perspective of infinity, where the current systems were mostly studied from the perspective of finite sets of reasons and agents, but also finite weight values. From a logic and mathematical perspective, this is a relevant and usual extension.

The impact of agent voting on detachment functions was studied by implementing it in IHiBO. Future work may involve formal study and principle-based analysis. In this case, the weight system used by a detachment function is one that is the consequence of adding multiple weight systems together. There is a potential for some of the principles to change their evaluation, as they have not been proven for this circumstance. And new principles to be found. Maybe one that relates to the caveat case where the collective permits what all individuals forbid.

For IHiBO, the reason-based balancing smart contract can be extended to dynamic dual-scale detachment. Later, with further research on detachment functions, a more general formulation of detachment functions can be implemented. Such that it allows users to choose the operator used for aggregating the weights but also to constrain the process to exist within particular principles.

Additionally, further functionalities can be developed for IHiBO users to address the gap between voting in argumentation and voting with weighted reasons. This is part of the larger effort to improve the tools that inform decision-making. For example, when users have abstract argumentation graphs on their side, how can they be transformed into input in the shape of weighted reasons. Writing such functions allows users to transition more easily from abstract argumentation to this weighted reason system.

The dissertation introduces the concept of the recursive construction of context, which could be a topic for further study. At the moment, the description of this will start with an atomic reason and then construct more complex reason using that initial reason and the outcomes of reason processes that follow that initial reason. Although the idea is interesting, something will need to be added in order for it to be worthwhile pursing and testing. At the moment many cases will just be generated while the purpose would be for it to find what is permitted.

To implement full-scale testing of the introduced IHiBO-NB system, the market simulation should be further developed so that it allows artificial agents to trade using detachment functions. This would provide a full sandbox for thorough study and verification of the system's operations. At the moment, the simulation only allows agents to buy or sell one stock at a time, and the agents do not interact with the market. Once implemented, reinforcement learning can be applied to the reputation of the agents. This may be a novel contribution to reinforcement learning as the reputation factor is different then for example Q-values in Q-learning.

Further understanding of detachment and artificial neural networks or other sub-symbolic approaches will be useful. For one, because it allows IHiBO users to also submit parts of such models to the IHiBO through voting. Furthermore, it may allow us to consider not only elements relevant for decisions, but also where the weight comes from. This is in particular relevant for the sub-symbolic context as often there many weights are set; however, exactly why the weights are what they are is unclear. By building a history of the model and showing which weights change due to which training data, we may understand more about why each weight in the system is what it is at any given time.

The research suggests a principle-based approach to principal-agent problems. Once more principles are specified, as well as principles for the other domains than the detachment relation. Then we may find ourselves in a position where we can use principles to instantiate agents, thus creating trustworthiness through the guaranteed principles. The principles may be applied to the input space, the reason space, the detachment relation, the output space, the weight function, and the context space. The detachment function approach allows all these dimensions to be interrelated in a sensible way. Verification of the principles the agent employs allows us to understand how it operates. Thus, further study of the principles allows agent construction to focus on agent instantiation using specific principles. By implementing these principles, the key part of the agent's functioning can give guarantees on agent execution. With a better understanding of the principles and potential applications, more precise guarantees can be provided in the future.

The handling of negation needs close attention when using the system. In the ethical circumstance, there is the benefit that all options that exist can be assumed to be "on the table". However, when considering the agent perspective, we need to respect the possibility that the viable options that exist are not part of the options that the agent considers. This means that negation has a different meaning. Negation can then be understood as the other options <u>inside</u> the domain that are not the particular option. In comparison, from an ethical perspective, there are all options that are not that particular option. Therefore, finding the answer to the juxtaposition of an option o with its negation $\neg o$ is complex. It can be understood as the result of the dynamic scale with respect to the option o, however, this would interpret it as o losing from a specific different option. Whereas the competition against $\neg o$ may attach different weights than those that occur in the comparisons used for the dynamic outcome and may also include more reasons. In this context, it becomes also more interesting to introduce voting weight on grounds. Since if you value eco-friendliness, presumably you would not change your support for this ground. However, this introduces the path of a different person interpreting a ground as a reason for options that the voter might not understand that ground as a reason for.

Lastly, two slightly more experimental options for future research can be considered. These are relations to quantum computing and recursive self improvement. The polarities in the reasons remind of the states of qubits potentially there is a way these are connected. Recursive self improvement or RSI is considered the path to artificial general intelligence (AGI) and artificial super intelligence (ASI). The idea of recursive part discussed in the dissertation builds a universe of reasons giving different weights to each contextualization of reasons. Building all these cases and finding good weights is hard but may coincide with the endeavor of RSI.

Epilogue

After reading through the dissertation myself, I was left not completely satisfied. Although much work was done, some parts were left incomplete. The line of the research is visible; however, at points, it was weakened. In this epilogue, I want to clarify this and leave the document complete. In the sense that the research line is clear. The discrepancy between the implemented and ideal implementations is clear. And to paint a picture of what I think this system can do.

First, I show a AI generated summary of the work. This gives the basic impression of the research thread. Then I will use this summary as a baseline for what readers will remember, to reflect on what was achieved and what was not. Lastly, I will paint a picture of what this proposed system should be capable of.

AI summary

In this section, a 4-part summary generated by generative AI is presented. The text is not modified at all. In the next section, I reflect on the generated summary.

Very Short Abstract

This dissertation develops a formal model of dual-scale detachment, a framework for weighing justifying and requiring reasons, and applies it to blockchain-based decision documentation. It conducts a principle-based analysis comparing dual-scale detachment with single-scale, uniform, and maximum models to assess expressivity and normative behaviour. The work then implements numerical balancing of weighted reasons inside IHiBO, a smart-contract–driven documentation system for fund-management decisions, showing significantly lower computational cost than the existing argumentation-based contract. Applications, limitations, and multi-agent issues—such as collective acceptance paradoxes—are explored, along with paths for future work.

Presentation-Ready Summary

This dissertation bridges normative reasoning, AI transparency, and blockchain technology. It begins by formalizing the recently introduced dual-scale model of weighted reasons, where each reason has two dimensions: justifying and requiring weight. Using the framework of detachment functions, it compares dual-scale reasoning against established models through a nine-principle axiomatic analysis, revealing that dual-scale is more expressive—capable of representing dilemmas and weak permissions.

The second part of the work applies this theory in practice by implementing a reason-balancing smart contract for the IHiBO decision-documentation protocol used in fund management. A benchmark shows that numerical balancing has linear GAS-cost growth, significantly outperforming the exponential cost of the existing abstract-argumentation smart contract.

Finally, the dissertation identifies conceptual limitations—most notably the possibility that a group may collectively accept a decision that no individual would endorse—and outlines future directions, including full dynamic dual-scale integration and deeper alignment with argumentation theory.

Chapter-by-Chapter Synopsis

Chapter 1 — Introduction. Introduces the central challenge of transparent multi-agent decision-making, especially in regulated environments such as fund management. Presents the research questions: (1) formalizing dual-scale detachment, (2) implementing it in IHiBO, (3) examining its limitations.

Chapter 2 — Additional Background. Explains the foundations relevant to the project: pillars of AI, symbolic vs. subsymbolic methods, input/output relations, principal–agent problems, blockchain oracles, and commensurability. This provides context for why reason-based systems are needed for transparency and control.

Chapter 3 — Balancing Weighted Reasons. Presents the formal machinery of reasons, grounds, contexts, weights, and detachment functions. Defines dual-scale detachment and compares it (via nine principles) to single-scale, uniform, and maximum models. Key result: dual-scale is more expressive and admits dilemmas; single-scale cannot.

Chapter 4 — Smart Contract Implementation. Implements a numerical balancing smart contract for IHiBO. Shows that this method has linear on-chain computational cost, contrasting sharply with the exponential cost of IHiBO's abstract-argumentation smart contract. Demonstrates feasibility for real blockchain deployment.

Chapter 5 — Discussion of Perspectives. Analyses conceptual consequences: dual-scale's expressivity, multi-agent limitations (e.g., collective acceptance vs. individual rejection), and connections to ethics, legal reasoning, argumentation theory, and neural-network analogies.

Chapter 6 — Conclusion. Summarizes contributions and outlines future work: dynamic dual-scale implementation, deeper comparison with argumentation-based methods, and extended multi-agent modelling. Appendices include proofs, full smart-contract code, trading simulations, and application case studies.

Layperson-Friendly Version

Imagine you want a team of people—or AIs—to make decisions that you can trust, especially when money or ethics are involved. You want them not just to decide, but to explain clearly why they decided. This dissertation builds a mathematical system for doing exactly that.

It starts with a simple idea: every reason someone might have can count for an option (like "buy the stock") or against it ("too risky"). But recent research shows each reason actually has two kinds of strength: one saying how much it justifies an action, and another saying how strongly it rules out alternatives. The dissertation formalizes this into a "dual-scale" model, which turns out to be much richer and more realistic than older models.

Next, this reasoning system is placed inside a blockchain-based tool that helps document how fund managers make decisions for investors. Why blockchain? Because once a decision and its reasons are recorded there, they can't be changed—and everyone can verify what happened.

A new smart contract was created that uses numerical weights instead of more complex argument graphs. Testing shows it is far cheaper and simpler to run on the Ethereum blockchain than the previous system.

Finally, the dissertation explores deeper issues: how groups sometimes reach decisions no individual supports, how to interpret dilemmas, and how this reasoning approach relates to ethics, AI safety, and even legal decision-making.

In short: the work provides a mathematically precise and computationally efficient way to weigh reasons, document decisions, and build trust between humans and intelligent systems.

Review of Generated summary by Author

Here, a reflection of the generated summary is given. The LLM has understood the dissertation in a particular way. It is possible that other readers are left with the same impression as the LLM. Here I reflect and adjust this impression.

The summary is actually quite good. And there is nothing that I would change explicitly. However, in order to see the fully realized version of what the system can be, the limits of the work should be noted. In a more positive contextualization, this can be seen as what work can still be done in the future.

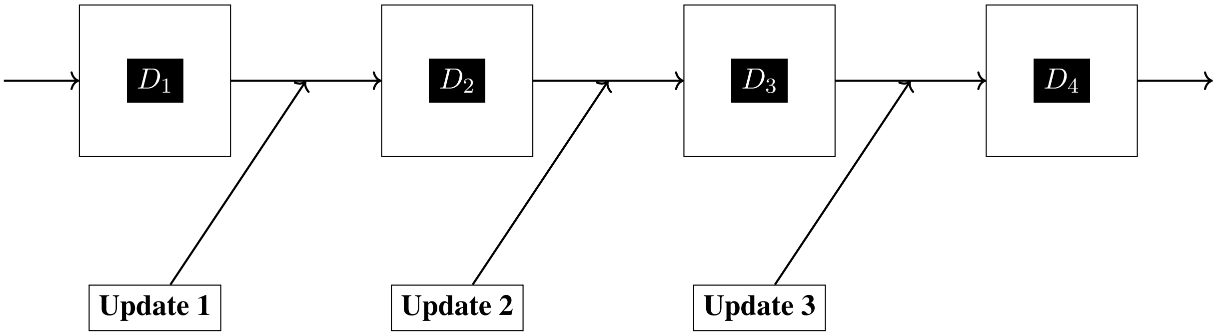
The strength of the chapter 3 is that it renders a dual scale as a detachment function and performs a principle-based analysis of 4 detachment functions using 9 principles. A weakness is that this analysis could entail more principles and more detachment functions. Also, multiple avenues are opened up, though not established concretely. Especially the idea of applying principles to other domains: grounds, options, reasons, function, context. These are also related to the connection being made between artificial neural networks and abstract argumentation. This connection is described, though it remains unproven. Also, an initial idea for specifying the context iteratively is noted, though it remains comparable to a footnote.

In chapter 4, weighted reasons are recorded on the Ethereum blockchain for the IHiBO architecture. The chapter shows that it functions and that the smart contract outperforms the abstract argumentation smart contract when applied to the same problem. The main limitation here is that full dynamic dual-scale detachment is not implemented. Furthermore, it is not completely clear what subset of the dynamic dual scale is implemented. Nevertheless, weighted reasons for IHiBO have been shown to be computationally efficient. Here, also, iterative instantiation of context is touched on again. In this case a IHiBO-NB instance, its output is understood as a new reason, thereby building context. Although the multi-agent dimension is discusses it would have been nice to have a more formal rendering of the operations involved in the multi-agent dimension. With this, I mean a calculus that extends detachment functions for the multi-agent domain. To some extent, this is obtained by understanding an agent as a carrier of a weight function.

Chapter 5 discusses the construction of the combination of IHiBO and Dual Scale Detachment. First, the comparison of chapters 3 and 4 with the literature is made. Then a discussion follows in which parts of IHiBO and detachment are discussed. Then multiple other topics follow, among which are caveats and constructions. As the summary notes, a noteworthy caveat is that the multi-agent dynamic dual-scale detachment allows a case in which each individual rejects an option but the group collectively permits it. Also, the idea of expanding the reasons function such that the dynamic dual scale can include artificial neural networks and abstract argumentation is further discussed. This discussion uses the XOR-gate to make the comparison more tangible. However, also here, a proof of this idea remains unwritten, though the example shows feasibility.

System Construction

Given the realized work and assuming the projected additional work, the full system and its capabilities can be imagined. Considering dynamic dual scale detachment as an IHiBO smart contract. In addition to the expressivity of abstract argumentation and artificial neural networks within dynamic dual-scale detachment. Such that context can be built from ground elements via the dynamic dual-scale detachment process, and each subsequent element is given weight based on previous weight conclusions of the dynamic dual scale. This means that it will be possible to model a wide variety of models inside a smart contract. Furthermore, the smart contract functionality allows managers to modify the model while tracking the changes and the manager who suggested them.



Depending on what is saved during the process, a reconstruction of past decision-making processes can be made. For example, if model D_3 made a particular decision, then there is a record of that decision on the blockchain. But also the updates after that decision and the current model. So, using the updates model, D_3 can be reconstructed, or the model can be built from the ground up using the updates that lead to D_3. Potentially, D_3 is accessible from the receipt. The updates related to D_3 can be considered to understand why D_3 was developed and, consequently, why it produced the detachments it did.

Each of these models then has a reasons function that saves all the possible reasons that it knows about. The model also includes a weight function that assigns a weight to each reason in each context and for each pairwise comparison. Of course, many of these weights may be zero. To use the system, a user may interact with it by specifying a set of reasons or grounds and a balancing operation. This will then output the options in that context, including which are permitted and which are not.

Thus, the detachment functions can be studied to see how they change from one version to the next and what update caused the change. Therefore, if one uses a detachment function and considers why the detachment function outputs a particular permission, then it is possible to look at which update this output exists in the detachment function. But also, if at any time the option had a different deontic status, and why it changed. This may lead back to an update that caused the detachment function to assign a specific deontic status to the option.

An update to the detachment function changes the weight function and the reasons function. An update may introduce new reasons and may change the weights associated with the reasons in contexts and comparisons (cogs). The sources of these updates can be various and can result from processes instantiated by managers. The update of weights may come from a loss function that specifies how weights should be changed to modify the detachment accordingly.

This is similar to a batch learning update in machine learning. To use this to track weight function updates of machine learning. In that case, the weight of a reason is changed; however, the data and learning process used may also be documented and referenced. This additional information may be understood as the source of the weight function update (vote).

Reducing dynamic scale to a single comparison

Considering a circumstance where there are three options o_1, o_2, and o_3. For the dynamic scale, we find multiple comparisons: o_1 vs o_2, o_1 vs o_3, o_2 vs o_1, o_2 vs o_3, o_3 vs o_1 and o_3 vs o_2. This is a total of 6 comparisons, but if we account for symmetric cases where a symmetric weight function can be assumed, there are only 3 comparisons. When a fourth option is added, we have more comparisons to make: o_1 vs o_4, o_2 vs o_4, o_3 vs o_4, o_4 vs o_1, o_4 vs o_2, and o_4 vs o_3. This adds another 6 comparisons, for a total of 12, and 3 + 3 = 6 in the symmetrical weight-function case.

