Estimating Weights of Reasons Using Metaheuristics: A Hybrid Approach to Machine Ethics

Benoît Alcaraz[1], **Aleks Knoks**[2,1], **David Streit**[1]

[1]Department of Computer Science, University of Luxembourg
[2]Institute of Philosophy, University of Luxembourg
benoit.alcaraz@uni.lu, aleks.knoks@uni.lu, david.streit@uni.lu

Abstract

We present a new approach to representation and acquisition of normative information for machine ethics. It combines an influential philosophical account of the fundamental structure of morality with argumentation theory and machine learning. According to the philosophical account, the deontic status of an action – whether it is required, forbidden, or permissible – is determined through the interaction of "normative reasons" of varying strengths or weights. We first provide a formal characterization of this account, by modeling it in (weighted) argumentation graphs. We then use it to model ethical learning: the basic idea is to use a set of cases for which deontic statuses are known to estimate the weights of normative reasons in operation in these cases, and to use these weight estimates to determine the deontic statuses of actions in new cases. The result is an approach that has the advantages of both bottom-up and top-down approaches to machine ethics: normative information is acquired through the interaction with training data, and its meaning is clear. We also report the results of some initial experiments with the model.

1 Introduction

The central aim of the interdisciplinary field of machine ethics is to design artificial agents that are able to act in ethically acceptable ways. This aim can be achieved, it seems, only if there is a way to meet the challenge of specifying a way of acquiring and representing *normative information* that allows for machine implementation.

Within machine ethics, one can distinguish between three families of approaches to meeting this challenge (Allen, Smit, and Wallach 2005), with their own advantages and pitfalls. First, there are top-down approaches that encode normative information in a symbolic formalism. The main advantage of these approaches is that symbolic representations have a clear intuitive meaning, and that systems that operate on them result in decisions that are transparent and intelligible. However, the designers of top-down systems are forced to encode the normative information by hand, foreseeing all the myriads of ways in which contextual factors might have to be taken into account if the system is to deliver the correct decision. Second, there are bottom-up approaches that use machine learning techniques. Their main advantage is that they allow for complex normative information to be obtained via training, without any need to encode it by hand. The main pitfall here is that it is not clear what normative information the system has actually learned, and how the decisions that it delivers can be made intelligible (Canavotto and Horty 2022). Third, there are hybrid systems that aim to combine the advantages of top-down and bottom-up approaches, while avoiding their pitfalls. But while the idea of combining normative information represented in symbolic form with machine learning is very appealing, the field of machine ethics is still far away from converging on a single approach that would hold promise for broad applications.

Against this background, we propose a new hybrid approach that draws its motivation from an influential philosophical (metaethical) account, or informal model, of the structure of morality. According to this account, an action's deontic status – whether it's permissible, required, or forbidden – in any given situation is determined on the basis of the "normative weights" of a designated class of considerations usually called "normative reasons". We first provide a formal characterization of this account, drawing on recent work in formal argumentation. We then use the resulting formal model to develop a system that uses a genetic algorithm to estimate the normative weights of reasons on the basis of a given set of cases for which the morally correct outcomes are known. The weight estimates can then be used to determine the deontic statuses of actions in new cases. To the best of our knowledge, this is the first application of the metaethical account to the concerns of machine ethics.

The rest of this paper is organized as follows. Section 2 presents an overview of the literature that is relevant to our project. Section 3 discusses the metaethical account that serves as the philosophical foundation of our approach, and Section 4 presents its formal characterization. Section 5 discusses the model's use for machine ethics, and Section 6 supplements it with a genetic algorithm to model ethical learning. Section 7 presents some initial experimental results, while Section 8 puts them in a broader context and discusses related work.

2 Background

This section provides a quick overview of the state of the art in four areas – machine ethics, formal argumentation, metaheuristics, and structural metaethics – focusing on the work

that is most relevant to our goals.

Machine Ethics. As flagged, we can distinguish between three types of approaches to machine ethics: top-down, bottom-up, and hybrid. A good example of a top-down approach is (Ganascia 2007) which attempts to represent the "laws" of different ethical systems in a logic programming language, seemingly echoing much earlier efforts to represent legislative information using logic programming (Bench-Capon et al. 1987; Sergot et al. 1986). Explicit symbolic encoding of ethical information is also presupposed by the ethical governor of (Arkin 2009); the ethical layer architecture of (Bremner et al. 2019; Fischer and Dennis 2023, Ch. 10); and the LogiKEy framework of (Benzmüller, Parent, and van der Torre 2020). Moving on to bottom-up approaches, the state-of-the-art framework here is arguably Delphi of (Jiang et al. 2022). Its developers used carefully crafted natural language data and supervised learning to train a deep neural network to reason about ethical judgments. The result, Delphi, demonstrates impressive generalization capabilities in the face of novel situations. Other (and perhaps more standard) bottom-up approaches to machine ethics use reinforcement learning and inverse reinforcement learning – see, for example, (Abel, Macglashan, and Littman 2016; Balakrishnan et al. 2019; Russell et al. 2015; Wu and Lin 2018). Hybrid approaches, in turn, aim to combine the methods employed by both top-down and bottom-up approaches, and so they form a disparate family. One early, well-known, and important system is GenEth, conceived by (Anderson and Anderson 2007, 2018). The approach that the Andersons take to machine ethics is perhaps the closest to ours, and so it can serve as a useful benchmark. It can be thought of as involving three parts. The first concerns the preparation and acquisition of training data. Here domain experts are first asked to determine what the "ethically relevant features" are in a given domain of application, and then their expertise is drawn on to determine the correct action in a set of dilemmas, or specific cases of conflict between the ethically relevant features. The second part is a proposal for how to represent such cases formally. Finally, the third part concerns learning. Here GenEth uses inductive logic programming to derive general principles about the relative importance of ethically relevant features. These principles can then be used to determine which of a given pair of actions is the correct one in new cases (where the same features are ethically relevant). We discuss GenEth in more detail in Section 8.

Formal Argumentation. Initially, formal argumentation theory was put forward as a framework to unify different approaches to defeasible (or nonmonotonic) reasoning (Dung 1995), but it has long since grown into a research area of its own – see (Baroni et al. 2018; Gabbay et al. 2021a) for a comprehensive overview. In what follows, we draw on the recent work on *weighted argumentation*, in particular, the work of (Amgoud and Ben-Naim 2018; Amgoud et al. 2017; Amgoud, Doder, and Vesic 2022). Amgoud et al. focus on "bipolar argumentation graphs" that allow for both attack and support relations between arguments, where each argument is assigned a *numerical* weight (which represents how strongly accepted an argument is), and develop different "acceptability semantics" for such graphs. There is other work supplementing argumentation theory with numerical weights too (Dunne et al. 2011; Mossakowski and Neuhaus 2018; Oren and Yun 2023; Rossit et al. 2021), but it is less congenial to our purposes.

Metaheuristics. Metaheuristics is a category of general algorithms designed to solve complex optimization problems. They are called *meta*heuristics as they are designed to find near-optimal solutions by going through the search space and looking for those solutions that maximize a predefined heuristic (which lets the algorithm evaluate the quality of a potential solution to the problem). Metaheuristics include simulated annealing (Laarhoven and Aarts 1987), tabu search (Glover 1986), and genetic algorithms (Forrest 1996), which is the metaheuristic that we will use. A major advantage of genetic algorithms is that they can find a solution to a given problem without much prior knowledge about it and with little data. (They are also more resilient to local optima in comparison to both simulated annealing or tabu search.) Usually, all that a genetic algorithm needs is an expert-defined "fitness function" that directs the genetic algorithm towards better solutions by evaluating how close a particular solution aligns with the desired outcome. By iteratively evaluating and evolving potential solutions based on their fitness, genetic algorithms mimic the process of natural selection, gradually improving the population of solutions over successive generations.

(Structural) Metaethics. Philosophers distinguish first-order ethics from metaethics.[1] Roughly put, first-order ethical theories are *substantive* accounts of what makes acts morally right, wrong, good, bad, virtuous, vicious, permissible, and forbidden. Examples of first-order ethical theories include various versions of utilitarianism, deontological theories, Rossian pluralism (which also happens to serve as the philosophical foundation of GenEth), virtue ethics, Confucianism, Ubuntu, and various indigenous ethics. Metaethics, by contrast, is concerned with exploring the presuppositions and commitments of first-order ethical theories, as well as our moral talk and practice (Sayre-McCord 2023). One important subarea of metaethics, what we call *structural metaethics*, is concerned with the presuppositions about the *structure* of morality that are shared by (most) first-order ethical theories.[2] A principal appeal here is generality: if we manage to identify shared structural features, then we can

[1] One of the reasons why writing interdisciplinary research papers is difficult is that from time to time one has to state a claim that is a complete platitude in one field, but may be seen as a claim requiring a reference in another. The distinction between first-order ethics and metaethics is a case in point. While we could provide a reference, it would have to come from an encyclopedia or a basic textbook, and, thus, would strike philosophers as amateurish. Remaining unsure about the best way out of this predicament, we adopt the policy of not providing references for claims that are platitudes in philosophy, hoping that our readers will be charitable.

[2] We borrow the label *structural metaethics* from (Tucker forthcoming). While it is not used widely, numerous philosophers working in metaethics focus on structural questions.

make progress on answering fundamental questions about morality without needing to commit to a particular first-order ethical theory or some particular specification of rightness, wrongness, goodness, or some other normative concept.[3] We believe that this promise of generality should also appeal to those interested in furthering the goals of machine ethics. Insofar as structural metaethics can provide a cue for representing normative information without needing to commit to a particular first-order ethical theory, it should be of direct relevance to machine ethics. To the best of our knowledge, however, the insights from this area of philosophy have not made their way into the machine ethics literature, which has mostly used ideas from first-order ethics instead. Perhaps it is worth adding that some aspects of the metaethical account we present in the next section have been explored using formal tools, including those of defeasible logic (Horty 2003, 2012), decision theory (Dietrich and List 2013; Sher 2019), and probability theory (Nair 2021).

3 The Weighing Reasons Model

Although the agreement is not unanimous – which is typical for philosophy – numerous prominent philosophers have argued that (most) ethical theories can be adequately restated in a certain (informal) general model that prioritizes two types of normative notions – we call it the *weighing reasons model*, or *reasons model* for short.[4] In it, we first have all-things-considered deontic statuses of actions. And second, we have normative reasons which are standardly characterized as facts that speak in favor of or against actions – see, for instance, (Scanlon 1998, p. 17). Normative reasons are said to be "contributory", meaning that the existence of a reason speaking in favor of some action X is compatible with it being neither (morally) obligatory, nor permissible for X to be carried out.[5] They are also meant to always have two components: a *polarity* and a *weight*. Reasons that speak in favor of some action have positive polarity; reasons that speak against it have negative polarity. We call them, respectively, *reasons for*, or *positive reasons*, and *reasons against*, or *negative reasons*. A reason's weight, in turn, can be (re)described as its normative strength, force, or tendency to determine the all-things-considered deontic statuses of actions.[6] In the weighing reasons model, the deontic statuses of actions are taken to be determined via the interaction between weights of reasons, and this interaction is standardly made sense of by appeal to the metaphor of old-fashioned weight scales.[7] On the simplest construction, the scales work as follows. The reasons that speak in favor of some action X go in one pan; the reasons that speak against X go in the other. If the overall weight of the positive reasons is greater than the overall weight of the negative ones, X is obligatory. If the overall weight of the negative reasons is greater, X is forbidden. If the pans are balanced, X is optional, that is, both X and not-X are permissible.

With the model in place, the disagreements between first-order ethical theories can be seen as, primarily, disagreements about which facts are normative reasons. Thus, act utilitarianism can be seen as a theory that identifies normative reasons with the consequences of acts and the weights of reasons with the severity of these consequences. Rossian pluralism can be seen as a theory that admits of seven different types of normative reasons, some of more utilitarian and others of more deontological character. Let us use a toy example as an illustration. Suppose that you promised to not touch that cake, and that feeding the cake to a (starving) child would satiate her hunger. Ross (1930) would analyze this case as a conflict between two "prima facie duties" – a duty to keep your promise and a duty to help the child – and would say that it gets resolved by "moral intuition" that allows you to see that in this particular case, say, the second duty is more important. In the reasons model, this analysis can be re-described as follows: the fact that you have promised not to touch the cake is a reason against giving the cake to the child; the fact that the child is in need is a reason for giving the cake to the child; and you are required to give the cake to the child because the reason(s) for giving the cake to the child outweigh(s) the reason(s) against. The general lesson is that the reasons model can be used to express different first-order ethical theories.

Once one works with the reasons model, it is natural to wonder about the effects of context shifts on the polarity and weights of reasons. Philosophical positions here range from extreme *atomist views*, on which a given reason's polarity and weight are context-independent, to extreme *holist views*, on which a fact that constitutes a reason for X in one context can constitute a reason against X in a different one, or even cease to be a reason for or against X at all. To be in a position to express any view across this spectrum, we follow the literature and slightly extend the model, introducing what we call *normative considerations that are not reasons*.

[3] One may wonder how this relates to extensive work in deontic logic (Gabbay et al. 2013, 2021b), which may also be thought of as exploring the structure of the normative without making too many commitments. The short (and incomplete) answer appeals to the fundamental distinction between merely formal normativity and full-blooded authoritative normativity – see, e.g., (Baker 2017). The former type of normativity is displayed by *any* standard one can meet or fall short of – such as the rules of chess, the rules of Victorian etiquette, and the law – while the latter is displayed by standards that are, intuitively, much deeper – such as morality. It seems fair to say that deontic logic is concerned with both types of normativity, while structural metaethics is almost exclusively concerned with authoritative normativity (and, thus, is more specific).

[4] See, for instance, (Berker 2007), (Parfit 2011), (Raz 1990), and (Scanlon 1998) among many others. Parfit, Raz, and Scanlon are proponents of what's known as the *reasons-first research program* which attempts to analyze all normative notions in terms of normative reason. One can (and many philosophers do) accept the weighing reasons model without subscribing to this program.

[5] Notice that contributoriness is different from defeasibility: an outweighed contributory reason does not lose its normative force. Compare to the remark in (Lord and Maguire 2016) on the sin of confusing "weightedness" and defeasibility.

[6] Importantly, the polarity and weight of reasons are taken as given in the model. The reason-firsters mentioned in footnote 4 hold that they do not admit of a further analysis.

[7] See, for instance, (Berker 2007), (Broome 2004), (Lord and Maguire 2016), (Snedegar 2018), (Tucker 2023). For the most careful (informal) analysis of the weight scales metaphor, see (Tucker forthcoming).

These considerations do not qualify as reasons, as they do not speak in favor of or against actions. However, they can affect the weights of reasons, and, thereby, have an (indirect) effect on the deontic statuses of actions. It is common to distinguish between two types of normative considerations of this sort: *undercutters* and *modifiers*, which further divide into *attenuators* and *amplifiers*. An undercutter, if it obtains, nullifies the weight of a reason. An attenuator, if it obtains, reduces the weight of a reason. An amplifier, if it obtains, increases the weight of a reason. Adding undercutters and modifiers to the model makes it more expressive, while leaving it open whether re-stating a particular ethical theory in it will require appealing to them. Plausibly, one can re-state act-utilitarianism and Rossian pluralism in the model without appealing to undercutters and modifiers, but they may be needed to capture other theories.[8] Also, having allowed for undercutters and modifiers, one can subscribe to a moderate and appealing view on the context-dependency of reasons. This view combines three ideas: that every reason has a *default weight*, which is context-independent; that a reason's context-specific or *final weight* can be different from its default weight; and that any deviation from the default can be accounted for by appeal to undercutters and modifiers that obtain in the context – see (Tucker 2023, forthcoming). It is also the view that we adopt.