If it were possible to determine the permissibility of each option with a single comparison, that would reduce the number of comparisons. In the case of 3 options with symmetrical weights, it does not reduce it, since that is already 3 options. In that case, if the weight function is not symmetrical, the comparisons are halved from 6 to 3 comparisons. In the 4-option case, when the weight function is symmetric, the number of comparisons is reduced from 6 to 4; in the unsymmetrical case, from 12 to 4.

To do this reduction, we may consider comparing an option with its negation or complement. So instead of comparing o_1 to each other option, we only compare o_1 with $\neg o_1$ or, if you will, $\bar{o_1}$. However, how should that be done, which cogs count toward o_1 and which count toward $\bar{o_1}$?

References

Alcaraz, B., Knoks, A., & Streit, D. (2024). Estimating weights of reasons using metaheuristics: A hybrid approach to machine ethics. In S. D. e. al (Ed.), *Proceedings of the Seventh AAAI/ACM Conference on AI, Ethics, and Society (AIES-2024)* (pp. 27–38). ACM Press.

Amgoud, L., Doder, D., & Vesic, S. (2022). Evaluation of argument strength in attack graphs: Foundations and semantics. *Artificial Intelligence*, *302*, 1–61.

Amgoud, L., Ben-Naim, J., Doder, D., & Vesic, S. (2017). Acceptability semantics for weighted argumentation frameworks. *Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI 2017)*.

Amgoud, L., & Vesic, S. (2014). Rich preference-based argumentation frameworks. *International Journal of Approximate Reasoning*, *55*(2), 585–606.

and European Data Protection Supervisor. (2025). *Ai act regulation (eu) 2024/1689 – regulation (eu) 2024/1689 of the european parliament and of the council of 13 june 2024 laying down harmonised rules on artificial intelligence and amending regulations (ec) no 300/2008, (eu) no 167/2013, (eu) no 168/2013, (eu) 2018/858, (eu) 2018/1139 and (eu) 2019/2144 and directives 2014/90/eu, (eu) 2016/797 and (eu) 2020/1828 (artificial intelligence act) (text with eea relevance)*. Publications Office of the European Union. https://doi.org/doi/10.2804/4225375

Baroni, P., Gabbay, D., Giacomin, M., & Van der Torre, L. (2018). *Handbook of Formal Argumentation*. London, England: College Publications.

Baroni, P., & Giacomin, M. (2007). On principle-based evaluation of extension-based argumentation semantics. *Artificial Intelligence*, *171*(10-15), 675–700.

Bratman, M. (1987). Intention, plans, and practical reason.

Bronstein, M. M., Bruna, J., Cohen, T., & Velickovic, P. (2021). Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. *CoRR*, *abs/2104.13478*. https://arxiv.org/abs/2104.13478

Buterin, V., et al. (2013). Ethereum white paper. *GitHub repository*, *1*(22-23), 5–7.

Chisholm, R. M. (1963). Contrary-to-duty imperatives and deontic logic. *Analysis*, *24*(2), 33–36.

Chopra, A., Van der Torre, L., Verhagen, H., & Villata, S. (2018). *Handbook of Normative Multiagent Systems*. London, England: College Publications.

Dancy, J. (2009). Reasons and rationality. *Spheres of reason*, 93–112.

de Wit, V. (2024). Trinominal/IHiBO_2.0: Balancin Normative Reasons. https://github.com/Trinominal/IHiBO_2.0

de Wit, V., Doder, D., & Meyer, J. (2021). *A Probabilistic Deontic Logic* (Vol. 12897 LNAI). https://doi.org/10.1007/978-3-030-86772-0{_}44

de Wit, V. (2023). Rule rather than exception: Defeasible probabilistic dyadic deontic logic. *DEON*.

de Wit, V., Doder, D., & Meyer, J.-J. (2023). Probabilistic Deontic Logics for Reasoning about Uncertain Norms. *Journal of Applied Logics-IfColog Journal of Logics and their Applications*, *10*(2), 193–220.

de Wit, V., Knoks, A., & Van der Torre, L. (2025a). Dual scale detachment. *Logics in Artificial Intelligence: 19th European Conference, JELIA 2025, Kutaisi, Georgia, September 1–4, 2025, Proceedings, Part I*, 205–219. https://doi.org/10.1007/978-3-032-04587-4_13

de Wit, V., Knoks, A., & Van der Torre, L. (2025b). Toward a principle-based analysis of dual-scale detachment. *Deontic Logic and Normative Systems-17th International Conference*.

de Wit, V., Yu, L., Najjar, A., & Markovich, R. (2025). Balancing normative reasons for the intelligent human-input-based blockchain oracle (ihibo). In Y. Mualla, L. Yu, D. Liga, I. Tchappi, & R. Markovich (Eds.), *Advances in explainability, agents, and large language models* (pp. 33–47). Springer Nature Switzerland.

Dung, P. M. (1995). On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games [Publisher: Elsevier]. *Artificial intelligence*, *77*(2), 321–357.

Francesconi, E. (2022). The winter, the summer and the summer dream of artificial intelligence in law. *Artificial Intelligence and Law*, *30*(2), 147–161. https://doi.org/10.1007/s10506-022-09309-8

Gabbay, D., Horty, J., Parent, X., Van der Meyden, R., & Van der Torre, L. (2013). Handbook of deontic logic and normative systems
Publisher: College Publication.

Gabbay, D., Horty, J., Parent, X., van der Meyden, R., & van der Torre, L. (Eds.). (2021). *Handbook of deontic logic and normative system, volume 2*. College Publications.

Gabbay, D. M., Guenthner, F., et al. (2002). Handbook of philosophical logic.

Grossi, D., & Pigozzi, G. (2022). *Judgment aggregation: A primer*. Springer Nature.

Hintikka, J. (2011). What is the axiomatic method? *Synthese*, *183*(1), 69–85.

Horty, J. (2007). Reasons as Defaults [Publisher: OUP USA]. *Philosophers' Imprint*, *7*, 1–28.

Horty, J. (2012). *Reasons as Defaults*. Oxford University Press.

Jørgensen, J. (1937). Imperatives and logic. *Erkenntnis*, *7*, 288–296.

Kamp, H. (2013). Free choice permission. In *Meaning and the dynamics of interpretation* (pp. 169–184). Brill.

Kampik, T. (2022). Principle-based Non-Monotonic Reasoning-From Humans to Machines.

Kirilenko, A., Kyle, A. S., Samadi, M., & Tuzun, T. (2017). The flash crash: High-frequency trading in an electronic market. *The Journal of Finance*, *72*(3), 967–998.

Knoks, A., Shao, M., Van der Torre, L., de Wit, V., & Yu, L. (2024). A Principle-based Analysis for Numerical Balancing. *Logics for New-Generation Artificial Intelligence (LNGAI2024)*.

Knoks, A., & Van der Torre, L. (2023). Reason-based detachment. In B. Bentzen, B. Liao, D. Liga, R. Markovich, B. Wei, M. Xiong, & T. Xu (Eds.), *Joint Proceedings of the 3rd International Workshop on Logics for New-Generation Artificial Intelligence and the International Workshop on Logic, AI, and Law (LNGAI/LAIL2023, Hangzhou)* (pp. 49–65). College Publications.

List, C., Puppe, C., et al. (2009). Judgement aggregation: A survey.

Logic, F. (2006). Stanford encyclopedia of philosophy. *Bryant University*, *23*.

Maginn, J. L., Tuttle, D. L., McLeavey, D. W., & Pinto, J. E. (2007). *Managing investment portfolios: A dynamic process* (Vol. 3). John Wiley & Sons.

Makinson, D., & Torre, L. v. d. (2001). Constraints for input/output logics. *Journal of Philosophical Logic*, *30*, 155–85.

Makinson, D. (2005, January). *Bridges from classical to nonmonotonic logic*.

Makinson, D., & Van der Torre, L. (2000). Input/output logics [Publisher: Springer]. *Journal of philosophical logic*, *29*, 383–408.

Manolache, M. A., Manolache, S., & Tapus, N. (2022). Decision making using the blockchain proof of authority consensus [Publisher: Elsevier]. *Procedia Computer Science*, *199*, 580–588.

McCarthy, J. (1959). Programs with common sense.

McCarthy, J., Minsky, M. L., Rochester, N., & Shannon, C. E. (2006). A proposal for the dartmouth summer research project on artificial intelligence: August 31, 1955. *AI Mag.*, *27*(4), 12–14. https://doi.org/10.1609/aimag.v27i4.1904

Meyer, J.-j., Dignum, F., & Wieringa, R. (2001). The paradoxes of deontic logic revisited: A computer science perspective.

Modgil, S., & Prakken, H. (2014). The ASPIC + framework for structured argumentation: A tutorial [Publisher: IOS Press]. *Argument & Computation*, *5*(1), 31–62. https://doi.org/10.1080/19462166.2013.869766

Moor, J. (2006). The dartmouth college artificial intelligence conference: The next fifty years. *AI Magazine*, *27*, 87–91.

Nair, S. (2021). "Adding Up" Reasons: Lessons for Reductive and Nonreductive Approaches [Publisher: The University of Chicago Press Chicago, IL]. *Ethics*, *132*(1), 38–88.

Nakamoto, S., Bit, B., et al. (2007). Bitcoin: A peer-to-peer electronic cash system. *2008*.

Nash Jr, J. F. (1950). Equilibrium points in n-person games. *Proceedings of the national academy of sciences*, *36*(1), 48–49.

Nick, B. (2014). Superintelligence: Paths, dangers, strategies. *Strategies*.

Norvig, P. R., & Intelligence, S. A. (2002). A modern approach. *Prentice Hall Upper Saddle River, NJ, USA: Rani, M., Nayak, R., & Vyas, OP (2015). An ontology-based adaptive personalized* *e-learning system, assisted by software agents on cloud storage. Knowledge-Based Systems*, *90*, 33–48.

Pandžić, S., & Graff, J. (2024). A Logic of Weighted Reasons for Explainable Inference in AI. *World Conference on Explainable Artificial Intelligence*, 243–267.

Parent, X., & Torre, L. v. d. (2013). Input/output logic. In L. v. d. Torre, D. Gabbay, J. Horty, & R. v. d. Meyden (Eds.), *Handbook of Deontic Logic, vol. 1* (Vol. 1). College Publications.

Parent, X., & Van der Torre, L. (2018). *Introduction to deontic logic and normative systems*. College Publications, London, United Kingdom.

Paulin, J., Calinescu, A., & Wooldridge, M. (2019). Understanding flash crash contagion and systemic risk: A micro–macro agent-based approach. *Journal of Economic Dynamics and Control*, *100*, 200–229.

Ross, A. (1944). Imperatives and logic. *Philosophy of Science*, *11*(1), 30–46.

Schmidhuber, J. (2015). Deep Learning in Neural Networks: An Overview [arXiv:1404.7828 [cs]]. *Neural Networks*, *61*, 85–117. https://doi.org/10.1016/j.neunet.2014.09.003

Schroeder, M. (2021a). *Reasons First*. Oxford University Press.

Schroeder, M. (2021b). The fundamental reason for reasons fundamentalism. *Philosophical Studies: An International Journal for Philosophy in the Analytic Tradition*, 3107–3127.

Schroeder, M. (2021c). *Reasons first*. Oxford University Press.

Sher, I. (2019). Comparative value and the weight of reasons [Publisher: Cambridge University Press]. *Economics & Philosophy*, *35*(1), 103–158.

Sobek, T., & Montag, J. (2018). Proportionality test. In *Encyclopedia of law and economics* (pp. 1–5). Springer.

Spielthenner, G. (2017). The principle-based method of practical ethics. *Health Care Analysis*, *25*(3), 275–289.

Streit, D. (2024). *From Metaethics to Machine Decisions: Formal Models of Normative Reasons and their Application in Philosophy and Machine Ethics* [PhD Thesis]. University of Luxembourg.

Streit, D., de Wit, V., & Knoks, A. (2023). Reasons in Weighted Argumentation Graphs. *International Workshop on Logic, Rationality and Interaction*, 251–259.

Tsakyrakis, S. (2009, July). Proportionality: An Assault on Human Rights? https://doi.org/10.1093/icon/mop011

Tucker, C. (2025). *The Weight of Reasons: A Framework for Ethics*. Oxford University Press.

Van der Torre, L., & Vesic, S. (2018). The principle-based approach to abstract argumentation semantics. In P. Baroni, D. Gabbay, M. Giacomin, & L. Van der Torre (Eds.), *Handbook of Formal Argumentation*. College Publications.

Vassiliades, A., Bassiliades, N., & Patkos, T. (2021). Argumentation and explainable artificial intelligence: A survey [Publisher: Cambridge University Press]. *The Knowledge Engineering Review*, *36*, e5.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, *30*.

Von Wright, G. H. (1951). Deontic logic [Publisher: JSTOR]. *Mind*, *60*(237), 1–15.

Yu, L. (2023). *Distributed argumentation technology: Advancing risk analysis and regulatory compliance of distributed ledger technology (dlt) for transaction and management of securities* [Doctoral dissertation, Unilu - University of Luxembourg [Science Technology and Medicine], Belval, Luxembourg].

Yu, L., & Gabbay, D. (2022). Case-Based Reasoning via Comparing the Strength Order of Features. *International Workshop on Explainable, Transparent Autonomous Agents and Multi-Agent Systems*, 143–151.

Yu, L., Zichichi, M., Markovich, R., & Najjar, A. (2022). Enhancing Trust in Trust Services: Towards an Intelligent Human-input-based Blockchain Oracle (IHiBO). *Proceedings of the 55th Annual Hawaii International Conference on System Sciences*. https://doi.org/10.5220/0010945300003116

Yu, L., Zichichi, M., Najjar, A., & Markovich, R. (2021). Argumentation in Trust Services within a Blockchain Environment [Place: Schifflange, Luxembourg]. *Proceedings of the 33rd Benelux Conference on Artificial Intelligence and the 30th Belgian Dutch Conference on Machine Learning (BNAIC/BENELEARN 2021)*.

Yu, L., Zichichi, M., Réka, M., Amro, N., et al. (2022). Intelligent Human-input-based Blockchain Oracle (IHiBO). *Proceedings of the 14th International Conference on Agents and Artificial Intelligence (ICAART 2022)*, 1–12.

Zichichi, M. (2021). Miker83z/IHiBO: Intelligent Human-input-based Blockchain Oracle. https://github.com/miker83z/IHiBO

Appendix A

Proofs

For the following proofs to reduce repetition, we introduce the following aliases. Given inputs: R, o_1, o_2, RR and c; for a detachment function D.

- $sjo_1 = \Sigma_{(g_1,o_1)\in RR} jw(R, o_1, o_2, c, (g_1, o_1))$
- $sro_2 = \Sigma_{(g_2,o_2)\in RR} rw(R, o_1, o_2, c, (g_2, o_2))$
- $sjo_2 = \Sigma_{(g_2,o_2)\in RR} jw(R, o_1, o_2, c, (g_2, o_2))$
- $sro_1 = \Sigma_{(g_1,o_1)\in RR} rw(R, o_1, o_2, c, (g_1, o_1))$

For Maximum detachment we use and m instead of an s for the respective maximums.

- $mjo_1 = \max(\{jw(R, o_1, o_2, c, (g_1, o_1)) \mid (g_1, o_1) \in RR\})$
- $mro_2 = \max(\{rw(R, o_1, o_2, c, (g_2, o_2)) \mid (g_2, o_2) \in RR\})$
- $mjo_2 = \max(\{jw(R, o_1, o_2, c, (g_2, o_2)) \mid (g_2, o_2) \in RR\})$
- $mro_1 = \max(\{rw(R, o_1, o_2, c, (g_1, o_1)) \mid (g_1, o_1) \in RR\})$

A.1 Dual Scale Detachment

Proposition 4. *Dual scale detachment satisfies groundedness.*

Proof. We use a direct proof by cases to prove that dual scale detachment satisfies groundedness. Consider some dual scale detachment function D_w. Take $o_1, o_2 \in O$ and $c \in C$ and $RR \subseteq R(o_1, o_2)$ such that $D_w(R, o_1, o_2, RR, c) = (v_1, v_2)$.