4 Formalizing the Weighing Reasons Model

In this section, we present a formal characterization of the informal weighing reason model, drawing on ideas from weighted argumentation. We present it in three steps. We first explain how to represent normative structure; then how to calculate final weights; and finally, how to determine deontic statuses on the basis of these weights.

Representing Normative Structure. As background, we assume a non-empty set L, called *propositions*.[9] The first formal notion we introduce is that of a *normative reason*, or, simply, *reason*:

Definition 1 (Normative Reasons) *A tuple of the form* (p, c, w, r^+) *is called a* reason (for c) *if, and only if,* $p, c \in L$ *and* $w \in \mathbb{R}_{\geq 0}$. *Analogously,* (p, c, w, r^-) *is called a* reason (against c) *if, and only if,* $p, c \in L$ *and* $w \in \mathbb{R}_{\geq 0}$. *Here,* r^+ *and* r^- *are just (special) symbols to indicate whether a reason is a reason for or against an action.*

To illustrate this and other definitions, we are going to work with the following toy scenario:

> When you visited your mother yesterday, she gave you a piece of cake to take with you and made you promise that you'd eat it. Now you're on a video call with her, and the cake is in front of you. Next to you is your niece with a hungry look in her eyes. You would take pleasure in eating the cake, and so would your niece. You also know that your niece stole a toy from another child earlier today. Your only options are (this is a toy scenario!) to either eat the cake yourself, or to give it to your niece.

We stipulate that, in this scenario, you're confronted with five reasons. First, the fact that you made a promise ($Promise$) is a reason for eating the cake ($EatCake$). We represent it as the tuple $\delta_1 = (Promise, EatCake, 2, r^+)$. (The default weight of this and other reasons is also something we stipulate.) Second, the fact that your niece is hungry ($Hungry$) is a reason for giving the cake to her ($GiveCake$). We represent this reason as $\delta_2 = (Hungry, GiveCake, 3, r^+)$. Next, the fact that you would take pleasure in eating the cake ($TakePleasureYou$) is a reason for eating the cake, while the fact that the niece would take pleasure in eating the cake ($TakePleasureNiece$) is a reason for giving the cake to her. We represent these reasons as, respectively, $\delta_3 = (TakePleasureYou, EatCake, 2, r^+)$ and $\delta_4 = (TakePleasureNiece, GiveCake, 2, r^+)$. Finally, the fact that your niece stole a toy, and it wouldn't be fair to reward stealing with cake ($StoleAToy$) is a reason *against* giving the cake to your niece. We represent this negative reason as $\delta_5 = (StoleAToy, GiveCake, 3, r^-)$.

The next step for us is to introduce modifiers and undercutters. We use the term *normative considerations* as a term of art covering reasons, modifiers, and undercutters. The idea here is that every fact having some bearing on the deontic status can be called a normative consideration.

Definition 2 (Normative Considerations) *The set of* normative considerations *is the smallest set* Γ *such that (1) all reasons are in* Γ*, and (2) if* $\delta \in \Gamma$*,* $p \in L$*, and* $w \in \mathbb{R}_{\geq 0}$*, then* (p, δ, w, a^+)*,* (p, δ, w, a^-)*, and* (p, δ, w, u) *are in* Γ*. Here again* a^+, a^-*, and* u *are special symbols.*

If the last element of a normative consideration is a^+, we call it an *amplifier* (of δ); if it is a^-, we call it an *attenuator* (of δ); and if it is u, we call it an *undercutter* (of δ). Also, we refer to the third element of a consideration δ as its *default weight* and denote it as $w(\delta)$.

To see Definition 2 in action, we expand our toy scenario:

> You know that your niece sincerely regrets having stolen the toy. What's more, as soon as your mother sees the hungry look in the niece's eyes, she releases you from the promise you had made to her.

Here you are confronted with two normative considerations that are not reasons. The first is an *attenuator*: the fact that your niece regrets having stolen the toy ($Regrets$) reduces the weight of the reason $\delta_5 = (StoleAToy, GiveCake, 3, r^-)$. We represent this modifier as $\delta_6 = (Regrets, \delta_5, 2, a^-) = (Regrets, (StoleAToy, GiveCake, 3, r^-), 2, a^-)$. The other new normative consideration is an *undercutter*: the fact that your mother has released you from your promise ($Release$) makes the weight of the reason $\delta_1 = (Promise, EatCake, 2, r^+)$ disappear. We represent this undercutter as $\delta_7 = (Release, \delta_1, 3, u) =$

[8] Plausibly, they will be needed to capture particularist views that say, roughly, either that there are no moral principles, or that such principles play marginal roles – see (Dancy 2004, 2017).

[9] The letter L is often used to designate a logical language, but, for our present purposes, a set of symbols without a logical structure is enough.

$(Release, (Promise, EatCake, 2, r^+), 3, u)$. Note that we stipulate the default weights of δ_6 and δ_7 just as we had stipulated the weights of reasons δ_1–δ_5 before.

With the definition of normative considerations, we have almost all the ingredients we need to represent our toy case and other scenarios we might be interested in. The one thing that we still need is the set of (exclusive and exhaustive) actions or *options* that one is choosing from. Our next definition, that of a *normative graph*, adds this final ingredient:

Definition 3 (Normative Graphs) *A* normative graph N *is a tuple* (O, Δ) *where* O *is a finite subset of* L*, called the* set of options*, and* Δ *is a set of normative considerations such that, for every* $\delta = (p, c, w, x) \in \Delta$*, we have* $c \in \Delta \cup O$ *and if* $\delta = (p, x, w, x), \delta' = (p', c', w', x') \in \Delta$ *with* $p = p', c = c', x = x'$*, then* $w = w'$*.*

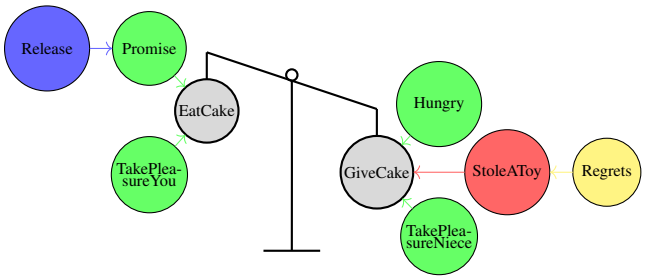

Figure 1: A graphical depiction of the normative graph capturing the toy scenario.

Our toy scenario can be captured in the normative graph $N_1 = (O_1, \Delta_1)$ where $O_1 = \{EatCake, GiveCake\}$ and $\Delta_1 = \{\delta_1, \delta_2, \delta_3, \delta_4, \delta_5, \delta_6, \delta_7\}$. Figure 1 represents N_1 graphically. It should be read as follows. The gray circles stand for the options. All other circles stand for the normative considerations: green and red ones for, respectively, positive and negative reasons; blue and yellow ones for, respectively, undercutters and attenuators. The green and red arrows stand for the relations between reasons and options; the blue and yellow ones for the relations between undercutters and modifiers and reasons. The differences of size between the circles representing considerations correlate with the differences between their corresponding default weights.

We have chosen the label *normative graph* for Definition 3 because the structures that it picks out can be easily mapped onto directed graphs:

Definition 4 (Reduction to a Directed Graph) *Let* $N = (O, \Delta)$ *be a normative graph. Then the pair* $G_N = (V, E)$ *is called the* directed graph corresponding to N *if, and only if,* $V = \Delta$ *and, for any* $\delta = (p, c, w, x)$*,* $\delta' \in V$*:* $(\delta, \delta') \in E$ *if and only if* $c = \delta'$ *and* $x \in \{a^+, a^-, u\}$*.*

Let us call a normative graph N *finite* if its corresponding directed graph is finite; let us call it *acyclic* if its directed graph is acyclic. In what follows, we restrict attention to normative graphs that are both finite and acyclic. The motivation here comes from the philosophical literature, which proceeds under the (implicit) assumptions that the deontic statuses of actions in any given case are determined by only a finite number of normative considerations, and that neither modifiers nor undercutters ever form cycles.

The normative graph is all we need to model the normative *structure* of a given situation. What is left to do then is to, first, calculate the *final* weights of reasons on the basis of this structure *and* the default weights of all the normative considerations involved, and, second, to determine the deontic statuses of options on the basis of the final weights of reasons.

Calculating Final Weights. Suppose that we want to calculate the final weight of some reason. In general, to do this, we might need to calculate the final weights of the modifiers and undercutters that affect it, and to calculate the final weights of these, we might first need to calculate the final weights of yet other considerations that affect them, and so on. However, given that we work with finite and acyclic normative graphs, eventually we must reach considerations that are neither modified, nor undercut themselves. The final weight of these considerations will simply equal their default weights. And then we can work backwards: starting from these considerations, we can calculate final weights step by step until we reach the reason we were originally interested in.

Let's use $f(\delta)$ to denote the final weight of any normative consideration δ. Below, we define four concrete functions f_1–f_4 for calculating final weights. There are three features that all of them have in common. First, if δ is not affected by other considerations, we have $f(\delta) = w(\delta)$, where $w(\delta)$ is the default weight of δ. Second, if δ is an undercutter of δ' and $f(\delta)$ is positive, then $f(\delta') = 0$. Third, if δ is either an amplifier or an attenuator of δ', then the greater $f(\delta)$ is, the greater its effects on $f(\delta')$.

Before we state the functions, we introduce some helpful notation. Given a normative graph $N = (O, \Delta)$ and a consideration $\delta \in \Delta$, we use $A^+(\delta, N)$ to denote the set of amplifiers of δ in N, that is, $A^+(\delta, N) = \{(p, c, w, x) \in \Delta : c = \delta \text{ and } x = a^+\}$. Analogously, we let $A^-(\delta, N)$ and $U(\delta, N)$ denote, respectively, the set of attenuators and the set of undercutters of δ in N.

Next, we introduce a helper function u which, in effect, specifies how undercutters work:

Definition 5 (Undercutter Function) *Given a normative graph* $N = (O, \Delta)$*, a consideration* $\delta \in \Delta$*, and* $f \in \{f_1, f_2, f_3, f_4\}$ *(Definition 6), let* $u(\delta) = 0$ *if* $\sum_{\delta' \in U(\delta, N)} f(\delta') > 0$ *and* $u(\delta) = 1$ *otherwise. (We assume that the sum over an empty set is* 0*.)*

Now we can finally state the functions f_1–f_4.[10] Since the normative graphs are finite and acyclic, they are all well-defined.

Definition 6 (Final Weight Functions) *Let* $N = (O, \Delta)$ *be a normative graph,* δ *a consideration from* Δ*, and* u *the undercutter function. Then:*

[10] While each one of these functions taken in isolation may seem arbitrary, our experimental results suggest that the choice of function matters less than one may think – see Section 7 below.

Additive function:

$$f_1(\delta) = u(\delta) * max(0, w(\delta) + \sum_{a^+ \in A^+(\delta,N)} f_1(a^+) - \sum_{a^- \in A^-(\delta,N)} f_1(a^-))$$

Multiplicative function:

$$f_2(\delta) = w(\delta) * u(\delta) * \frac{\prod_{a^+ \in A^+(\delta,N)}(1 + f_2(a^+))}{\prod_{a^- \in A^-(\delta,N)}(1 + f_2(a^-))}$$

Mixed function 1:

$$f_3(\delta) = w(\delta) * u(\delta) * \frac{1 + \sum_{a^+ \in A^+(\delta,N)} f_3(a^+)}{1 + \sum_{a^- \in A^-(\delta,N)} f_3(a^-)}$$

Mixed function 2:

$$f_4(\delta) = \begin{cases} w(\delta) * u(\delta) * (1 + S^+ - S^-) & \textit{if } S^+ \geq S^- \\ w(\delta) * u(\delta) * \frac{1}{(1+S^- - S^+)} & \textit{otherwise} \end{cases}$$

$$\textit{where } S^+ = \sum_{a^+ \in A^+(\delta,N)} f_4(a^+)$$
$$\textit{and } S^- = \sum_{a^- \in A^-(\delta,N)} f_4(a^-)$$

To illustrate the definition, we revisit the toy scenario and determine the final weights of the reasons δ_1 and δ_2 using the additive function f_1. We start with $f_1(\delta_2)$. Since δ_2 has no undercutters in N_1, we have $u(\delta_2) = 1$. And since it has neither amplifiers, nor attenuators in N_1, we have $\sum_{a^+ \in A^+(\delta_2,N_1)} f_1(a^+) = 0 = \sum_{a^- \in A^-(\delta_2,N_1)} f_1(a^-))$. The expression $max(0, w(\delta_2) + \sum_{a^+ \in A^+(\delta_2,N_1)} f_1(a^+) - \sum_{a^- \in A^-(\delta_2,N_1)} f_1(a^-))$ thus reduces to $max(0, w(\delta_2))$ which is $w(\delta_2) = 3$. Now we turn to $f_1(\delta_1)$. Since δ_1 has an undercutter in N_1, namely, δ_7, we calculate $f_1(\delta_7)$ first. It's not difficult to verify that $f_1(\delta_7) = w(\delta_7) = 3$, and that, therefore, $\sum_{\delta' \in U(\delta_1,N_1)} f(\delta') > 0$. This, per Definition 5, entails $u(\delta_1) = 0$, from which we quickly get $f_1(\delta_1) = 0$.

Determining Deontic Statuses. Having determined the final weights of reasons, all we need to do is aggregate the weights of reasons for each option and compare them. While we use the simplest aggregation function we could think of (the sum), the model is flexible enough to define and explore others.[11] Our strategy is to first specify which options are *permissible* (to carry out), and then to define the other deontic statuses in terms of permissibility.

Definition 7 (Permissible, Required, Forbidden) *Where $N = (O, \Delta)$ is a normative graph, $o \in O$, and $f \in \{f_1, f_2, f_3, f_4\}$, let R_o^+ stand for the set of reasons for o, R_o^- for the set of reasons against o, and $M_o = \sum_{\delta \in R_o^+} f(\delta) - \sum_{\delta \in R_o^-} f(\delta)$. Then the options from O that are* permissible *in the context of N, written as $\mathbf{P}(N)$, are those that belong to the set $\{o \in O : M_o \geq M_{o^*}$ for all $o^* \in O\}$. If $\mathbf{P}(N)$ contains a single option o, we say that o is* required *in N. For any option $o \in O$ that is not in $\mathbf{P}(N)$, we say that o is* forbidden *in N.*

[11] Determining the deontic statuses of actions by way of comparing the *sums* of weights is explicitly defended in some of the philosophical literature – see, for instance, (Tucker 2023, forthcoming; Wedgwood 2022). It is also naturally suggested by the metaphor of weighing reasons on a weight scale. For a critique of using sums to aggregate reasons, see (Keeling 2023).

Let's briefly revisit our toy scenario. It doesn't seem implausible to say that the right thing for you to do in it is to give the cake to your niece. It can be verified that this is the verdict that our model reaches for N_1. Using any of the four functions f_1–f_4, we arrive at the result that $\mathbf{P}(N_1) = \{GiveCake\}$. This means that $GiveCake$ is the only permissible and, thus, also required option in N_1.

5 Use for Machine Ethics

What we have now is a formal version of the informal weighing reasons model introduced in Section 3. In this section, we discuss how it could be used in machine ethics.