If $v_1 = +$ or $v_2 = -$ we prove that there exists a $(g, o_1) \in RR$. Then $sjo_1 - sro_2 > 0$ and or $C - D < 0$. In order for $sjo_1 - sro_2 > 0$ we need $sjo_1 > sro_2$ and so $sjo_1 > 0$ this is only possible if there is a reason $r = (g, o_1) \in RR$ with $jw(r) > 0$. Similarly for $sjo_2 - sro_1 < 0$ to be true $sjo_2 < sro_1$ which means $sro_1 > 0$ which is only possible if there is a reason $r' = (g', o_1) \in RR$ with $rw(r') > 0$.

If $v_1 = -$ or $v_2 = +$ we prove that there exist a $(g, o_2) \in RR$. Then $sjo_1 - sro_2 < 0$ and or $sjo_2 - sro_1 > 0$. In order for $sjo_1 - sro_2 < 0$ we need $sjo_1 < sro_2$ and so $sro_2 > 0$ this is only possible if there is a reason $r = (g, o_2) \in RR$ with $rw(r) > 0$. Similarly for $sjo_2 - sro_1 > 0$ to be true $sjo_2 > sro_1$ which means $sjo_2 > 0$ which is only possible if there is a reason $r' = (g', o_2) \in RR$ with $jw(r') > 0$.

We conclude that dual scale satisfies groundedness. □

Proposition 5. *Dual scale detachment fails anonymity.*

Proof. We use a counterexample to prove that there exists a weight function that fails anonymity. Let $r_1 = (g_1, o_1)$ and $r_2 = (g_2, o_2)$, and let $(v_1, v_2) = D_w(R, o_1, o_2, \{r_1, r_2\}, c)$. And define π such that $r_1' = (\pi(g_1), o_1)$ and $r_2' = (\pi(g_2), o_2)$, such that $(v_1', v_2') = D_w(R, o_1, o_2, \{r_1', r_2'\}, c)$. We show that there exists w such that $v_1 \neq v_1' \vee v_2 \neq v_2'$.

Since r_1 is a different reason than r_1' it need not be that $w(r_1) = w(r_1')$, and similarly $w(r_2) = w(r_2')$ does not need to hold.

Let $w(r_1) = (1, 3)$, $w(r_2) = (2, 1)$, $w(r_1') = (2, 1)$, and $w(r_2') = (3, 1)$ then $v_1 = 0$ since $jw(r_1) = 1 = rw(r_2)$ and $v_2 = -$ since $jw(r_2) = 2 < 3 = rw(r_1)$. Moreover, $v_1' = +$ since $jw(r_1') = 2 > 1 = rw(r_2')$ and $v_2' = +$ since $jw(r_2') = 3 > 1 = rw(r_1')$. So we have $v_1 = 0 \neq + = v_1'$ and $v_2 = - \neq + = v_2'$. This completes the counter example.

We conclude that dual scale detachment fails anonymity. □

Proposition 6. *Dual scale detachment fails unanimity.*

Proof. We use a counterexample to prove that there exists a weight function that fails unanimity. We assume RR to be non-empty, since for empty RR there is nothing to prove. Let $r_1 = (g_1, o_1)$ and let $(v_1, v_2) = D_w(R, o_1, o_2, RR = \{r_1\}, c)$, such that $w(r_1) = (x, y)$.

We cannot have $x = 0$ and $y = 0$ otherwise $r_1 \notin RR$. Then if $x = 0$ and $y \neq 0$ we have $D_w(R, o_1, o_2, \{r_1\}, c) = (0, -)$ since $0 > -y$.And if $x \neq 0$ and $y = 0$ we have $D_w(R, o_1, o_2, \{r_1\}, c) = (+, 0)$ since $x > 0$. Both of these outputs are not $(+, -)$. This shows that there exist weight functions such that dual scale detachment fails unanimity.

We conclude that dual-scale detachment fails unanimity. □

Proposition 7. *Dual scale detachment satisfies polarity monotony.*

Proof. We use a direct proof to prove that dual scale detachment satisfies polarity monotony. Consider some dual scale detachment function D_w. Take $o_1, o_2 \in O$ and $c \in C$ and $RR \subseteq R(o_1, o_2)$ such that $D_w(R, o_1, o_2, RR, c) = (v_1, v_2)$. And take $r = (g, o_1) \in R(o_1, o_2)\backslash RR$ such that $D_w(R, o_1, o_2, RR \cup \{r\}, c) = (v'_1, v'_2)$.

We show that $v'_1 \geq v_1$; for v_1 we compare sjo_1 to sro_2 and for v'_1 we compare $sjo_1 + jw(r)$ and sro_2, observe that $sjo_1 \leq sjo_1 + jw(r)$ from this follows that $v'_1 \geq v_1$. Similarly we can show that $v'_2 \geq v_2$ by observing that for v_2 we compare sjo_2 to sro_1 and for v'_2 we compare sjo_2 with $sro_1 + rw(r)$, observe that $sro_1 \leq sro_1 + rw(r)$ from this follows that $v'_2 \leq v_2$.

When we add a reason $r' = (g, o_2)$ instead of r we find that $v'_1 \leq v_1$ since $sro_2 + rw(r') \geq sro_2$ and we find $v'_2 \geq v_2$ since $sjo_2 \leq sjo_2 + jw(r')$.

This proves polarity monotony holds for dual scale detachment. We conclude that dual scale satisfies polarity monotony. □

Proposition 8. *Dual scale detachment fails union monotony.*

Proof. We use a counter example to prove that dual scale detachment fails union monotony. Consider some dual scale detachment function D_w. Take $o_1, o_2 \in O$ and $c \in C$ and $RR_1, RR_2 \subseteq R(o_1, o_2)$ such that $D_w(R, o_1, o_2, RR_1, c) = (v_1, v_2)$, $D_w(R, o_1, o_2, RR_2, c) = (v_1, v_2)$ and $D_w(R, o_1, o_2, RR_1 \cup RR_2, c) = (v'_1, v'_2)$. We show that it is possible for $v_1 \neq v'_1 \vee v_2 \neq v'_2$ to hold.

Let $r_1 = (g_1, o_1)$, $r_2 = (g_2, o_2)$, $r_3 = (g_3, o_2)$, with $RR_1 = \{(g_1, o_1), (g_2, o_2)\}$ and $RR_2 =$

$\{(g_1, o_1), (g_3, o_2)\}$ and the following weight function: $w(r_1) = (3, 3)$, $w(r_2) = (2, 2)$, $w(r_3) = (2, 2)$.

Then $v_1 = +$ since $jw(r_1) = 3 > 2 = rw(r_2) = rw(r_3)$, and $v_2 = -$ since $rw(r_1) = 3 > 2 = jw(r_2) = jw(r_3)$. Furthermore $v_1' = -$ since $jw(r_1) = 3 < 4 = rw(r_2) + rw(r_3)$ and $v_2' = +$ since $jw(r_2) + jw(r_3) = 4 > 3 = rw(r_1)$. Observe that $v_1 \neq v_1'$, and $v_2 \neq v_2'$.

We conclude that duals scale fails union monotony. □

Proposition 9. *Dual scale detachment satisfies sensitivity.*

Proof. We use a direct proof to prove that dual scale detachment satisfies sensitivity. Consider some dual scale detachment function D_w. Take $o_1, o_2 \in O$ and $c \in C$ and $RR \subseteq R(o_1, o_2)$ such that $D_w(R, o_1, o_2, RR, c) = (0, 0)$ and an $r \in R(o_1, o_2)$.

We show that for $(v_1, v_2) \in D_w(R, o_1, o_2, RR \cup \{r\}, c)$ that $v_1 \neq 0 \vee v_2 \neq 0$. Since $D_w(R, o_1, o_2, RR, c) = (0, 0)$ we know that $sjo_1 - sro_2 = 0 = sjo_2 - sro_1$. We also know that for $w(r) = (w_1, w_2)$ at least w_1 or w_2 is positive.

From this it follows that if r is a reason for o_1 then $sjo_1 + jw(r) \geq sro_2$ and $sjo_2 \leq sro_1 + rw(r)$ therefore $v_1 = + \vee v_2 = -$ is true. Similarly, if r is a reason for o_2 then $sjo_1 \leq sro_2 + rw(r)$ and $sjo_2 + jw(r) \geq sro_1$ and therefore $v_1 = - \vee v_2 = +$ is true. This proves that sensitivity holds for dual scale detachment.

We conclude that dual scale satisfies sensitivity. □

Proposition 10. *Dual scale detachment satisfies input symmetry.*

Proof. We use a direct proof to prove that dual scale detachment satisfies input symmetry. Consider some dual scale detachment function D_w. Take $o_1, o_2 \in O$ and $c \in C$ and $RR \subseteq R(o_1, o_2)$ such that $D_w(R, o_1, o_2, RR, c) = (v_1, v_2)$, and $D_w(R, o_2, o_1, RR, c) = (v_1', v_2')$.

We show that $(v_2, v_1) \in D_w(R, o_2, o_1, RR, c)$ and so that $v_1 = v_2'$ and that $v_2 = v_1'$. We have the following:

- $sjo_1 = \Sigma_{(g_1, o_1) \in RR} jw(R, o_1, o_2, c, (g_1, o_1)) = \Sigma_{(g_1, o_1) \in RR} jw((g_1, o_1))$
- $sro_2 = \Sigma_{(g_2, o_2) \in RR} rw(R, o_1, o_2, c, (g_2, o_2)) = \Sigma_{(g_2, o_2) \in RR} rw((g_2, o_2))$

- $sjo_2 = \Sigma_{(g_2,o_2)\in RR} jw(R, o_1, o_2, c, (g_2, o_2)) = \Sigma_{(g_2,o_2)\in RR} jw((g_2, o_2))$
- $sro_1 = \Sigma_{(g_1,o_1)\in RR} rw(R, o_1, o_2, c, (g_1, o_1)) = \Sigma_{(g_1,o_1)\in RR} rw((g_1, o_1))$

But also:

- $sjo'_1 = \Sigma_{(g_2,o_2)\in RR} jw(R, o_2, o_1, c, (g_2, o_2)) = \Sigma_{(g_2,o_2)\in RR} jw((g_2, o_2))$
- $sro'_2 = \Sigma_{(g_1,o_1)\in RR} rw(R, o_2, o_1, c, (g_1, o_1)) = \Sigma_{(g_1,o_1)\in RR} rw((g_1, o_1))$
- $sjo'_2 = \Sigma_{(g_1,o_1)\in RR} jw(R, o_2, o_1, c, (g_1, o_1)) = \Sigma_{(g_1,o_1)\in RR} jw((g_1, o_1))$
- $sro'_1 = \Sigma_{(g_2,o_2)\in RR} rw(R, o_2, o_1, c, (g_2, o_2)) = \Sigma_{(g_2,o_2)\in RR} rw((g_2, o_2))$

Observe that $sjo_1 = sjo'_2$, $sro_2 = sro'_1$, $sjo_2 = sjo'_1$, and $sro_1 = sro'_2$. Given the definition of dual scale we have $v_1 = v'_2$ exactly then when $sjo_1 = sjo'_2$ together with $sro_2 = sro'_1$. And we have $v_2 = v'_1$ exactly when $sjo_2 = sjo'_1$ and $sro_1 = sro'_2$. This proves that input symmetry holds for dual scale detachment.

We conclude that dual scale satisfies input symmetry.

□

Proposition 11. *Dual scale detachment fails no dilemmas.*

Proof. We use a counterexample to prove that there exists a weight function that fails no dilemmas.

We use proof by counterexample; we present a weighing function such that dual scale detachment fails No Dilemmas. Let $r_1 = (g_1, o_1)$ and $r_2 = (g_2, o_2)$. And let $(v_1, v_2) = D_w(R, o_1, o_2, \{r_1, r_2\}, c)$ such that $w(R, o_1, o_2, c, r_1) = (1, 3)$, $w(R, o_1, o_2, c, r_2) = (1, 3)$. Then since $jw(r_1) - rw(r_2) = 1 - 3 = -2$ we have $v_1 = -$ and since $jw(r_2) - rw(r_1) = 1 - 3 = -2$ we have $v_2 = -$, then $D_w(\{r_1, r_2\}) = (-, -)$, which completes the counterexample.

We conclude that dual scale detachment fails no dilemma. □

Proposition 12. *Dual scale detachment fails one permission.*

Proof. We use a counterexample to prove that there exists a weight function that fails one permission. We complete the counterexample if we can define a weight function such that there exists a $(v_1, v_2) = D_w(R, o_1, o_2, RR, c)$, where $(v_1, v_2) = (+, 0)$.

Let $r_1 = (g_1, o_1)$ and $r_2 = (g_2, o_2)$, also let $(v_1, v_2) = D_w(R, o_1, o_2, \{r_1, r_2\}, c)$, such that $w(r_1) = (3, 1)$ and $w(r_1) = (1, 1)$. Then $v_1 = +$ since $sjo_1 = 3 > 1 = sro_2$ and $v_2 = 0$ since $sjo_2 = 1 = sro_1$, therefore $D_w(\{r_1, r_2\}) = (+, 0)$, which completes the counterexample.

We conclude that dual scale detachment fails one permission. □

A.2 Single Scale Detachment

Proposition 13. *Single scale detachment satisfies groundedness.*

Proof. We use a direct proof to prove that single scale detachment satisfies groundedness. Consider some single scale detachment function D_w. Take $o_1, o_2 \in O$ and $c \in C$ and $RR \subseteq R(o_1, o_2)$ such that $D_w(R, o_1, o_2, RR, c) = (v_1, v_2)$.

If $v_1 = +$ or $v_2 = -$ we prove that there exists a $(g, o_1) \in RR$. Then $sjo_1 - sro_2 > 0$ and or $sjo_2 - sro_1 < 0$. In order for $sjo_1 - sro_2 > 0$ we need $sjo_1 > sro_2$ and so $sjo_1 > 0$ this is only possible if there is a reason $r = (g, o_1) \in RR$ with $jw(r) > 0$. Similarly for $sjo_2 - sro_1 < 0$ to be true $sjo_2 < sro_1$ which means $sro_1 > 0$ which is only possible if there is a reason $r' = (g', o_1) \in RR$ with $rw(r') > 0$.

If $v_1 = -$ or $v_2 = +$ we prove that there exist a $(g, o_2) \in RR$. Then $sjo_1 - sro_2 < 0$ and or $sjo_2 - sro_1 > 0$. In order for $sjo_1 - sro_2 < 0$ we need $sjo_1 < sro_2$ and so $sro_2 > 0$ this is only possible if there is a reason $r = (g, o_2) \in RR$ with $rw(r) > 0$. Similarly, for $sjo_2 - sro_1 > 0$ to be true, $sjo_2 > sro_1$, which means $sjo_2 > 0$, which is only possible if there is a reason $r' = (g', o_2) \in RR$ with $jw(r') > 0$.

We conclude that single scale satisfies groundedness. □

Proposition 14. *Single scale detachment fails anonymity.*

Proof. We use a counterexample to prove that there exists a weight function that fails anonymity. Let $r_1 = (g_1, o_1)$ and $r_2 = (g_2, o_2)$, and let $(v_1, v_2) = D_w(R, o_1, o_2, \{r_1, r_2\}, c)$. And define π such that $r'_1 = (\pi(g_1), o_1)$ and $r'_2 = (\pi(g_2), o_2)$, such that $(v'_1, v'_2) = D_w(R, o_1, o_2, \{r'_1, r'_2\}, c)$. We show that there exists w such that $v_1 \neq v'_1 \vee v_2 \neq v'_2$.