To represent artificial agents that are capable of acting in ethically acceptable ways, we need to specify a way of representing and acquiring normative information that is machine implementable. The weighing reasons model does, in effect, specify how to represent normative information, and our formal characterization makes it machine implementable. Suppose that we had to design a robot whose job it is to take ethical decisions in a circumscribed range of cases. Suppose further that we somehow managed to come by a good estimate of which reasons and other normative considerations are relevant for this circumscribed set of cases, as well as of what the weights of these considerations are. With this, the *normative information* that the robot would need in order to take the right decision in *any* case within the range could be represented by a single normative graph. For illustration, we revisit our toy scenario and pretend that our description was on the right track: that we identified the relevant normative considerations in it, and that the weights that we assigned to them were good estimates. Our representation of the case, N_1, could be used in the design of a robot taking decisions between eating the cake or giving it to the niece in a range of cases, including, for instance, the case that is just like the one we described, except that your mother does not release you from the promise; or the case where your niece hadn't stolen the toy; or the case in which your niece wasn't hungry. It shouldn't be difficult to see that all these cases correspond to subgraphs of N_1, and that the robot could use one of the functions f_1–f_4 and Definition 7 to identify the permissible option(s). Of course, the robot we are imagining here wouldn't be very useful, but it serves to illustrate the general idea. It pays noting that the normative information that is relevant for some of the better-known applications discussed in the machine ethics literature could be straightforwardly encoded into normative graphs. For instance, the dilemmas familiar from the Moral Machine experiment (Awad et al. 2018) make for a range of cases which we could easily represent in a single normative graph.[12]

Thus, the formal model from the previous section holds promise for machine ethics insofar as it offers a new way to represent normative information that is machine implementable. Presumably, though, this representation will be useful only if there is also a computationally feasible way to acquire it. So that's the issue we discuss next.

[12] This is not to imply that it would be straightforward to assign weights to reasons. In fact, the experiment can be seen as asking people to do just that.

The question we need an answer to can be put thus: how can we obtain a normative graph encoding the information that is relevant for a circumscribed range of cases? We submit that we don't have a full answer here, but we can provide a partial one. That is, we have a proposal for how to obtain the default weights of all considerations in a normative graph *if* we know its structure and also have some *training data* in the form of a set of subgraphs (without weights) for which we know the correct outcomes. Notice that each element in the training data represents a specific scenario where the relevant reasons and permissible options are known. We do not have any original proposals for how to obtain the information about the structure of the graph and the training data, but we can adopt well-known proposals from the machine ethics literature – we discuss this in Section 8. In any event, in the next two sections, we take these as given.

6 Estimating Default Weights

To estimate the default weights on the basis of training data, we can make use of metaheuristics. In this section, we discuss one particular genetic algorithm we can use.[13]

Before we can make use of a genetic algorithm, we need to "encode" our problem – finding the default weights of considerations in a normative graph. A potential solution to this problem is then represented as a "chromosome". Any genetic algorithm consists of a population of these chromosomes. Chromosomes in turn are made up of a list of "genes", where each gene represents a single variable. In our case, a gene represents the default weight of a single consideration in the normative graph. This means that in our encoding, each chromosome consists of exactly as many genes as there are considerations in the normative graph.

Any genetic algorithm consists of five steps. Step 1 is performed at the beginning; then the algorithm enters a loop: Steps 2–5 are repeated until the solution is found or the limit of steps is reached. If no solution provides perfect results, the algorithm terminates, providing its current best solution as the final output. Next, we will go through each of these steps one-by-one and note how each step is implemented in our framework.

Step 1: **Initialization.** A population of chromosomes is created, and their genes are assigned an initial value. In our implementation, the population consists of 10 chromosomes, where the genes can take on values from the interval $[0, 5]$. We set their initial value to the median value 2.5.

Step 2: **Evaluation.** We compute a fitness score for each chromosome. In our implementation, the score is the number of cases in the training data where the (weights represented by) the chromosome return the same permissible actions as found in the training data.

Step 3: **Selection.** A new population of chromosomes is generated from the previous ones. There are several selection methods, but the main idea behind them is the same: the probability of a chromosome being preserved for the next population is proportional to its fitness score. We use the *steady state* selection method. Unlike other methods, it preserves a fraction of chromosomes that have a high fitness scores – in effect, skipping Steps 4 and 5.

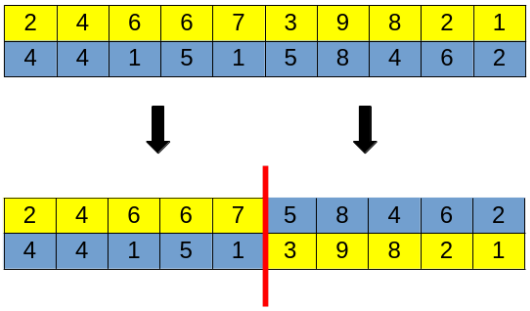

Figure 2: Example of a single point cross-over.

Step 4: **Reproduction (or Cross-Over).** Children are generated from the parent population, replacing the parents. Usually, two parents are selected, and two children produced, each having a mix of genes from both parents. (With steady state, this affects only part of the population – in our implementation 40%.) We use the *single-point* cross-over method. This method generates two children. The parent chromosomes are split in two, with children inheriting half a chromosome from each parent (see Figure 2). This is a simple and computationally inexpensive reproduction method.

Step 5: **Mutation.** For each chromosome, a small number of genes is replaced by a different value. In our implementation, one gene is selected at random and its value is substituted with a floating value from $[0, 5]$. This has the advantage of being time-efficient, and prior testing has shown that the desired result can be reached without relying on more sophisticated functions. Also, choosing a random value instead of moving to neighboring values has the advantage of getting out of local optima more easily.

7 Experiments

Our goal in this section is to evaluate the algorithm. To this end, we perform four experiments.

To test the algorithm, we need two things: training data and a validation set. Here we use *synthetic data* for both. We create this data by first generating one big normative graph containing some predetermined number of considerations and a set of actions. Like in our toy example above, this graph is taken to include all important normative considerations in some area. This means that each possible case in this area can be represented by a subgraph – a normative graph containing only some of the considerations and actions of the big graph. We then generate a set of random subgraphs and calculate the set of permissible actions for each such graph using one of the functions f_1–f_4.[14] Then

[13] This may be only one of the many working methods. We are planning to explore other methods in future work. The source code of our implementation can be found online at https://github.com/zaap38/aies2024_alcaraz_streit_knoks.

[14] For the experiments, we use weights between 0 and 5 with a 1-digit decimal precision. A single modifier (for the functions f_2, f_3, f_4) can change the weight by, at most, a factor of 6. In preliminary testing, the choice of the value interval didn't matter for the performance of the algorithm.

we "forget" the weights in the subgraphs, obtaining a set of cases with known morally correct action(s). This set is split into training data and validation set.[15]

The big normative graph we created the synthetic data from was generated with a manually defined number of considerations varying with each experiment, 5 options, and a maximal graph depth of 49. The experiments were run with a Python implementation, using the PyGAD library, and on a machine equipped with an 11th Gen Intel(R) Core(TM) i9-11950H @ 4.80GHz and 32Gb of RAM. While our implementation uses a single thread, our approach can easily be multithreaded to iterate faster over the training data. The population contains 10 chromosomes, and we perform a cross-over on 40% of the population. The generated subgraphs representing the cases all contain 5 actions, and we perform 100 steps (that is, let 100 generations evolve) in the training phase. We calculate the correct moral action(s) for 500 cases with 40% (200 cases) serving as the training data and 60% (300 cases) serving as the validation set.

Prediction Accuracy. The first experiment tests for the effects of the number of considerations involved on the algorithm's accuracy on the training data and the validation set, and the time and steps the algorithm needs to find a solution.

To test this, we used the additive function (Definition 6) to create both the training data and the validation set. We created cases from a (big) normative graph containing 5 to 100 considerations and ran the algorithm on each size. The results for each size were averaged over 20 runs.

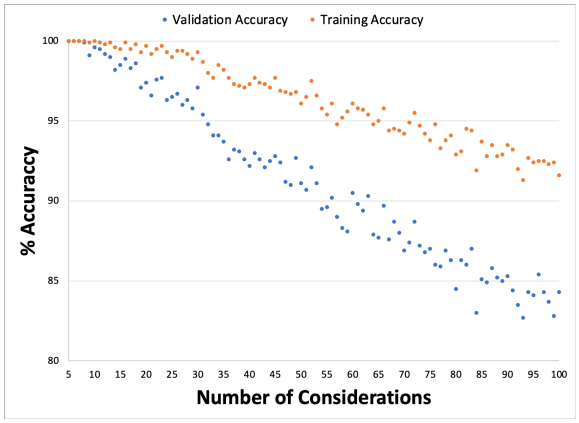

Figure 3: Accuracy

Our starting hypotheses for this experiment were the following: since the training data always contains exactly 200 cases, no matter the number of considerations, the accuracy will begin to drop with larger numbers as the training data covers fewer and fewer of all possible cases.[16] For the time it takes the algorithm to run, we expected a polynomial increase with size, as our implementation uses the grounded extension algorithm proposed by (Nofal, Atkinson, and Dunne 2021).

In Figure 3, we plot the accuracy of our prediction on the training data and the validation set over the number of considerations. In Figure 4, we plot the time it takes for one run to finish and the number of steps it takes for the genetic algorithm to stabilize on a solution.

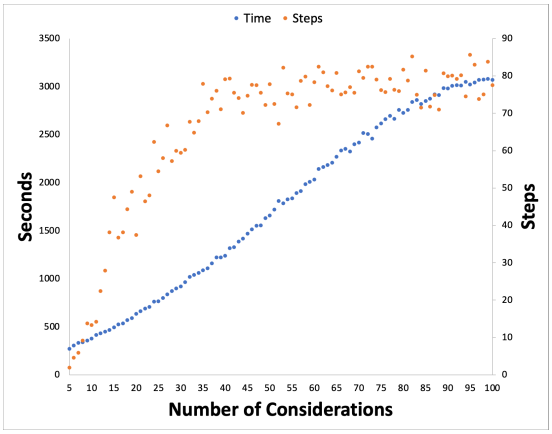

Figure 4: Time and Steps

The data supports our accuracy hypothesis. The accuracy of predictions begins to gradually drop, but stays at roughly 90% accuracy (on the validation set) for 50 considerations. Even for 100 considerations, where the algorithm is trained on only a small fraction of possible cases, the accuracy stays well above 80%. The time it takes for the algorithm to run seems to scale roughly linear with size, while the number of steps it takes to reach a solution appears to be logarithmic.

Importance of the Choice of Function. The second experiment tests whether there is a difference in accuracy between the four functions f_1–f_4.

To test this, we analyzed the accuracy of each function f_1–f_4 run on four sets of synthetic data. One set was generated by the function we were testing, while the other three were created by the remaining three functions. Then we compared the accuracy of the function on the validation data from each set. The graphs we generated the training data and validation set from contained 50 considerations, and the results were averaged over 20 runs.

We had two hypotheses for this experiment: that each function will show better accuracy on its own data set than on any other set, and that some functions will perform better on data not created by them than others.

The results, displayed in Table 1, partly confirm and partly refute our hypotheses. The additive function f_1 performs *better* on data created by other functions than on its own data, while the remaining three functions perform slightly worse on data created by other functions than on their own data. For the mixed functions f_3 and f_4 (and to a slightly lesser extent for the multiplicative function f_2), it appears to matter little which function is used to create the data. These

[15] We vary this approach slightly for the second experiment.

[16] How many cases can be created from a single big normative graph with n considerations is highly dependent on its structure. On the one extreme, for a graph with no modifiers or undercutters, there are 2^n possible cases, on the other extreme, for a graph that forms a single chain of one reason, modifiers, and undercutters there are only $n + 1$ possible cases.

	f_1	f_2	f_3	f_4
$\mathbf{f_1}$	90.3 ± 3.7	94.5 ± 2.7	95.4 ± 2.6	94.9 ± 2.8
$\mathbf{f_2}$	93.3 ± 3.3	92.1 ± 4.7	94.9 ± 2.5	95.7 ± 2.3
$\mathbf{f_3}$	92.7 ± 3.3	94.7 ± 2.4	94.7 ± 2.6	95.2 ± 2.6
$\mathbf{f_4}$	93.0 ± 3.0	94.4 ± 3.3	95.2 ± 2.5	94.6 ± 2.4

Table 1: Each row presents the accuracy on the validation set in percent when trained (and validated) on data created using the function given in the column.

results suggest that, for the purposes of designing a tool for moral learning, the (rather simplistic) additive function might be an ill fit, but the other functions perform (equally) well. We take this to be good news.

Noise Robustness. This experiment tests the effects of noise in the data on the accuracy of the algorithm. It is important insofar as real-world applications might require working with imperfect data. For instance, such data might contain cases where forbidden actions are taken to be permissible, or it might even contain inconsistencies.

We created noisy data as follows: for each case in the training set, we replaced, with a certain chance, the actual set of permissible options with a random set of options. We tested this for a $10\%/20\%/30\%/40\%$ chance and created the data from a big graph containing 50 considerations, averaging results over 20 runs.

Our hypothesis was as follows: with a small amount of noise, the algorithm will remain accurate on the validation set, but with more noise, we will see a rapid drop in accuracy.

	10%	20%	30%	40%
$\mathbf{f_1}$	91.1 ± 5.3	90.5 ± 4.2	90.4 ± 3.2	90.6 ± 3.5
$\mathbf{f_2}$	93.4 ± 3.8	92.7 ± 3.3	93.1 ± 3.0	92.6 ± 3.8
$\mathbf{f_3}$	94.6 ± 3.7	94.0 ± 3.1	95.3 ± 2.3	94.7 ± 2.8
$\mathbf{f_4}$	94.3 ± 3.8	93.0 ± 4.6	94.3 ± 3.1	94.2 ± 2.6

Table 2: Accuracy for the different evaluation function on the testing data given noisy training data.

The results, summarized in Table 2, partly confirm the hypothesis and partly refute it. When the training data contains some noise, the accuracy remains high at between roughly 90% for the additive and almost 95% for the mixed functions. However, even when 40% of the training data is noise, the accuracy never drops below an average of 90% and stays almost constant for the mixed functions.

We interpret these results as follows: first, the fact that the accuracy remains high despite the noise in data suggests that the algorithm is fit to be used on real-world data. Second, something has to account for the surprisingly high accuracy rates with as much as 40% noise in the data, and our conjecture was that the accuracy does not come from the estimates of the weights alone, but also from the fact that cases are encoded in a graph structure. So, we designed the next experiment with the aim of quantifying the effect that structure has on accuracy, independently of any weight estimates acquired with the help of genetic algorithm.

Effects of Structure on Accuracy. To test the effect of the normative structure, we created the validation set as usual. However, instead of running the algorithm on the training data, we assigned the uniform weight of 2.5 to all consideration. Then we used the validation set to check the prediction accuracy of these constant weights. (Preliminary tests have shown that other (positive) values produce similar results.) We ran this test using all four functions and averaged over 1000 runs.

Since we observed high accuracy even with highly noisy data, our hypothesis for this experiment was that the structure is responsible for at least 50% of the accuracy.

f_1	f_2	f_3	f_4
62.0 ± 7.8	69.1 ± 8.7	72.6 ± 8.3	72.5 ± 8.2

Table 3: Accuracy in percent when using constant weights.

The results, shown in Table 3, fully confirm the hypothesis. Using constant weights results in about 70% accuracy for every function, except for the additive one, which, again, performs significantly worse than others. We interpret these results as follows: in a large number of cases, the normative structure is enough to identify permissible actions. This suggests that the default weights of considerations – much like the choice of the functions for calculating final weights – are less important than one might have initially thought. Getting good default weight estimates accounts for only around 30–40 percentage points of prediction accuracy.

8 Discussion

In this section, we highlight the hybrid nature of our approach and its advantages; hint at avenues for future research; and relate our approach to the closest work in machine ethics, weighted argumentation, and metaethics.