Since r_1 is a different reason than r_1' it need not be that $w(r_1) = w(r_1')$, and similarly $w(r_2) = w(r_2')$ does not need to hold.

Let $w(r_1) = (3,3)$, $w(r_2) = (2,2)$, $w(r_1') = (2,2)$, and $w(r_2') = (3,3)$ then $v_1 = +$ since $jw(r_1) = 3 > 2 = rw(r_2)$ and $v_2 = -$ since $jw(r_2) = 2 < 3 = rw(r_1)$. Moreover, $v_1' = -$ since $jw(r_1') = 2 < 3 = rw(r_2')$ and $v_2' = +$ since $jw(r_2') = 3 > 2 = rw(r_1')$. So we have $v_1 = + \neq - = v_1'$ and $v_2 = - \neq + = v_2'$. This completes the counter example.

We conclude that single scale detachment fails anonymity. □

Proposition 15. *Single scale detachment satisfies unanimity.*

Proof. We use a direct proof to prove that single scale detachment satisfies unanimity. We assume RR to be non-empty, since for empty RR there is nothing to prove. Consider some single scale detachment function D_w. Take $o_1, o_2 \in O$ and $c \in C$ and $RR \subseteq R(o_1, o_2)$ such that $D_w(R, o_1, o_2, RR, c) = (v_1, v_2)$.

If $RR = \{(g, o_1) \in RR : g \in G' \subseteq G\}$ then we know that $sjo_1 > sro_2$ and $sjo_2 < sro_1$ due to single proportion assumption the weights justifying and requiring weight of a reason are the same. Therefore, since each reason favors o_1 we have $sjo_1 > 0$, $sro_2 = 0$, $sjo_2 = 0$ and $sro_1 > 0$. So $v_1 = +$ and $v_2 = -$, then $D_w(R, o_1, o_2, RR, c) = (+, -)$ similarly if each reason favors o_2 we find $D_w(R, o_1, o_2, RR, c) = (-, +)$.

We conclude that single scale detachment satisfies unanimity. □

Proposition 16. *Single scale detachment satisfies polarity monotony.*

Proof. We use a direct proof to prove that single scale detachment satisfies polarity monotony. Consider some single scale detachment function D_w. Take $o_1, o_2 \in O$ and $c \in C$ and $RR \subseteq R(o_1, o_2)$ such that $D_w(R, o_1, o_2, RR, c) = (v_1, v_2)$. And take $r = (g, o_1) \in R(o_1, o_2) \backslash RR$ such that $D_w(R, o_1, o_2, RR \cup \{r\}, c) = (v_1', v_2')$.

We show that $v_1' \geq v_1$; for v_1 we compare sjo_1 to sro_2 and for v_1' we compare $sjo_1 + jw(r)$ and sro_2, observe that $sjo_1 \leq sjo_1 + jw(r)$ from this follows that $v_1' \geq v_1$. Similarly we can show that $v_2' \geq v_2$ by observing that for v_2 we compare sjo_2 to sro_1 and for v_2' we compare sjo_2 with $sro_1 + rw(r)$, observe that $sro_1 \leq sro_1 + rw(r)$ from this follows that $v_2' \leq v_2$.

When we add a reason $r' = (g, o_2)$ instead of r we find that $v'_1 \leq v_1$ since $sro_2 + rw(r') \geq sro_2$ and we find $v'_2 \geq v_2$ since $sjo_2 \leq sjo_2 + jw(r')$.

This proves polarity monotony holds for single scale detachment. We conclude that single scale satisfies polarity monotony. □

Proposition 17. *Single scale detachment fails union monotony.*

Proof. We use a counterexample to prove that single scale detachment fails union monotony. Consider some single scale detachment function D_w. Take $o_1, o_2 \in O$ and $c \in C$ and $RR_1, RR_2 \subseteq R(o_1, o_2)$ such that $D_w(R, o_1, o_2, RR_1, c) = (v_1, v_2)$, $D_w(R, o_1, o_2, RR_2, c) = (v_1, v_2)$ and $D_w(R, o_1, o_2, RR_1 \cup RR_2, c) = (v'_1, v'_2)$. We show that it is possible for $v_1 \neq v'_1$ and $v_2 \neq v'_2$ to hold.

Let $r_1 = (g_1, o_1)$, $r_2 = (g_2, o_2)$, $r_3 = (g_3, o_2)$, with $RR_1 = \{(g_1, o_1), (g_2, o_2)\}$ and $RR_2 = \{(g_1, o_1), (g_3, o_2)\}$ and the following weight function: $w(r_1) = (3, 3)$, $w(r_2) = (2, 2)$, $w(r_3) = (2, 2)$.

Then $v_1 = +$ since $jw(r_1) = 3 > 2 = rw(r_2) = rw(r_3)$, and $v_2 = -$ since $rw(r_1) = 3 > 2 = jw(r_2) = jw(r_3)$. Furthermore $v'_1 = -$ since $jw(r_1) = 3 < 4 = rw(r_2) + rw(r_3)$ and $v'_2 = +$ since $jw(r_2) + jw(r_3) = 4 > 3 = rw(r_1)$. Observe that $v_1 \neq v'_1$, and $v_2 \neq v'_2$.

We conclude that single scale fails union monotony. □

Proposition 18. *Single scale detachment satisfies sensitivity.*

Proof. We use a direct proof to prove that single scale detachment satisfies sensitivity. Consider some single scale detachment function D_w. Take $o_1, o_2 \in O$ and $c \in C$ and $RR \subseteq R(o_1, o_2)$ such that $D_w(R, o_1, o_2, RR, c) = (0, 0)$ and an $r \in R(o_1, o_2)$.

We show that for $(v_1, v_2) \in D_w(R, o_1, o_2, RR \cup \{r\}, c)$ that $v_1 \neq 0 \vee v_2 \neq 0$. Since $D_w(R, o_1, o_2, RR, c) = (0, 0)$ we know that $sjo_1 - sro_2 = 0 = sjo_2 - sro_1$. We also know that for $w(r) = (w_1, w_2)$ the justifying and requiring weight are equal and positive $w_1 = w_2 > 0$.

From this it follows that if r is a reason for o_1 then $sjo_1 + jw(r) \geq sro_2$ and $sjo_2 \leq sro_1 + rw(r)$ therefore $v_1 = + \wedge v_2 = -$ is true. Similarly, if r is a reason for o_2 then $sjo_1 \leq sro_2 + rw(r)$ and $sjo_2 + jw(r) \geq sro_1$ and therefore $v_1 = - \wedge v_2 = +$ is true. This proves that sensitivity holds for single scale detachment.

We conclude that single scale detachment satisfies sensitivity. □

Proposition 19. *Single scale detachment satisfies input symmetry.*

Proof. We use a direct proof to prove that single scale detachment satisfies input symmetry. Consider some single scale detachment function D_w. Take $o_1, o_2 \in O$ and $c \in C$ and $RR \subseteq R(o_1, o_2)$ such that $D_w(R, o_1, o_2, RR, c) = (v_1, v_2)$ and $D_w(R, o_2, o_1, RR, c) = (v_1', v_2')$. We show that $(v_2, v_1) \in D_w(R, o_2, o_1, RR, c)$ and so that $v_1 = v_2'$ and that $v_2 = v_1'$.

We have the following:

- $sjo_1 = \Sigma_{(g_1,o_1)\in RR} jw(R, o_1, o_2, c, (g_1, o_1)) = \Sigma_{(g_1,o_1)\in RR} jw((g_1, o_1))$
- $sro_2 = \Sigma_{(g_2,o_2)\in RR} rw(R, o_1, o_2, c, (g_2, o_2)) = \Sigma_{(g_2,o_2)\in RR} rw((g_2, o_2))$
- $sjo_2 = \Sigma_{(g_2,o_2)\in RR} jw(R, o_1, o_2, c, (g_2, o_2)) = \Sigma_{(g_2,o_2)\in RR} jw((g_2, o_2))$
- $sro_1 = \Sigma_{(g_1,o_1)\in RR} rw(R, o_1, o_2, c, (g_1, o_1)) = \Sigma_{(g_1,o_1)\in RR} rw((g_1, o_1))$

But also:

- $sjo_1' = \Sigma_{(g_2,o_2)\in RR} jw(R, o_2, o_1, c, (g_2, o_2)) = \Sigma_{(g_2,o_2)\in RR} jw((g_2, o_2))$
- $sro_2' = \Sigma_{(g_1,o_1)\in RR} rw(R, o_2, o_1, c, (g_1, o_1)) = \Sigma_{(g_1,o_1)\in RR} rw((g_1, o_1))$
- $sjo_2' = \Sigma_{(g_1,o_1)\in RR} jw(R, o_2, o_1, c, (g_1, o_1)) = \Sigma_{(g_1,o_1)\in RR} jw((g_1, o_1))$
- $sro_1' = \Sigma_{(g_2,o_2)\in RR} rw(R, o_2, o_1, c, (g_2, o_2)) = \Sigma_{(g_2,o_2)\in RR} rw((g_2, o_2))$

Observe that $sjo_1 = sjo_2'$, $sro_2 = sro_1'$, $sjo_2 = sjo_1'$, and $sro_1 = sro_2'$. Given the definition of single scale we have $v_1 = v_2'$ exactly then when $sjo_1 = sjo_2'$ together with $sro_2 = sro_1'$. And we have $v_2 = v_1'$ exactly when $sjo_2 = sjo_1'$ and $sro_1 = sro_2'$. This proves that input symmetry holds for single scale detachment.

We conclude that single scale satisfies input symmetry. □

Proposition 20. *Single scale detachment satisfies no dilemmas.*

Proof. Suppose, toward a contradiction, that single scale detachment does not satisfy No Dilemmas. This means that there is some single detachment function D such that, for some o_1, o_2, $RR \subseteq R(o_1, o_2)$, and c, we have $D(R, o_1, o_2, RR, c) = (-, -)$. By Definition 18 (Single Scale), we know that there is some weighting function w such that $D_w(R, o_1, o_2, RR, c) = (-, -)$ and $jw = rw$. By Definition 4 (Additive Dual Scale Weighing), we know, first, that $\Sigma_{(g_1,o_1)\in RR} jw((g_1, o_1)) - \Sigma_{(g_2,o_2)\in RR} rw((g_2, o_2)) < 0$, as well as, second, that $\Sigma_{(g_2,o_2)\in RR} jw((g_2, o_2)) - \Sigma_{(g_1,o_1)\in RR} rw((g_1, o_1)) < 0$. But $jw = rw$ tells us that $\Sigma_{(g_1,o_1)\in RR} jw((g_1, o_1)) = \Sigma_{(g_1,o_1)\in RR} rw((g_1, o_1)) = x$. Similarly, it tells us that $\Sigma_{(g_2,o_2)\in RR} jw((g_2, o_2)) = \Sigma_{(g_2,o_2)\in RR} rw((g_2, o_2)) = y$. So, we have $y < x$ and $x > y$, which is a contradiction. □

Proposition 21. *Single scale detachment satisfies one permission.*

Proof. We use a direct proof to prove that single scale detachment satisfies one permission. Consider some single scale detachment function D_w. Take $o_1, o_2 \in O$ and $c \in C$ and $RR \subseteq R(o_1, o_2)$ such that $D_w(R, o_1, o_2, RR, c) = (v_1, v_2)$. For D_w to be a single scale detachment function, we require that $\forall r \in R.jw(r) = rw(r)$.

The principle is satisfied if three output pairs are **not** possible: $(+, +)$, $(+, 0)$, and $(0, +)$. These occur when $sjo_1 \geq sro_2$ together with $sjo_2 \geq sro_1$.

$$x = sjo_1 - sro_2 = \Sigma_{(g_1,o_1)\in RR} jw((g_1, o_1)) - \Sigma_{(g_2,o_2)\in RR} rw((g_2, o_2))$$

$$y = sjo_2 - sro_1 = \Sigma_{(g_2,o_2)\in RR} jw((g_2, o_2)) - \Sigma_{(g_1,o_1)\in RR} rw((g_1, o_1))$$

We prove that $x > 0$ if and only if $y < 0$. Due to single proportion, we can write:

$$x = sjo_1 - sjo_2 = \Sigma_{(g_1,o_1)\in RR} jw((g_1, o_1)) - \Sigma_{(g_2,o_2)\in RR} jw((g_2, o_2))$$

$$y = sjo_2 - sjo_1 = \Sigma_{(g_2,o_2)\in RR} jw((g_2, o_2)) - \Sigma_{(g_1,o_1)\in RR} jw((g_1, o_1))$$

From this we can see that if $x > 0$ then $sjo_1 > sjo_2$ therefore $sjo_2 - sjo_1 < 0$ and so $y < 0$ similarly if $x < 0$ then $sjo_1 < sjo_2$ therefore $sjo_2 - sjo_1 > 0$ and so $y > 0$.

We conclude that single scale satisfies one permission. □

A.3 Uniform Detachment

Proposition 22. *Uniform detachment satisfies groundedness.*

Proof. We use a direct proof to prove that uniform detachment satisfies groundedness. Consider some uniform detachment function D_w. Take $o_1, o_2 \in O$ and $c \in C$ and $RR \subseteq R(o_1, o_2)$ such that $D_w(R, o_1, o_2, RR, c) = (v_1, v_2)$.

If $v_1 = +$ or $v_2 = -$ we prove that there exists a $(g, o_1) \in RR$. Then $sjo_1 - sro_2 > 0$ and or $sjo_2 - sro_1 < 0$. In order for $sjo_1 - sro_2 > 0$ we need $sjo_1 > sro_2$ and so $sjo_1 > 0$ this is only possible if there is a reason $r = (g, o_1) \in RR$ with $jw(r) > 0$. Similarly for $sjo_2 - sro_1 < 0$ to be true $sjo_2 < sro_1$ which means $sro_1 > 0$ which is only possible if there is a reason $r' = (g', o_1) \in RR$ with $rw(r') > 0$.

If $v_1 = -$ or $v_2 = +$ we prove that there exist a $(g, o_2) \in RR$. Then $sjo_1 - sro_2 < 0$ and or $sjo_2 - sro_1 > 0$. In order for $sjo_1 - sro_2 < 0$ we need $sjo_1 < sro_2$ and so $sro_2 > 0$ this is only possible if there is a reason $r = (g, o_2) \in RR$ with $rw(r) > 0$. Similarly, for $sjo_2 - sro_1 > 0$ to be true $sjo_2 > sro_1$ which means $sjo_2 > 0$ which is only possible if there is a reason $r' = (g', o_2) \in RR$ with $jw(r') > 0$.

We conclude that uniform detachment satisfies groundedness. □

Proposition 23. *Uniform detachment satisfies anonymity.*

Proof. We use a direct proof to prove that uniform detachment satisfies anonymity. Consider some uniform detachment function D_w. Take $o_1, o_2 \in O$ and $c \in C$ and $RR \subseteq R(o_1, o_2)$ such that $D_w(R, o_1, o_2, RR, c) = (v_1, v_2)$ and
$D_w(R, o_1, o_2, \{(\pi(g), o) \mid (g, o) \in RR\}, c) = (v_1', v_2')$.

Let $RR_{o'} = \{(g, o') \in RR \mid g \in G\}$ then we know that since all weights are equal we have $sjo_1 = sro_1 = |RR_{o_1}| = |\{(\pi(g), o_1) \mid (g, o_1) \in RR\}|$ and also for o_2 we have $sro_2 = sjo_2 = |RR_{o_2}| = |\{(\pi(g), o_2) \mid (g, o_2) \in RR\}|$. Then it follows that $v_1 = v_1'$ and $v_2 = v_2'$.