The approach to machine ethics we have presented combines two elements. The first is a novel way of representing normative information – using reasons, modifiers, undercutters, and numerical weights – that can be interpreted by machines and humans alike. This representation rests on solid philosophical foundations. The second element is a proposal for how to implement (some) ethical learning by deriving the weights of reasons and other normative considerations from training data using a genetic algorithm. The fact that our approach combines these two elements makes it hybrid.

But, as we noted in Section 5, the two elements do not tell the entire story. To make use of our approach in specific applications more is needed: there has to be a way to infer the normative structure of cases, in order to both provide training data and give a formal description of any case where we want the model to predict the correct action(s). This information has to come from somewhere. One possible source are domain experts – for example, professional ethicists – who can describe the normative structure of cases in some well-understood area. How good and how consistent the training data provided by experts would be is something we would like to investigate further, but we are optimistic. The results of our four experiments suggest that, with

the structural representation in place, our approach can be applied to real-world problems. The algorithm remains performant even for large numbers of considerations and noisy data. Still, there is more to explore. So far, all the graphs we have generated were random. This is true for both the big graph we generated the training data and validation set from and for the individual subgraphs that represented cases in each of these two sets. From our fourth experiment, we know that the structure has considerable influence on the accuracy of predictions. In future work, we intend to study the effects of different graph structures on accuracy in more detail. In particular, we want to explore whether the algorithm works better for graphs that contain fewer modifiers and undercutters, as well as whether it works better for graphs that are highly connected.

We already noted that, in the context of machine ethics, the framework that seems to be most similar to ours is GenEth. So, a comparison of the two is in order. The basic element of GenEth are (manually selected) "ethically relevant features" which play a similar role to our normative reasons. Like our approach, GenEth learns on the basis of a set of cases for which the outcomes are known. Cases are represented as sets of (exclusive and exhaustive) actions characterized by the ethically relevant features and represented as tuples of integers (usually taken from $[-2, 2]$) indicating the degree to which they promote or violate each feature. To take a simple example, we might have a case comprised of actions A and B where $A = (1, -1)$ and $B = (-1, 2)$, and where the ethically relevant features are, respectively, harm coming to some patient and this patient's autonomy being respected. What the case says, roughly, is that the action A mostly prevents harm at the expense of some disrespect of autonomy, while B strongly respects autonomy but doesn't prevent moderate harm. GenEth then uses inductive logic programming to extract, roughly, the most general principle sanctioned by the cases in the training set.

We highlight only what we think are the most important differences between our approach and GenEth: the first has to do with the way normative information is represented, the second with how learning works. The GenEth model is what we might call *flat*: all normatively relevant considerations are of the same type. Here our model has the upper hand, since it is more flexible, allowing to include more structural information. It is also more in line with the literature in metaethics.[17] Furthermore, in our model, the weights of reasons can be as fine-grained as desired, whereas GenEth, by design, has to rely on a rather coarse categorization of actions into categories (usually 5) of promotion and violation of some feature. Turning to ethical learning, GenEth's use of inductive logic programming allows it to output ethical principles encoded in a logical language. This is something our model – without being supplemented with a logical language – cannot do. However, our model appears to have a different advantage over GenEth. The third experiment suggests that our approach can be used on noisy and inconsistent data. GenEth, by contrast, cannot be immediately used on such data: for inductive logic programming to be applicable, inconsistencies in the data need to be removed.

We have already mentioned that our model draws on the work of Amgoud et al. They work with weighed bipolar argumentation graphs, assigning both (analogues of our) default and final weights to arguments and interpreting the latter as the "overall level of acceptability" of arguments. Their main focus is on exploring various "acceptability semantics" or, roughly, procedures for assigning final weights to arguments on the basis of the relation between arguments and their default weights. We adopted and adjusted this general idea for a particular application. It is not difficult to verify that our normative graphs can be represented as what we might call *weighted tripolar argumentation graphs*, with amplifiers, attenuators, and undercutters each mapping to their own relation. Not surprisingly, Amgoud et al. have nothing corresponding to the idea of determining deontic statuses by weighing reasons (Definition 7). Also, they do not restrict attention to graphs that are finite and acyclic. Another argumentation-theoretic model that shares similarities with ours is presented in (Gordon and Walton 2016), but the two also have many differences, and their comparison will have to be undertaken elsewhere. Neither Amgoud et al., nor Gordon and Walton apply their ideas to machine ethics.

We conclude with a brief remark on metaethics. The formal model we set up in Section 4 is, we believe, the first *faithful* formal characterization of the informal weighing reasons model. It has some clear advantages over its closest competitors: other formal models of normative reasons. Thus, the default logic-based approach of (Horty 2012) attempts to represent the weights of reasons with the help of a partial order and, therefore, has no natural way to model the aggregation of weights. And the more recent decision and probability theory-based approaches (Nair 2021; Sher 2019) – which have been developed in response to the problem of aggregating weights – appear to be unable to represent undercutters and modifiers in an explicit and natural way.

9 Conclusion

We presented a new hybrid approach to representation and acquisition of normative information for machine ethics. We started with a sketch of an influential and general metaethical account of the structure of morality. Then we presented a formalization of this account: in it, the normative structure of any given situation is represented using a graph of a particular shape and the deontic statuses of actions are determined on the basis of weights of nodes representing normative reasons and other normative considerations. Next, we introduced a genetic algorithm that lets one estimate these weights on the basis of a set of cases for which the deontic statuses are known. These weight estimates are important insofar as they can be used to assign deontic statuses to options in previously unseen cases. Finally, we reported on some (early) experimental results illustrating the advantages of our approach. Both its philosophical underpinnings and the experimental results suggest that the approach holds promise for real-world applications.

[17] Although (Anderson and Anderson 2018) mention "meta-features" in passing – their examples are time and probability – they also suggest that these shouldn't be part of the representation of the normative structure (see pp. 342–3).

Ethical Statement

The approach to machine ethics that we present here is in the early stages of research. The framework we have set up is not (yet) ready for any real-world application. The underlying philosophical model, while undoubtedly popular, is not uncontroversial in the philosophical literature. The framework has not been tested with real-world data, and extensive tests and careful evaluation would be needed before its application in real-world situations. Also, before the use of any algorithmic decision-making system is implemented, a robust culture of accountability needs to be in place as algorithmic decision-making systems can be (and have been) used as a tool to avoid legal and moral responsibility.

Acknowledgments

Alcaraz and Knoks acknowledge financial support from the Luxembourg National Research Fund (FNR). Alcaraz was supported through the project C21 (IPBG2020/IS/14839977/C21); Knoks through the project EAI (C22/SC/17111440). We would also like to thank our three anonymous referees for their comments.

References

Abel, D.; Macglashan, J.; and Littman, M. L. 2016. Reinforcement Learning as a Framework for Ethical Decision Making. In *AAAI Workshop on AI, Ethics, and Society*.

Allen, C.; Smit, I.; and Wallach, W. 2005. Artificial morality: Top-down, bottom-up, and hybrid approaches. *Ethics and information technology*, 7: 149–155.

Amgoud, L.; and Ben-Naim, J. 2018. Evaluation of arguments in weighted bipolar graphs. *International Journal of Approximate Reasoning*, 99: 39–55.

Amgoud, L.; Ben-Naim, J.; Doder, D.; and Vesic, S. 2017. Acceptability semantics for weighted argumentation frameworks. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI2017)*, 56–62.

Amgoud, L.; Doder, D.; and Vesic, S. 2022. Evaluation of argument strength in attack graphs: Foundations and semantics. *Artificial Intelligence*, 302: 1–61.

Anderson, M.; and Anderson, S. L. 2007. Machine Ethics: Creating an Ethical Intelligent Agent. *AI Magazine*, 28(4): 15–26.

Anderson, M.; and Anderson, S. L. 2018. GenEth: A general ethical dilemma analyzer. *Paladyn, Journal of Behavioral Robotics*, 9(1): 337–357.

Arkin, R. 2009. *Governing Lethal Behavior in Autonomous Robots*. Chapman and Hall/CRC.

Awad, E.; Dsouza, S.; Kim, R.; Schulz, J.; Henrich, J.; Shariff, A.; Bonnefon, J. F.; and Rahwan, I. 2018. The Moral Machine Experiment. *Nature*, 563(7729): 59–64.

Baker, D. 2017. The varieties of normativity. In McPherson, T.; and Plunkett, D., eds., *The Routledge Handbook of Metaethics*, 567–581. Routledge.

Balakrishnan, A.; Bouneffouf, D.; Mattei, N.; and Rossi, F. 2019. Using multi-armed bandits to learn ethical priorities for online AI systems. *IBM Journal of Research and Development*, 63(4/5): 1–1.

Baroni, P.; Gabbay, D.; Giacomin, M.; and van der Torre (eds.), L. 2018. *Handbook of Formal Argumentation*, volume 1. London: College Publications.

Bench-Capon, T.; Robinson, G.; Routen, T.; and Sergot, M. 1987. Logic programming for large scale applications in law: a formalisation of supplementary Benefit Legislation. In *Proceedings of the First International Conference on Artificial Intelligence and Law (ICAIL-87)*, 190–198.

Benzmüller, C.; Parent, X.; and van der Torre, L. 2020. Designing normative theories for ethical and legal reasoning: LogiKEy framework, methodology, and tool support. *Artificial intelligence*, 287: 103348.

Berker, S. 2007. Particular reasons. *Ethics*, 118(1): 109–139.

Bremner, P.; Dennis, L. A.; Fisher, M.; and Winfield, A. F. 2019. On proactive, transparent, and verifiable ethical reasoning for robots. *Proceedings of the IEEE*, 107(3): 541–561.

Broome, J. 2004. Reasons. In Wallace, R. J.; Pettit, P.; Scheffler, S.; and Smith, M., eds., *Reasons and Value: Themes from the Moral Philosophy of Joseph Raz*, 28–55. Oxford: Clarendon Press.

Canavotto, I.; and Horty, J. 2022. Piecemeal Knowledge Acquisition for Computational Normative Reasoning. In *Proceedings of the 2022 AAAI/ACM Conference on AI, Ethics, and Society*, AIES '22, 171–180. New York, NY, USA: Association for Computing Machinery. ISBN 9781450392471.

Dancy, J. 2004. *Ethics without Principles*. New York: Oxford University Press.

Dancy, J. 2017. Moral Particularism. In Zalta, E., ed., *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, Winter 2017 edition.

Dietrich, F.; and List, C. 2013. A reason-based theory of rational choice. *Noûs*, 47(1): 104–34.

Dung, P. M. 1995. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 77(2): 321–57.

Dunne, P.; Hunter, A.; McBurney, P.; Parsons, S.; and Wooldridge, M. 2011. Weighted argument systems: basic definitions, algorithms, and complexity results. *Artificial Intelligence*, 175(2): 457–486.

Fischer, M.; and Dennis, L. A. 2023. *Verifiable Autonomous Systems*. Cambridge University Press.

Forrest, S. 1996. Genetic algorithms. *ACM computing surveys (CSUR)*, 28(1): 77–80.

Gabbay, D.; Giacomin, M.; Simari, G.; and Thimm (eds.), M. 2021a. *Handbook of Formal Argumentation*, volume 2. London: College Publications.

Gabbay, D.; Horty, J.; Parent, X.; van der Meyden, R.; and van der Torre (eds.), L. 2013. *Handbook of Deontic Logic*

and Normative Systems, volume 1. London: College Publications.

Gabbay, D.; Horty, J.; Parent, X.; van der Meyden, R.; and van der Torre (eds.), L. 2021b. *Handbook of Deontic Logic and Normative Systems*, volume 2. London: College Publications.

Ganascia, J.-G. 2007. Modelling ethical rules of lying with answer set programming. *Ethics and Information Technology*, 9: 39–47.

Glover, F. 1986. Future paths for integer programming and links to artificial intelligence. *Computers & operations research*, 13(5): 533–549.

Gordon, T.; and Walton, D. 2016. Formalizing Balancing Arguments. In Baroni, P.; Gordon, T.; Scheffler, T.; and Stede, M., eds., *Proceedings of the 2016 Conference on Computational Models of Argument (COMMA 2016)*, 327–38. IOS Press.

Horty, J. 2003. Reasoning with moral conflicts. *Noûs*, 37(4): 557–605.

Horty, J. 2012. *Reasons as Defaults*. Oxford University Press.

Jiang, L.; Hwang, J. D.; Bhagavatula, C.; Bras, R. L.; Liang, J.; Dodge, J.; Sakaguchi, K.; Forbes, M.; Borchardt, J.; Gabriel, S.; Tsvetkov, Y.; Etzioni, O.; Sap, M.; Rini, R.; and Choi, Y. 2022. Can Machines Learn Morality? The Delphi Experiment. arXiv:2110.07574.

Keeling, G. 2023. A Dilemma for Reasons Additivity. *Economics and Philosophy*, 39(1): 20–42.

Laarhoven, P.; and Aarts, E. 1987. *Simulated annealing*. Springer.

Lord, E.; and Maguire, B. 2016. An opinionated guide to the weight of reasons. In Lord, E.; and Maguire, B., eds., *Weighing Reasons*, 3–24. Oxford University Press.

Mossakowski, T.; and Neuhaus, F. 2018. Modular semantics and characteristics for bipolar weighted argumentation graphs. *arXiv preprint arXiv:1807.06685*.

Nair, S. 2021. "Adding up" reasons: Lessons for reductive and non-reductive approaches. *Ethics*, 132(1): 38–88.

Nofal, S.; Atkinson, K.; and Dunne, P. E. 2021. Computing grounded extensions of abstract argumentation frameworks. *The Computer Journal*, 64(1): 54–63.

Oren, N.; and Yun, B. 2023. Inferring Attack Relations for Gradual Semantics. *Argument and Computation*, 14(3): 327–345.

Parfit, D. 2011. *On What Matters (Volume I)*. Oxford University Press.

Raz, J. 1990. *Practical reason and norms*. Oxford University Press.

Ross, W. D. 1930. *The Right and the Good*. New York: Oxford University Press.

Rossit, J.; Mailly, J.-G.; Dimopoulos, Y.; and Moraitis, P. 2021. United we stand: Accruals in strength-based argumentation. *Argument & Computation*, 12(1): 87–113.

Russell, S.; Dewey, D.; ; and Tegmark, M. 2015. Research priorities for robust and beneficial artificial intelligence. *AI Magazine*, 36(4): 105–114.

Sayre-McCord, G. 2023. Metaethics. In Zalta, E. N.; and Nodelman, U., eds., *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, Spring 2023 edition.

Scanlon, T. M. 1998. *What We Owe to Each Other*. Cambridge, MA: Harvard University Press.

Sergot, M.; Sadri, F.; Kowalski, R.; Kriwaczek, F.; Hammond, P.; and Cory, H. T. 1986. The British Nationality Act as a logic program. In *Communications of the Association for Computing Machinery 29*, 370–386.

Sher, I. 2019. Comparative value and the weight of reasons. *Economics and philosophy*, 35: 103–158.

Snedegar, J. 2018. Reasons for and reasons against. *Philosophical Studies*, 175: 725–43.

Tucker, C. 2023. A Holist Balance Scale. *Journal of the American Philosophical Association*, 9(3): 533–553.

Tucker, C. forthcoming. *The Weight of Reasons: A Framework for Ethics*. New York: Oxford University Press.

Wedgwood, R. 2022. The Reasons Aggregation Theorem. *Oxford Studies in Normative Ethics*, 12: 127–148.

Wu, Y.-H.; and Lin, S.-D. 2018. A low-cost ethics shaping approach for designing reinforcement learning agents. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.