We conclude that uniform detachment satisfies anonymity. □

Proposition 24. *Uniform detachment satisfies unanimity.*

Proof. We use a direct proof to prove that uniform detachment satisfies unanimity. We assume RR to be non-empty, since for empty RR there is nothing to prove. Consider some uniform detachment function D_w. Take $o_1, o_2 \in O$ and $c \in C$ and $RR \subseteq R(o_1, o_2)$ such that there are $v_1, v_2 \in V$ with $D_w(R, o_1, o_2, RR, c) = (v_1, v_2)$.

If $RR = \{(g, o_1) \in RR : g \in G' \subseteq G\}$ then we know that due to uniform weights each weight is the same number. Therefore, since each reason favors o_1 we have $sjo_1 = |RR|$, $sro_2 = 0$, $sjo_2 = 0$ and $sro_1 = |RR|$. So $v_1 = +$ since $sjo_1 > sro_2$ and $v_2 = -$ since $sjo_2 < sro_1$, then $D_w(R, o_1, o_2, RR, c) = (+, -)$ similarly if each reason favors o_2 we find $D_w(R, o_1, o_2, RR, c) = (-, +)$.

We conclude that uniform detachment satisfies unanimity. □

Proposition 25. *Uniform detachment satisfies polarity monotony.*

Proof. We use direct proof to prove that uniform detachment satisfies polarity monotony. Consider some uniform detachment function D_w. Take $o_1, o_2 \in O$ and $c \in C$ and $RR \subseteq R(o_1, o_2)$ such that $D_w(R, o_1, o_2, RR, c) = (v_1, v_2)$. And take $r = (g, o_1) \in R(o_1, o_2) \backslash RR$ such that $D_w(R, o_1, o_2, RR \cup \{r\}, c) = (v'_1, v'_2)$.

We show that $v'_1 \geq v_1$; for v_1 we compare sjo_1 to sro_2 and for v'_1 we compare $sjo_1 + jw(r)$ and sro_2, observe that $sjo_1 \leq sjo_1 + jw(r)$ from this follows that $v'_1 \geq v_1$. Similarly we can show that $v'_2 \geq v_2$ by observing that for v_2 we compare sjo_2 to sro_1 and for v'_2 we compare sjo_2 with $sro_1 + rw(r)$, observe that $sro_1 \leq sro_1 + rw(r)$ from this follows that $v'_2 \leq v_2$.

When we add a reason $r' = (g, o_2)$ instead of r we find that $v'_1 \leq v_1$ since $sro_2 + rw(r') \geq sro_2$ and we find $v'_2 \geq v_2$ since $sjo_2 \leq sjo_2 + jw(r')$.

This proves polarity monotony holds for uniform detachment. We conclude that uniform detachment satisfies polarity monotony. □

Proposition 26. *Uniform detachment fails union monotony.*

Proof. We use a counter example to prove that uniform detachment fails union monotony. Consider some dual scale detachment function D_w. Take $o_1, o_2 \in O$ and $c \in C$ and $RR_1, RR_2 \subseteq R(o_1, o_2)$ such that $D_w(R, o_1, o_2, RR_1, c) = (v_1, v_2)$, $D_w(R, o_1, o_2, RR_2, c) = (v_1, v_2)$ and $D_w(R, o_1, o_2, RR_1 \cup RR_2, c) = (v'_1, v'_2)$. We show that it is possible for $v_1 \neq v'_1$ and $v_2 \neq v'_2$ to hold.

Let $r_1 = (g_1, o_1)$, $r_2 = (g_2, o_1)$,$r_3 = (g_3, o_1)$, $r_4 = (g_3, o_2)$, $r_5 = (g_5, o_2)$, $r_6 = (g_6, o_2)$, $r_7 = (g_7, o_2)$ and $RR_1 = \{r_1, r_2, r_3, r_4, r_5\}$ and $RR_2 = \{r_1, r_2, r_3, r_6, r_7\}$.

Then $v_1 = +$ since $jw(r_1)+jw(r_2)+jw(r_3) = 3 > 2 = rw(r_4)+rw(r_5) = rw(r_6)+rw(r_7)$, and $v_2 = -$ since $rw(r_1) + rw(r_2) + rw(r_3) = 3 > 2 = jw(r_4) + jw(r_5) = jw(r_6) + jw(r_7)$.

Furthermore, $v_1' = -$ since $jw(r_1)+jw(r_2)+jw(r_3) = 3 < 4 = rw(r_4)+rw(r_5)+rw(r_6)+rw(r_7)$ and $v_2' = +$ since $rw(r_1) + rw(r_2) + rw(r_3) = 3 < 4 = jw(r_4) + jw(r_5) = jw(r_6) + jw(r_7)$. Observe that $v_1 \neq v_1'$, and $v_2 \neq v_2'$.

We conclude that uniform detachment fails union monotony.

□

Proposition 27. *Uniform detachment satisfies sensitivity.*

Proof. We use a direct proof to prove that uniform detachment satisfies sensitivity. Consider some uniform detachment function D_w. Take $o_1, o_2 \in O$ and $c \in C$ and $RR \subseteq R(o_1, o_2)$ such that $D_w(R, o_1, o_2, RR, c) = (0, 0)$ and an $r \in R(o_1, o_2)$.

We show that for $(v_1, v_2) \in D_w(R, o_1, o_2, RR \cup \{r\}, c)$ that $v_1 \neq 0 \vee v_2 \neq 0$. Since $D_w(R, o_1, o_2, RR, c) = (0, 0)$ we know that $sjo_1 - sro_2 = 0 = sjo_2 - sro_1$. We also know that for $w(r) = (w_1, w_2)$ the weights are all equal and positive $w_1 = w_2 = w_0 > 0$.

From this it follows that if r is a reason for o_1 then $sjo_1 + jw(r) \geq sro_2$ and $sjo_2 \leq sro_1 + rw(r)$ therefore $v_1 = + \wedge v_2 = -$ is true. Similarly, if r is a reason for o_2 then $sjo_1 \leq sro_2 + rw(r)$ and $sjo_2 + jw(r) \geq sro_1$ and therefore $v_1 = - \wedge v_2 = +$ is true. This proves that sensitivity holds for uniform detachment.

We conclude that uniform detachment satisfies sensitivity. □

Proposition 28. *Uniform detachment satisfies input symmetry.*

Proof. We use a direct proof to prove that uniform detachment satisfies input symmetry. Consider some uniform detachment function D_w. Take $o_1, o_2 \in O$ and $c \in C$ and $RR \subseteq R(o_1, o_2)$ such that $D_w(R, o_1, o_2, RR, c) = (v_1, v_2)$ and $D_w(R, o_2, o_1, RR, c) = (v_1', v_2')$. We show that $(v_2, v_1) \in D_w(R, o_2, o_1, RR, c)$ and so that $v_1 = v_2'$ and that $v_2 = v_1'$. We have the following:

- $sjo_1 = \Sigma_{(g_1,o_1) \in RR} jw(R, o_1, o_2, c, (g_1, o_1)) = \Sigma_{(g_1,o_1) \in RR} jw((g_1, o_1))$
- $sro_2 = \Sigma_{(g_2,o_2) \in RR} rw(R, o_1, o_2, c, (g_2, o_2)) = \Sigma_{(g_2,o_2) \in RR} rw((g_2, o_2))$
- $sjo_2 = \Sigma_{(g_2,o_2) \in RR} jw(R, o_1, o_2, c, (g_2, o_2)) = \Sigma_{(g_2,o_2) \in RR} jw((g_2, o_2))$
- $sro_1 = \Sigma_{(g_1,o_1) \in RR} rw(R, o_1, o_2, c, (g_1, o_1)) = \Sigma_{(g_1,o_1) \in RR} rw((g_1, o_1))$

But also:

- $sjo'_1 = \Sigma_{(g_2,o_2)\in RR} jw(R, o_2, o_1, c, (g_2, o_2)) = \Sigma_{(g_2,o_2)\in RR} jw((g_2, o_2))$
- $sro'_2 = \Sigma_{(g_1,o_1)\in RR} rw(R, o_2, o_1, c, (g_1, o_1)) = \Sigma_{(g_1,o_1)\in RR} rw((g_1, o_1))$
- $sjo'_2 = \Sigma_{(g_1,o_1)\in RR} jw(R, o_2, o_1, c, (g_1, o_1)) = \Sigma_{(g_1,o_1)\in RR} jw((g_1, o_1))$
- $sro'_1 = \Sigma_{(g_2,o_2)\in RR} rw(R, o_2, o_1, c, (g_2, o_2)) = \Sigma_{(g_2,o_2)\in RR} rw((g_2, o_2))$

Observe that $sjo_1 = sjo'_2$, $sro_2 = sro'_1$, $sjo_2 = sjo'_1$, and $sro_1 = sro'_2$. Given the definition of uniform detachment we have $v_1 = v'_2$ exactly then when $sjo_1 = sjo'_2$ together with $sro_2 = sro'_1$. And we have $v_2 = v'_1$ exactly when $sjo_2 = sjo'_1$ and $sro_1 = sro'_2$. This proves that input symmetry holds for uniform detachment.

We conclude that uniform detachment satisfies input symmetry.

□

Proposition 29. *Uniform detachment satisfies no dilemmas.*

Proof. Suppose, toward a contradiction, that uniform detachment does not satisfy No Dilemmas. This means that there is some uniform detachment function D such that, for some o_1, o_2, $RR \subseteq R(o_1, o_2)$, and c, we have $D(R, o_1, o_2, RR, c) = (-,-)$. By Definition 19 (Uniform), we know that there is some weighting function w such that $D_w(R, o_1, o_2, RR, c) = (-,-)$ and $jw = rw = w_0$. By Definition 4 (Additive Dual Scale Weighing), we know, first, that $\Sigma_{(g_1,o_1)\in RR} jw((g_1, o_1)) - \Sigma_{(g_2,o_2)\in RR} rw((g_2, o_2)) < 0$, as well as, second, that $\Sigma_{(g_2,o_2)\in RR} jw((g_2, o_2)) - \Sigma_{(g_1,o_1)\in RR} rw((g_1, o_1)) < 0$. But $jw = rw$ tells us that $\Sigma_{(g_1,o_1)\in RR} jw((g_1, o_1)) = \Sigma_{(g_1,o_1)\in RR} rw((g_1, o_1)) = x$. Similarly, it tells us that $\Sigma_{(g_2,o_2)\in RR} jw((g_2, o_2)) = \Sigma_{(g_2,o_2)\in RR} rw((g_2, o_2)) = y$. So, we have $y < x$ and $x > y$, which is a contradiction. □

Proposition 30. *Uniform detachment satisfies one permission.*

Proof. We use a direct proof to prove that uniform detachment satisfies one permission. Consider some uniform detachment function D_w. Take $o_1, o_2 \in O$ and $c \in C$ and $RR \subseteq R(o_1, o_2)$ such that $D_w(R, o_1, o_2, RR, c) = (v_1, v_2)$. For D_w to be a uniform detachment function, we require that $\forall r \in R. jw(r) = rw(r) = w_0$.

The principle is satisfied if three output pairs are **not** possible: $(+,+)$, $(+,0)$, and $(0,+)$. These occur when $sjo_1 \geq sro_2$ together with $sjo_2 \geq sro_1$.

$$x = sjo_1 - sro_2 = \Sigma_{(g_1,o_1)\in RR} jw((g_1,o_1)) - \Sigma_{(g_2,o_2)\in RR} rw((g_2,o_2))$$

$$y = sjo_2 - sro_1 = \Sigma_{(g_2,o_2)\in RR} jw((g_2,o_2)) - \Sigma_{(g_1,o_1)\in RR} rw((g_1,o_1))$$

We prove that $x > 0$ if and only if $y < 0$. Since $\forall r \in R.jw(r) = rw(r) = w_0$, we can write:

$$x = sjo_1 - sjo_2 = \Sigma_{(g_1,o_1)\in RR} jw((g_1,o_1)) - \Sigma_{(g_2,o_2)\in RR} jw((g_2,o_2))$$

$$y = sjo_2 - sjo_1 = \Sigma_{(g_2,o_2)\in RR} jw((g_2,o_2)) - \Sigma_{(g_1,o_1)\in RR} jw((g_1,o_1))$$

From this we can see that if $x > 0$ then $sjo_1 > sjo_2$ therefore $sjo_2 - sjo_1 < 0$ and so $y < 0$ similarly if $x < 0$ then $sjo_1 < sjo_2$ therefore $sjo_2 - sjo_1 > 0$ and so $y > 0$.

We conclude that uniform satisfies one permission. □

A.4 Maximum Detachment

Proposition 31. *Maximum detachment satisfies groundedness.*

Proof. We use a direct proof to prove that Maximum detachment satisfies groundedness. Consider some Maximum detachment function D_w^M. Take $o_1, o_2 \in O$ and $c \in C$ and $RR \subseteq R(o_1, o_2)$ such that $D_w^M(R, o_1, o_2, RR, c) = (v_1, v_2)$.

If $v_1 = +$ or $v_2 = -$ we prove that there exists a $(g, o_1) \in RR$. Then $mjo_1 - mro_2 > 0$ and or $mjo_2 - mro_1 < 0$. In order for $mjo_1 - mro_2 > 0$ we need $mjo_1 > mro_2$ and so $mjo_1 > 0$ this is only possible if there is a reason $r = (g, o_1) \in RR$ with $jw(r) > 0$. Similarly, for $mjo_2 - mro_1 < 0$ to be true, $mjo_2 < mro_1$, which means $mro_1 > 0$, which is only possible if there is a reason $r' = (g', o_1) \in RR$ with $rw(r') > 0$.

If $v_1 = -$ or $v_2 = +$ we prove that there exists a $(g, o_2) \in RR$. Then $mjo_1 - mro_2 < 0$ and or $mjo_2 - mro_1 > 0$. In order for $mjo_1 - mro_2 < 0$ we need $mjo_1 < mro_2$ and so $mro_2 > 0$ this is only possible if there is a reason $r = (g, o_2) \in RR$ with $rw(r) > 0$. Similarly, for $mjo_2 - mro_1 > 0$

to be true, $mjo_2 > mro_1$, which means $mjo_2 > 0$, which is only possible if there is a reason $r' = (g', o_2) \in RR$ with $jw(r') > 0$.

We conclude that Maximum detachment satisfies groundedness. □

Proposition 32. *Maximum detachment fails anonymity.*

Proof. We use a counterexample to prove that there exists a max weight function that fails anonymity. Let $r_1 = (g_1, o_1)$ and $r_2 = (g_2, o_2)$, and let $(v_1, v_2) = D_w^M(R, o_1, o_2, \{r_1, r_2\}, c)$. And define π such that $r'_1 = (\pi(g_1), o_1)$ and $r'_2 = (\pi(g_2), o_2)$, such that $(v'_1, v'_2) = D_w^M(R, o_1, o_2, \{r'_1, r'_2\}, c)$. We show that there exists w such that $v_1 \neq v'_1 \vee v_2 \neq v'_2$.

Since r_1 is a different reason than r'_1 it need not be that $w(r_1) = w(r'_1)$, and similarly $w(r_2) = w(r'_2)$ does not need to hold.

Let $w(r_1) = (1, 3)$, $w(r_2) = (2, 1)$, $w(r'_1) = (2, 1)$, and $w(r'_2) = (3, 1)$ then $v_1 = 0$ since $jw(r_1) = 1 = rw(r_2)$ and $v_2 = -$ since $jw(r_2) = 2 < 3 = rw(r_1)$. Moreover, $v'_1 = +$ since $jw(r'_1) = 2 > 1 = rw(r'_2)$ and $v'_2 = +$ since $jw(r'_2) = 3 > 1 = rw(r'_1)$. So we have $v_1 = 0 \neq + = v'_1$ and $v_2 = - \neq + = v'_2$. This completes the counter example.

We conclude that Maximum detachment fails anonymity.

□

Proposition 33. *Maximum detachment fails unanimity.*

Proof. We use a counterexample to prove that there exists a weight function that fails unanimity. We assume RR to be non-empty, since for empty RR there is nothing to prove.

Let $r_1 = (g_1, o_1)$ and let $(v_1, v_2) = D_w^M(R, o_1, o_2, \{r_1\}, c)$, such that $w(r_1) = (x, y)$.

We cannot have $x = 0$ and $y = 0$ otherwise $r_1 \notin RR$. Then if $x = 0$ and $y \neq 0$ we have $D_w^M(R, o_1, o_2, \{r_1\}, c) = (0, -)$. And if $x \neq 0$ and $y = 0$ we have $D_w^M(R, o_1, o_2, \{r_1\}, c) = (+, 0)$. Neither of these outputs is $(+, -)$. This completes the counterexample.

We conclude that maximum detachment fails unanimity.

□

Proposition 34. *Maximum detachment satisfies polarity monotony.*

Proof. We use a direct proof to prove that Maximum detachment satisfies polarity monotony. Consider some maximum detachment function D_w^M. Take $o_1, o_2 \in O$ and $c \in C$ and $RR \subseteq R(o_1, o_2)$ such that $D_w^M(R, o_1, o_2, RR, c) = (v_1, v_2)$. And take $r = (g, o_1) \in R(o_1, o_2) \backslash RR$ such that $D_w^M(R, o_1, o_2, RR \cup \{r\}, c) = (v_1', v_2')$.

We show that $v_1' \geq v_1$; for v_1 we compare mjo_1 to mro_2 and for v_1' we compare $max(mjo_1, jw(r))$ and mro_2, observe that $mjo_1 \leq max(mjo_1, jw(r))$ from this follows that $v_1' \geq v_1$. Similarly, we can show that $v_2' \geq v_2$ by observing that for v_2 we compare mjo_2 to mro_1 and for v_2' we compare mjo_2 with $max(mro_1, rw(r))$, observe that $mro_1 \leq max(mro_1, rw(r))$ from this follows that $v_2' \leq v_2$.

When we add a reason $r' = (g, o_2)$ instead of r we find that $v_1' \leq v_1$ since $max(mro_2, rw(r')) \geq mro_2$ and we find $v_2' \geq v_2$ since $mjo_2 \leq max(mjo_2, jw(r'))$.

We conclude that maximum detachment satisfies polarity monotony.

□

Proposition 35. *Maximum detachment satisfies union monotony.*

Proof. We use a direct proof to prove that Maximum detachment satisfies union monotony. Consider some maximum detachment function D_w^M. Take $o_1, o_2 \in O$ and $c \in C$ and $RR', RR'' \subseteq R(o_1, o_2)$ such that $D_w^M(R, o_1, o_2, RR', c) = (v_1, v_2)$, $D_w^M(R, o_1, o_2, RR'', c) = (v_1, v_2)$ and $(v_1', v_2') \in D_w^M(R, o_1, o_2, RR' \cup RR'', c)$. We show that $v_1 = v_1'$ and that $v_2 = v_2'$.

Since $D_w^M(R, o_1, o_2, RR', c) = (v_1, v_2)$ and $D_w^M(R, o_1, o_2, RR'', c) = (v_1, v_2)$ we have that mjo_1' relates to mro_2' in the same way as mjo_1'' relates to mro_2'' and similarly for mjo_2' to mro_1' and mjo_2'' to mro_1''. Also from the definition of maximum detachment we know that application of the maximum function gives us the following: $mjo_1''' = max(mjo_1', mjo_1'')$, $mro_2''' = max(mro_2', mro_2'')$, $mjo_2''' = max(mjo_2', mjo_2'')$, and $mro_1''' = max(mro_1', mro_1'')$.

From this, we can derive that the comparisons remain intact. For example if $v_1 = +$ then both $mjo_1' > mro_2'$ and $mjo_1'' > mro_2''$ therefore $mjo_1''' = max(mjo_1', mjo_1'') > max(mro_2', mro_2'') = mro_2'''$, and we find that $v_1 = v_1'$ for $v_1 = +$. However this holds for all values and we find that $v_1 = v_1'$ and that $v_2 = v_2'$. This proves that union monotony holds for maximum detachment.

We conclude that maximum detachment satisfies union monotony.

□

Proposition 36. *Maximum detachment fails sensitivity.*

Proof. We use a counterexample to prove that there exists a weight function that fails sensitivity. Consider some Maximum detachment function D_w^M. Take $o_1, o_2 \in O$ and $c \in C$ and $RR \subseteq R(o_1, o_2)$ such that $D_w(R, o_1, o_2, RR, c) = (0, 0)$ and $r \in R(o_1, o_2)$ such that $(v_1, v_2) \in D_w(R, o_1, o_2, RR \cup \{r\}, c)$.

From $D_w(R, o_1, o_2, RR, c) = (0, 0)$ we know that $0 = mjo_1 - mro_2$ and $0 = mjo_2 - mro_1$, i.e. that the strongest justifying reason for o_1 is as strong as the strongest requiring reason for o_2 and similarly that the strongest justifying reason for o_2 is as strong as the strongest requiring reason for o_1.

Then if we have $r = (g, o_1)$ a reason that favors o_1 such that $jw(r) < mjo_1$ and $rw(r) < mro_1$ then the output does not change $D_w(R, o_1, o_2, RR \cup \{r\}, c) = (0, 0)$, since the maximums do not change. This proves that maximum detachment fails sensitivity.

We conclude that Maximum detachment fails sensitivity. □

Proposition 37. *Maximum detachment satisfies input symmetry.*

Proof. We use a direct proof to prove that Maximum detachment satisfies input symmetry. Consider some Maximum detachment function D_w^M. Take $o_1, o_2 \in O$ and $c \in C$ and $RR \subseteq R(o_1, o_2)$ such that $D_w^M(R, o_1, o_2, RR, c) = (v_1, v_2)$ and $D_w^M(R, o_2, o_1, RR, c) = (v_1', v_2')$.

We show that $(v_2, v_1) \in D_w^M(R, o_2, o_1, RR, c)$ and so that $v_1 = v_2'$ and that $v_2 = v_1'$.

We have the following:

- $mjo_1 = \max(\{jw(R, o_1, o_2, c, (g_1, o_1)) \mid (g_1, o_1) \in RR\})$
- $mro_2 = \max(\{rw(R, o_1, o_2, c, (g_2, o_2)) \mid (g_2, o_2) \in RR\})$
- $mjo_2 = \max(\{jw(R, o_1, o_2, c, (g_2, o_2)) \mid (g_2, o_2) \in RR\})$
- $mro_1 = \max(\{rw(R, o_1, o_2, c, (g_1, o_1)) \mid (g_1, o_1) \in RR\})$

But also:

- $mjo'_1 = \max(\{jw(R, o_2, o_1, c, (g_2, o_2)) \mid (g_2, o_2) \in RR\})$
- $mro'_2 = \max(\{rw(R, o_2, o_1, c, (g_1, o_1)) \mid (g_1, o_1) \in RR\})$
- $mjo'_2 = \max(\{jw(R, o_2, o_1, c, (g_1, o_1)) \mid (g_1, o_1) \in RR\})$
- $mro'_1 = \max(\{rw(R, o_2, o_1, c, (g_2, o_2)) \mid (g_2, o_2) \in RR\})$

Observe that $mjo_1 = mjo'_2$, $mro_2 = mro'_1$, $mjo_2 = mjo'_1$, and $mro_1 = mro'_2$.

Given the definition of maximum detachment, we have $v_1 = v'_2$ exactly when $mjo_1 = mjo'_2$ together with $mro_2 = mro'_1$. And we have $v_2 = v'_1$ exactly when $mjo_2 = mjo'_1$ and $mro_1 = mro'_2$. This proves that input symmetry holds for maximum detachment.

We conclude that maximum detachment satisfies input symmetry. □

Proposition 38. *Maximum detachment fails no dilemmas.*

Proof. We use a counterexample to prove that there exists a weight function that fails no dilemmas.

We use proof by counterexample; we present a weighing function such that maximum detachment fails No Dilemmas. Let $r_1 = (g_1, o_1)$ and $r_2 = (g_2, o_2)$. And let $(v_1, v_2) = D_w(R, o_1, o_2, \{r_1, r_2\}, c)$ such that $w(R, o_1, o_2, c, r_1) = (1, 3)$, $w(R, o_1, o_2, c, r_2) = (1, 3)$. Then since $jw(r_1) - rw(r_2) = 1 - 3 = -2$ we have $v_1 = -$ and since $jw(r_2) - rw(r_1) = 1 - 3 = -2$ we have $v_2 = -$, then $D_w(\{r_1, r_2\}) = (-, -)$, which completes the counterexample.

We conclude that Maximum detachment fails no dilemma.

□

Proposition 39. *Maximum detachment fails one permission.*

Proof. We use a counterexample to prove that there exists a weight function that fails one permission. We complete the counterexample if we can define a weight function such that there exists a $(v_1, v_2) = D^M_w(R, o_1, o_2, RR, c)$, where $(v_1, v_2) = (+, 0)$.

Let $r_1 = (g_1, o_1)$ and $r_2 = (g_2, o_2)$, also let $(v_1, v_2) = D^M_w(R, o_1, o_2, \{r_1, r_2\}, c)$, such that $w(r_1) = (3, 1)$ and $w(r_1) = (1, 1)$. Then $v_1 = +$ since $mjo_1 = 3 > 1 = mro_2$ and $v_2 = 0$ since $mjo_2 = 1 = mro_1$, therefore $D^M_w(\{r_1, r_2\}) = (+, 0)$, which completes the counterexample.

We conclude that maximum detachment fails one permission.

□

Appendix B

Code

B.1 IHiBO: reason based balancing smart contract

The code that was used to run the experiments in chapter 3. This can be found on the following address: https://github.com/Trinominal/IHiBO_2.0/blob/main/2.0/contracts/Balancing.sol.

```
// SPDX-License-Identifier: MIT

pragma solidity ^0.5.1;

pragma experimental ABIEncoderV2; // experimental feature | do not
   use in live deployments

// import "./DirectedGraph.sol";
// import "./EnumerableMap.sol";
import "./HitchensUnorderedKeySet.sol";

// /*
```

```
contract Balancing {
    using HitchensUnorderedKeySetLib for HitchensUnorderedKeySetLib
        .Set;

    struct Reason {
        string justification;
        string issue;
        string polarity;
    }

    struct Context {
        uint256 rcount;
        string[] reasons;
        string issue;
    }

    struct Decision {
        Context context;
        mapping(uint256 => uint256) weights;
        uint256 outputPolarity;
    }

    HitchensUnorderedKeySetLib.Set reasonsIds;
    mapping(uint256 => Reason) reasons;
    string issueTBD;
    Context contextTBD;
    mapping(uint256 => uint256) weights;

    // HitchensUnorderedKeySetLib.Set decisionsIds;
```

```
// mapping(uint256 => Decision) decisions;

mapping(address => HitchensUnorderedKeySetLib.Set) sources;

mapping(address => uint256) reputations;

// event Output(uint256 key);
event Output(string reasons, string issue, string polarity);


constructor() public {
    // for now constructor sets issue to 0 // this is a bit
       silly as there will only be one issue in the contract
       but there can be reasons that are relevant for another
       issue.

    issueTBD = '0';
}

// create functon to initialize contract. create access right
   for admin to set issue and read rights of the discourse.

function convertToString(uint256 value)
    internal
    pure
    returns (string memory)
{
    if (value == 0) {
        return "0";
```

```
    }

    uint256 temp = value;
    uint256 digits;

    while (temp != 0) {
        digits++;
        temp /= 10;
    }

    bytes memory buffer = new bytes(digits);

    while (value != 0) {
        digits--;
        buffer[digits] = bytes1(uint8(48 + (value % 10)));
        value /= 10;
    }

    return string(buffer);
}

function compareStrings(string memory a, string memory b)
   public view returns (bool) {
    return (keccak256(abi.encodePacked((a))) == keccak256(abi.
       encodePacked((b))));
}

function setReputation(uint256 rep, address subject)
    public
```

```
{
    assert(msg.sender!=subject);
    reputations[subject] = rep;
}

function returnWeight(uint256 r)
    public
    view
    returns (uint256 weight)
{
    weight = weights[r];
}

function changeWeight(uint256 reasonID, uint256 newWeight)
    public
{
    assert(reasonID != 0);
    weights[reasonID] = newWeight;
}

function getWeights()
    public
    view
    returns (
        uint256[] memory
    )
{
    uint256 rs = reasonsIds.count();
```

```
    uint256[] memory _weights = new uint256[](rs);

    for (uint256 i = 0; i < rs; i++) {
        _weights[i] = weights[i];
    }

    return _weights;
}

function getIssue()
    public
    view
    returns ( string memory)
{
    return issueTBD;
}

function setIssue(string memory issue)
    public
{
    // TODO change to depend on access rights and can only be
       called at the start
    if(true) {
        issueTBD = issue;
    }
}

function getReasons()
    public
```

```
    view
    returns (
        string[] memory justifications,
        string[] memory issues,
        string[] memory polarities
    )
{

    uint256 reasonsCount = reasonsIds.count();

    justifications = new string[](reasonsCount);
    issues = new string[](reasonsCount);
    polarities = new string[](reasonsCount);

    for (uint256 i = 0; i < reasonsIds.count(); i++) {
        uint256 reasonId = uint256(reasonsIds.keyAtIndex(i));
        Reason storage reason = reasons[reasonId];
        justifications[i] = reason.justification;
        issues[i] = reason.issue;
        polarities[i] = reason.polarity;
    }
}

function voteOnReason
(
    string memory justification,
    string memory issue,
    string memory polarity,
    uint256 magnitude
```

```
)
    public
    returns (int256 re)
{

    re = 0;
    // check that reason is not known yet.
    for (uint256 j = 1; j < reasonsIds.count() + 1; j++) {
        // if reason is known
        uint256 reasonId = uint256(reasonsIds.keyAtIndex(j));
        if (compareStrings(reasons[reasonId].justification,
            justification) &&
        compareStrings(reasons[reasonId].issue, issue) &&
        compareStrings(reasons[reasonId].polarity, polarity)) {
            // conclude reason is already in reasons
            // increase weight by +1
            weights[reasonId-1] = weights[reasonId-1] +
                magnitude*reputations[msg.sender];

            HitchensUnorderedKeySetLib.Set storage source =
                sources[
                 msg.sender
            ];
            source.insert(bytes32(reasonId));
            re = int256(reasonId);
            break;
        }
    }
```

```
    if (re == 0) {// reason is new add reason and set weight to
        1
        uint256 reasonID = reasonsIds.count() + 1;
        reasonsIds.insert(bytes32(reasonID));
        Reason storage reason = reasons[reasonID];
        reason.justification = justification;
        reason.issue = issue;
        reason.polarity = polarity;
        weights[reasonID-1] = magnitude*reputations[msg.sender
            ];

        HitchensUnorderedKeySetLib.Set storage source = sources
            [
             msg.sender
        ];
        source.insert(bytes32(reasonID));
        re = int256(reasonID);
    }
}


function procedureAdditive()
    public
    // view
    returns (string memory dec)
{

    dec = '';
    uint256 magn = 0;
```

```
uint256 pos = 0;
uint256 ppos = 0;
uint256 neg = 0;
uint256 nneg = 0;
int256 sum = 0;
uint256 neut = 0;
string memory cs = '';
uint256 rs = reasonsIds.count();


for (uint256 i = 0; i < rs; i++){

    uint256 reasonId = uint256(reasonsIds.keyAtIndex(i));
    Reason storage reason = reasons[reasonId];

    cs = string(abi.encodePacked(cs, "␣", i, "␣", reason.
        justification));

    if (compareStrings(reason.issue, issueTBD)) {
        if (compareStrings(reason.polarity, '0')) {
            neut += uint256(weights[i]);
        }
        else if (compareStrings(reason.polarity, '-')) {
            neg += uint256(weights[i]);
        }
        else if (compareStrings(reason.polarity, '+')) {
            pos += uint256(weights[i]);
        }
        else if (compareStrings(reason.polarity, '--')) {
```

```
                nneg += uint256(weights[i]);
            }
            else if (compareStrings(reason.polarity, '++')) {
                ppos += uint256(weights[i]);
            }
        }
    }

    sum = int256(pos) + 2*int256(ppos) - (int256(neg) + 2*
        int256(nneg));
    magn = sum < 0 ? uint256(-sum) : uint256(sum);

    // pos += 2*ppos;
    // neg += 2*nneg;
    // magn = pos > neg ? pos - neg : neg - pos;
    // magn = 2*magn < neut ? neut-magn : (2*magn-neut)/2;
    // string memory Magn = convertToString(magn);

    if (!(2*magn > neut)) {
        magn = neut;
        string memory Magn = convertToString(magn);
        dec = string(abi.encodePacked(dec, '(', Magn, ', ', '0'
            , ')'));
        // dec = string(magn + '0');
        // dec = '0';
    }
    else if (sum > 0 && ppos > pos) {
        // 2*pos > 2*(neut+2*neg)) {
        magn = ppos;
```

```
        string memory Magn = convertToString(magn);
        dec = string(abi.encodePacked(dec, '(', Magn, ',␣', '++
            ', ')'));
        // dec = string(magn + '++');
        // dec = '++';
    }
    else if (sum > 0) {
        // 2*pos > neut+2*neg) {
        magn = pos;
        string memory Magn = convertToString(magn);
        dec = string(abi.encodePacked(dec, '(', Magn, ',␣', '+'
            , ')'));
        // dec = string(magn + '+');
        // dec = '+';
    }
    else if (sum < 0 && nneg > neg) {
        // 2*neg > 2*(neut+2*pos)) {
        magn = nneg;
        string memory Magn = convertToString(magn);
        dec = string(abi.encodePacked(dec, '(', Magn, ',␣', '--
            ', ')'));
        // dec = string(magn + '--');
        // dec = '--';
    }
    else if (sum < 0) {
        // 2*neg > neut+2*pos) {
        magn = neg;
        string memory Magn = convertToString(magn);
```

```
            dec = string(abi.encodePacked(dec, '(', Magn, ', ', '-'
                , ')'));
            // dec = string(magn + '-');
            // dec = '-';
        }
        else {
            magn = 0;
            string memory Magn = convertToString(magn);
            dec = string(abi.encodePacked(dec, '(', Magn, ', ', '0'
                , ')'));
        }


    // untested code for emiting decisionKey
    // /*
        // contextTBD.rcount = rs;
        // contextTBD.reasons = reasons;
        // contextTBD.issue = issueTBD;

        // uint256 decisionID = decisionsIds.count() + 1;
        // decisionsIds.insert(bytes32(decisionID));
        // Decision storage decision = decisions[decisionID];
        // decision.context = contextTBD;
        // decision.weights = weights;
        // decision.outputPolarity = uint256(sum);

        // emit Output(decisionID);
        // */
```

```
        // emit Output(42); // ^^'
        emit Output(cs ,issueTBD, dec);

        return dec;
    }

}
```

B.2 Python Simulated Trading Experiment

This code was created using multiple iterations of LLM generated code. Two LLM's were used: ChatGPT, and Claude.

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt


# -------------------------------
# Grounds (boolean indicators)
# -------------------------------
def compute_grounds(prices, day, portfolio, grounds):
    """Return■boolean■grounds■for■the■given■day■and■portfolio■state
       """
    gs = {g: False for g in grounds}

    if day < 20:  # not enough history for indicators
        return gs

    if "MA_crossover" in grounds: # moving average crossover
        window_short = 10
```

```
window_long = 20
ma_short = prices.iloc[day - window_short:day].mean()
ma_long = prices.iloc[day - window_long:day].mean()
ma_crossover = ma_short > ma_long
gs["MA_crossover"] = ma_crossover # true means short term
    trend overtakes long term -> buy

# RSI
if "RSI_low" in grounds or "RSI_high" in grounds: # Relative
    Strength Index
    delta = prices.diff().iloc[1:day]
    gain = delta.where(delta > 0, 0).rolling(14).mean().iloc
        [-1]
    loss = -delta.where(delta < 0, 0).rolling(14).mean().iloc
        [-1]
    rs = gain / loss if loss != 0 else np.inf
    rsi = 100 - (100 / (1 + rs))
    if "RSI_low" in grounds:
        rsi_low = rsi < 30
        gs["RSI_low"] = rsi_low # true means oversold -> buy
    if "RSI_high" in grounds:
        rsi_high = rsi > 70
        gs["RSI_high"] = rsi_high # true means overbought ->
            sell

# MACD (12 vs 26 EMA)
if "MACD_positive" in grounds: # moving average convergence/
    divergence
    ema12 = prices.ewm(span=12, adjust=False).mean().iloc[day]
```

```
        ema26 = prices.ewm(span=26, adjust=False).mean().iloc[day]
        macd_positive = ema12 > ema26
        gs["MACD_positive"] = macd_positive # true means upward
            momentum -> buy
        gs["MACD_negative"] = not macd_positive # true means
            downward momentum -> sell

    # Equity-based grounds
    current_price = prices.iloc[day]
    stock_value = portfolio["stock"] * current_price
    if "EquityGainSell" in grounds:
        equity_gain_sell = stock_value > portfolio["cost_basis"]
        gs["EquityGainSell"] = equity_gain_sell
    if "EquityGainBuy" in grounds:
        equity_gain_buy = portfolio["cash"] > current_price  #
            enough cash to buy at least 1
        gs["EquityGainBuy"] = equity_gain_buy

    return gs

# ------------------------------
# Agent and Collective Agent
# ------------------------------
class Agent:
    def __init__(self, name, style, grounds):
        self.name = name
        self.style = style
        self.grounds = grounds
        self.weights = self.generate_weights()
```

```python
        self.portfolio = {"cash": 10000, "stock": 100, "cost_basis"
            : 10000}
        self.equity = 10000 + 100*100
        self.history = []
        self.spent = []

    def generate_weights(self):
        weights = {}
        rng = np.random.default_rng()

        if self.style == "collective":
            return weights

        # Style biases
        style_bias = {
            "conservative": {"Buy": 1/6, "Sell": 1/6, "Hold": 4/6},
            "balanced": {"Buy": 1/3, "Sell": 1/3, "Hold": 1/3},
            "aggressive": {"Buy": 5/12, "Sell": 5/12, "Hold": 1/6},
            "collective": {"Buy": 0, "Sell": 0, "Hold": 0},
        }

        for g in self.grounds:
            for a in ["Buy", "Sell", "Hold"]:
                for b in ["Buy", "Sell", "Hold"]:
                    if a == b:
                        continue
                    # Randomized weights scaled by style
                    jw = rng.uniform(0.5, 2.0) * style_bias[self.
                        style][a]
```

```
                rw = rng.uniform(0.5, 2.0) * style_bias[self.
                    style][b]
                weights[(g, a, b)] = (jw, rw)
        return weights

    def decide(self, day, prices):
        grounds = compute_grounds(prices, day, self.portfolio, self
            .grounds)
        options = ["Buy", "Sell", "Hold"]
        permitted = {opt: True for opt in options}

        for a in options:
            for b in options:
                if a == b:
                    continue
                jw_sum = sum(
                    self.weights[(g, a, b)][0]
                    for g, val in grounds.items()
                    if val
                )
                rw_sum = sum(
                    self.weights[(g, b, a)][1]
                    for g, val in grounds.items()
                    if val
                )
                if jw_sum < rw_sum:
                    permitted[a] = False

        for opt in options:
```

```
            if permitted[opt]:
                return opt
        return "Hold"

    def update_portfolio(self, decision, price):
        if decision == "Buy" and self.portfolio["cash"] >= price:
            self.portfolio["stock"] += 1
            self.portfolio["cash"] -= price
            self.portfolio["cost_basis"] += price
        elif decision == "Sell" and self.portfolio["stock"] > 0:
            self.portfolio["cash"] += price
            self.portfolio["stock"] -= 1
            self.portfolio["cost_basis"] -= (
                self.portfolio["cost_basis"] / (self.portfolio["
                    stock"] + 1)
            )
        # Update equity
        self.equity = (
            self.portfolio["cash"] + self.portfolio["stock"] *
                price
        )
        self.history.append(self.equity)

        self.spent.append(self.portfolio["cost_basis"])


class CollectiveAgent(Agent):
    def __init__(self, agents, grounds):
```

```
        super().__init__("Collective", style="collective", grounds=
            grounds)
        self.agents = agents
        self.weights = self.aggregate_weights(grounds)

    def aggregate_weights(self, grounds):
        agg = {}
        for g in grounds:
            for a in ["Buy", "Sell", "Hold"]:
                for b in ["Buy", "Sell", "Hold"]:
                    if a == b:
                        continue
                    jw_sum, rw_sum = 0.0, 0.0
                    for agent in self.agents:
                        jw, rw = agent.weights[(g, a, b)]
                        jw_sum += jw
                        rw_sum += rw
                    agg[(g, a, b)] = (jw_sum, rw_sum)
        return agg

# Maybe add that if no one would permit an option then that option
    is not considered. Not sure maybe given the reasons the option
    should be permitted?

# ------------------------------
# Experiment runner
# ------------------------------
def run_experiment(prices, grounds):
```

```
    conservative = Agent("Conservative", style="conservative",
        grounds=grounds)
    balanced = Agent("Balanced", style="balanced", grounds=grounds)
    aggressive = Agent("Aggressive", style="aggressive", grounds=
        grounds)
    collective = CollectiveAgent([conservative, balanced,
        aggressive], grounds)

    agents = [conservative, balanced, aggressive, collective]
    decisions = {a.name: [] for a in agents}
    warmup = 50

    for day in range(warmup, len(prices)):
        for agent in agents:
            choice = agent.decide(day, prices)
            agent.update_portfolio(choice, prices.iloc[day])
            decisions[agent.name].append(choice)

    return agents, decisions

# ------------------------------
# Plotting
# ------------------------------
def plot_results(prices, agents, decisions, warmup=50, show=True):
    # Build a clean x-axis (dates) after warmup and the price
        Series aligned to it
    idx_full = prices.index
    idx = idx_full[warmup:]            # x-axis for decisions/equity
    px  = prices.iloc[warmup:]         # prices aligned to idx
```

```
fig, axes = plt.subplots(3, 1, figsize=(12, 10), sharex=True)
ax0, ax1, ax2 = axes

# --- Price + decisions ---
ax0.plot(idx_full, prices.values, label="Price")

for name, decs in decisions.items():
    decs = list(decs)  # one decision per post-warmup day
    # positions relative to the post-warmup window
    buy_pos  = np.array([i for i, d in enumerate(decs) if d ==
        "Buy"], dtype=int)
    sell_pos = np.array([i for i, d in enumerate(decs) if d ==
        "Sell"], dtype=int)

    # Use idx[...] for x (dates) and px.iloc[...] for y (
        positional)
    if buy_pos.size:
        ax0.scatter(idx[buy_pos], px.iloc[buy_pos], marker="^
            ", s=36, alpha=0.7, label=f"{name} Buy")
    if sell_pos.size:
        ax0.scatter(idx[sell_pos], px.iloc[sell_pos], marker="v
            ", s=36, alpha=0.7, label=f"{name} Sell")

ax0.set_title("Market Data and Decisions")
ax0.legend()

# --- Equity curves (length-safe) ---
for agent in agents:
```

```
    hist = np.asarray(agent.history, dtype=float)
    # history should be post-warmup length; still guard for
        mismatches
    L = min(len(hist), len(idx))
    if L > 0:
        ax1.plot(idx[:L], hist[:L], label=agent.name)

ax1.set_title("Equity Curves")
ax1.legend()

# --- Equity - Price curve ---
for agent in agents:
    hist = np.asarray(agent.history, dtype=float)
    spent = np.asarray(agent.spent, dtype=float)
    # history should be post-warmup length; still guard for
        mismatches
    L = min(len(hist), len(idx))
    if L > 0:
        # not really proper to divide by 100 this should be the
             stock number at that time. I divided by 100 since
            that is what they start with. maybe should also be +
             money
        # how much value was gaind irrespective of the index.
        # at start agent has 10000 + value(100 stock) we want
            to track the value gained irrespective of the
            passive change in value from the index.
        ax2.plot(idx[:L], hist[:L]-spent[:L], label=agent.name)
# ax2.plot(idx_full, prices.values, label="Price")
ax2.set_title("Equity - expenses")
```

```python
    ax2.legend()

    plt.savefig("trading_agents_experiment.png", dpi=150)
    plt.tight_layout()
    if show:
        plt.show()


# ------------------------------
# Main
# ------------------------------
if __name__ == "__main__":
    np.random.seed(42)
    n_days = 50+356*2
    prices = pd.Series(np.cumsum(np.random.randn(n_days)) + 100)

    grounds = [
        "MA_crossover", "RSI_low", "RSI_high", "MACD_positive", "
            MACD_negative"
        , "EquityGainSell", "EquityGainBuy"
    ]

    agents, decisions = run_experiment(prices, grounds)
    plot_results(prices, agents, decisions)

    # Summary
    print("\nFinal equities:")
    for agent in agents:
        print(f"{agent.name}: {agent.equity:.2f}")
```

Appendix C

Applications

In this section multiple use-cases are considered, this serves two purposes. The first is that it makes the usage more intuitive by showing how the voting plays out in a particular application domain. And secondly, it shows the boundaries of applicability through areas where the application of the system is less obvious.

C.1 Milk comes from a factory

There is this story in which someone asks a child where milk comes from. The child replies that the milk comes from a factory. This is partly true today; however, the correct answer is that milk comes from a cow. The story demonstrates how removed modern people are from reality. People grow up in cities and do not know normal things like cows produce milk. This is because they do not see the cows and do not see the clues that show that they produce the milk. Maybe this example is a bit artificial; however, in the story of IHiBO this is relevant. That is, it is important to document the information taken into account during the process of determining what is permitted and what is not. However, at the same time, due to multiple constraints, it is not feasible to share and then document everything that ties into the conclusions. To some extent, information will be lost and a choice can be made in how much is lost. If a good choice is made in this regard, a balance is found such that the documentation and processing cost are low, while it is not forgotten that a cow produces the milk and not the factory.

The opposite is also true. In the movie "solyent green", people are misguided in the production process of solyent green. The real production process of solyent green is a secret, and the authorities want to keep it that way. Therefore, they fabricated a story about how it is made. Because civilians do not have access to information that they can trust more and that tells otherwise, they have to believe the story of the authorities.

In a third way, this is also just about privacy and memory efficiency. It may be inconvenient for particular parties to share the complete contextualization of their reasons while they still want to indicate that a particular reason has a particular amount of weight for them. Memory wise and description wise, it may just be infeasible and/or unnecessary to document the full contextualization of the reason. Therefore, a cutoff point of the description of the contextualized reason is reasonable.

C.2 Buying a television

Let us have a practical example of how balancing can work. Today, there are many options for buying a television. And when buying a television, many elements can be relevant to make a choice. However, for many people, they will first determine which one they can buy and which one they cannot. Once this is determined, they will somehow choose which one they would like to purchase. Also, it is possible that the store clerk attempts to push the boundaries on the budget, but let us try to leave this dynamic out of scope.

Properties of a TV:

- Price
- Screen Size
- Energy label
- Resolution
- Picture Quality
- Intended use
- Refresh rate

- Sound quality
- SmartTV

Consider a consumer who decides which television to purchase. The options are:

- o_1: Buy a high-end TV (*price €1,500, energy label A*).
- o_2: Buy a mid-range TV (*price €800, energy label B*).
- o_3: Buy a budget TV (*price €400, energy label C*).

Relevant grounds include:

- g_1: Lower price is preferable.
- g_2: Higher energy efficiency is preferable.
- g_3: Larger screen size improves viewing experience.

Reasons are constructed by linking grounds to options:

$$r_1 = (g_1, o_3, +),\ r_2 = (g_2, o_1, +),\ r_3 = (g_3, o_1, +),\ r_4 = (g_1, o_1, -).$$

Weights are assigned based on importance:

$$w(r_1) = 15,\ w(r_2) = 20,\ w(r_3) = 10,\ w(r_4) = 15.$$

Applying the additive dual-scale operator:

$$\text{For } o_1 \text{ vs } o_3 :\ JW_{o_1} = 30,\ RW_{o_3} = 15 \Rightarrow o_1 \text{ permitted.}$$

$$\text{For } o_3 \text{ vs } o_1 :\ JW_{o_3} = 15,\ RW_{o_1} = 15 \Rightarrow tie(0).$$

Dynamic scale evaluation:

$$o_1 : +,\ o_2 : 0,\ o_3 : +.$$

Interpretation: Both o_1 and o_3 are permissible, but o_1 is strongly justified by energy efficiency and screen size. The decision-maker may choose o_1 if sustainability is prioritized.

C.3 Drug control in harbor

This example attempts to push the boundary on what is possible with smart contracts and balancing in the way of the IHiBO framework. Today there is drug traffic that comes through harbors, can IHiBO be a tool to solve this problem. The way that IHiBO may solve this problem is that it is able to allow options in particular circumstances and make options forbidden in those circumstances. IHiBO can do this 24-seven without sharing the information required beyond those parties involved in the circumstance. This is important for this problem as multiple parties can be involved i.e. criminals, dockworkers, police; and all need to be able to accept the terms of IHiBO in order for IHiBO to take part. The problem setting is as follows:

Criminals send a container to the harbor by ship from another harbor. They have hidden the drugs in items inside the container. Once the container arrives in the harbor, only the dock workers and the police have access to the container. However, dockworkers can access containers more easily than police because they will first need to ask for official permission. The criminals' strategy is to make the dockworkers deliver the container to them, and the dockworkers are inclined to do this because the criminals exert manipulating forces on them.

How can IHiBO be used to solve this problem?

IHiBO can take away power from the dockworkers; this may reduce the incentive to use the dockworkers to access the container, since they may not be able to. The idea is that IHiBO controls access to the container. IHiBO needs to be presented with a particular input such that the input justifies IHiBO to give permission for the container to be accessed. If the input is not sufficient to allow access, then the agent will not be allowed to access the container.

This seems nice; however, in practice, it is not very obvious how to implement this.

One can go a step further by also starting to track the container once it is checked into a ship at a departing harbor. This will create a similar difficulty for sending criminals to send their goods by shipment. In addition, this will ensure that it is more difficult to manipulate the container during shipment.

Notice that one still needs to solve what the content of IHiBO should be to solve this problem. Also, what inputs can be given, what inputs dockworkers can give, and what the options are. Maybe time can be an important part, i.e. that once a contract starts there are windows before the contract, and thereby the container locks for a particular amount of time. It may be that this "solution" will slow down the flow of containers through a harbor, but maybe this cost is worth the prize, i.e., better control over drug traffic.

Scenario: A port authority must decide whether to release a container for delivery. The options are as follows.

- o_1: release the container immediately.
- o_2: Hold container for inspection.
- o_3: Deny release and escalate to law enforcement.

Relevant grounds:

- g_1: Container flagged by risk analysis.
- g_2: Delivery delay incurs financial penalties.
- g_3: Legal compliance requires inspection if risk threshold exceeded.

Reasons:

$$r_1 = (g_1, o_3, +),\ r_2 = (g_3, o_2, +),\ r_3 = (g_2, o_1, +),\ r_4 = (g_1, o_1, -).$$

Weights:

$$w(r_1) = 30,\ w(r_2) = 25,\ w(r_3) = 15,\ w(r_4) = 30.$$

Pairwise detachment:

$$o_1 \text{ vs } o_2 :\ JW_{o_1} = 15,\ RW_{o_2} = 25 \Rightarrow o_1 \text{ impermissible.}$$

$$o_2 \text{ vs } o_3 :\ JW_{o_2} = 25,\ RW_{o_3} = 30 \Rightarrow o_2 \text{ impermissible.}$$

Dynamic scale:

$$o_1 : -, o_2 : -, o_3 : +.$$

Interpretation: Escalation (o_3) is the only permissible option given strong security and compliance reasons. This outcome is documented on-chain for auditability.

C.4 The landmark SCHUFA case (case C-634/21)

Recently, the SCHUFA case took place [1]. This case is considered the first judgment regarding an automated decision-making case. SCHUFA AG holding is a german credit reference agency. When an event is brought forward in which a decision about someones credit worthiness was solely determined by an automated system. However, a human should be in the loop.

IHiBO may become more relevant as it can document not only the fund management decision but also other decisions. Since IHiBO can also register the address of the voter, it can be a document if the source of the vote is a human or a machine. This may then serve as a basis for proving whether something was determined with or without human input. In other words, if there was human and machine input, to what extent each agent influenced the outcome.

C.5 Homeless people with homes

This idea is very far away, but it may be possible to use IHiBO to manage the housing of people without homes. Of course, particular houses maybe flats are needed for this, but IHiBO may track the with other blockchain technology them checking into a flat and them leaving. While also doing this for other homeless people. The apartments reserved for them, but still need to be shared among them.

The opposite also exists and is even fairly normal. That is, people who borrow money from a bank to buy a house they cannot afford. These people own homes, although in some sense they do not own the house. This is because the bank trusts that it is a good investment for the bank. This is based on the fact that a house is not a very volatile investment. Furthermore, they will check if the person they lend the money to is in a good position to pay the depth. The bank trusts that they will not make

[1] https://www.tlt.com/insights-and-events/insight/SCHUFA case-cjeu-rules-on-scope-of-article-22/

a loss on the investment due to the circumstances. In addition, it may be that it is simply better than other investments.

C.6 Regulating Trilogues

Trilogues are a process that the EU uses to form legislation. However, one may argue that it is too powerful and undemocratic. If one wants to regulate these trilogues without diminishing the potential of it forming legislation, the approach of documenting the process in a particular way may be applied. If the outsiders trust the process set out in advance, then if the participants follow the procedure, they follow the bounds set by the outside system while still being able to have freedom to consider ideas freely.

C.7 Solving a Sudoku

Let us consider the case of solving a sudoku using the balance based detachment function. A sudoku is a game in which a grid of 9 squares needs to be filled with the numbers 1 to 9. Each number occurs 9 times, once per column, once per row, and once per square of 9 squares. Some numbers are given and cannot be changed. In figure C.1 an "easy" sudoku is depicted.

	7		5	8	9		2	
	5	9	2			3		
3	4				6	5		7
7	9	5				6	3	2
		3	6	9	7	1		
6	8				2	7		
9	1	4	8	3	5		7	6
	3		7		1	4	9	5
5	6	7	4	2	9		1	3

Table C.1: A sudoku

The three constraints introduced above create implications for the weight function. To use balance based detachment first options need to be determined. Although an option in the end is a full potential assignment of numbers to each of the empty squares, this is somewhat difficult for this use case.

Instead, an option is to put a particular number in an empty square. Denote each square with (x, y) where the bottom left square is $(1, 1)$; then $(1, 1) = 5$ and $(9, 1) = 3$. For an empty square, all the numbers can be considered: $o_1 : (7, 1) = 1$, $o_2 : (7, 1) = 2$, $o_3 : (7, 1) = 3$, $o_4 : (7, 1) = 4$, $o_5 : (7, 1) = 5$, $o_6 : (7, 1) = 6$, $o_7 : (7, 1) = 7$, $o_8 : (7, 1) = 8$, $o_9 : (7, 1) = 9$. For each of the two options, a pairwise comparison will be made; however, in this case it will not matter. The option o_8 is the only option allowed of these 9, due to row uniqueness; i.e., $(7, 1) = 8$. The other options are not permitted, how can this be expressed in balance based detachment?

The ground of row uniqueness makes each alternative to o_8 impermissible, and thus o_8 required or obligatory. Therefore, the weights should be $w((row - uniqueness, o_8)) = (0, 1)$. The justifying weight is unnecessary here, though maybe sudoku's can be solved under single proportion weights. Then $D * (o_8) = +$. The other options do not gain weight, since there are no grounds that are reasons to choose those options. One may also write $('(1, 1) = 5', o_5, -)$; i.e., that the ground $'(1, 1) = 5'$ serves as a reason against o_5. Thereby, adding requiring weight for alternatives. So $D * (o_5) = -$ as for o_i for i is $1, 2, 3, 4, 5, 6, 7, 9$ but not o_8 since $D * (o_8) = +$.

This small example shows that it is possible to select an option in this use-case using balance based detachment. Also, it shows a way in which conclusions on the dynamic level re-enter the later decision making. This is shown by selecting o_8 leads to $(7, 1) = 8$, which places additional constraints on the other squares. So far the context has not been used, but it may be used to point out the row, column, or block that the square is part of; or maybe all at three at the same time. Then, this depicts the elements that are contextually relevant and support the information for the reasons.

After selecting o_8 the context c that is represented by the sudoku in table C.1 can be changed to context c'. The context c' is the context c extended with move o_8.

Solving sudoku's is relevant as a benchmark as many other automatic systems have been tested to solve sudoku's. For example, logic programming approaches have solved sudoku's. Logic programming has been shown to be able to express abstract argumentation, therefore this could be another avenue to relate abstract argumentation to balance based detachment.

C.8 Ethics

The balancing of normative reasoning is a framework of ethics Tucker, 2025. The theory is not saying what is ethical, but shows how one should balance reasons correctly. By determining the weight of reasons, one thus determines what is ethical and what is not. The weights lead to statements regarding whether particular actions are permitted or not in given contexts.

By implementing the theory in IHiBO with respect to this theory, the idea of combining weight functions is added. This requires that there can exist multiple weight functions. Where each weight function belongs to an agent. Previously, it made sense to think that there can exist only one correct weight function. That takes an objective perspective. However, when agency is added, different opinions and perspectives are added to the fold.

Not only are there now different weight functions. Agency also contains the idea of multiple perspectives and access to different information. It is not so clear how multi agent systems combined with balancing relates to ethics. Is this inter-subjective view on what is permitted more ethical than a theory that aims at the objectively true weight function?

For one thing, these theories are not the same. By introducing the voting used in IHiBO to balancing things become possible that should not be possible in the theory without voting. That is, the agency that voting implements introduces the idea that different agents can model and perceive a situation differently. Therefore, the joint case after the vote may contain elements that contradict each other even though each individual agent has a consistent perspective on the situation. From a logic perspective, this looks like a bad basis.

Another point is that combining the weight functions is not obvious and has problems. The main problem is that the coherence of each individual weight function makes sense in its own right. However, when weight functions are combined, the attribution of weights may be done in such a different way that there is not a clearly correct way to combine the systems. This becomes especially clear when an agent considers only one reason in a circumstance. Then one can ask, is this a heavy or light reason? However, since it is the only reason the exact weight does not matter much as long as they are non-zero when they are intended to be non-zero. The effect of "5" justifying weight in this case is the same as "1" justifying weight. The difference between these values becomes relevant once there are more reasons.

The main concept that influences this process is the documentation of IHiBO. This creates an incentive to use weights that can be defended. If one uses a weight too large or too small, this will be recorded. Then if a “mistake” follows, there will be documentation that may indicate a responsibility for the mistake due to the chosen weights.

Also, the voting introduces the possibility for options to be collectively permitted while each individual would not permit the option. This idea does not seem very ethical. However, if all agents agree to the process and trust the process and believe that the process is correct. Then maybe this outcome is strangely enough the most ethical.

C.9 Problem of control in artificial intelligence

The relation that IHiBO aims to improve is the trust relationship between fiduciary and principal. The fiduciary has the duty to act in the best interest of the principal. In the fund management context, the fiduciary or, in fact, the fiduciaries are the fund or portfolio managers. Investors are the principal, as managers act in their best interest.

This trust problem between a trustor and a trustee is also considered in the Artificial Intelligence literature and is known as the principal problem Nick, 2014, ch. 9. However, people tend to refer to it as the control problem in the context of AI. The problem has two variants, but both are about a principal having to trust another agent that acts on their behalf in their best interest.

The book by Nick, 2014 touches on two versions. The first is the general case; in this case, a human trusts an agent. This agent tends to be a person, though it could also be a software agent. However, in the second, the focus is put specifically on the case in which the human has to trust a super-intelligent system.

Often the example of a paperclip making machine is brought forward. First, the machine is simple and only performs the work of a few factory workers. But as upgrades are made, less factory workers are required. At some point, the company upgrades the paperclip making machine with Artificial Intelligence. Eventually, the AI in the machine is intelligent enough to upgrade itself better than humans and becomes a universe devouring machine that reforms everything into a paperclip.

By not giving direct autonomy to the machine but first funneling it through something like IHiBO that documents the reasons and what is permitted, a measure of control can be implemented. Human managers can even limit the options that can be chosen to those they think are relevant.

C.10 Game Theory perspective

When talking about choosing options in a multi agent context one should think of Game Theory. Game theory studies the actions of agents in a shared circumstance. Given a circumstance, a single agent or any finite number of agents may have to choose an action. –One can also study the case of infinite agents.– Game theory tells us that each agent will be considering a particular set of possible options, normally in game theory called strategies. And each agent has a utility function, sometimes referred to as heuristic, that tells the agent how good the outcomes of that option are. Since there is a multi agent context, this utility function at least to some degree will take the other agents possible actions into account. If one takes the perspective of a given agent, it may consider a set of possible actions of another agent as an equivalence class among which it can not differentiate the options and so will attribute the same utility. In general game theory takes a more objective perspective by analyzing all the subjective utilities at once. In this domain John Nash made significant contributions among others by introducing the nash equilibrium. A nash equilibrium is a collective strategy such that each agent would not change its strategy since the other agents do not change their strategy Nash Jr, 1950. In other words a Nash Equilibrium is a state where no player can gain an advantage by changing their strategy alone, assuming that other players' strategies remain constant.

The prisoners dilemma is a common example as an introduction to game theory. The problem setting is that two people collaborated in committing a crime. Then after the crime they were caught and put in separate interrogation rooms while they both knew that the other was caught. They each have the option to confess and walk free if the other does not confess. However, if they both confess, then the sentence is reduced. If they both lie, then they get a small sentence. For both agents, the best outcome is if they confess while the other lies. See the table C.2 below for the payoffs.

The entry $(-7, -7)$ when they both confess is a nash equilibrium because if either would change to lie while the other would not change, then instead of receiving -7 utility, they receive -10 utility.

	Confess	Lie
Confess	(-7,-7)	(0,-10)
Lie	(-10,0)	(-1,-1)

Table C.2: Payoff matrix of Prisoners Dilemma
(utility (x,y), x for agent 1, y for agent 2)

The point here was to give an intuition of how game theory looks at these cases. To bring this back to multi agent detachment as in IHiBO is to relate it to the joint action profile. IHiBO at the moment allows agents to introduce actions called options. In the context of portfolio management, this makes sense, as managers are managing the same portfolio. However, when managers are more like agents, as in game theory, they may choose their own actions. The smart contract of IHiBO may then be a place to find a joint action profile. However, this will change some things considerably. That is, IHiBO will need to consider joint action profiles as possible output options. This requires additional logic on the options to determine the full table and each agents position to each joint profile. The smart contract then is able to detach which joint action profiles are permitted by the collective without sharing the weights of the individual agents.

The individual weights of the agents are also different then in game theory. In game theory the problem is analyzed from the perspective of a utility function, while in dual scale ground the reasons for the options are given weight. Thereby making option only permitted through the weight they have. It is tempting to identify utility as weight, and this is not a bad idea. However, in dual scale there are two weights and the weight counts towards permissibility where utility directly says that this action gives better utility than another action.