Ethics, Aesthetics and Computational Creativity

Daniel G. Brown
David R. Cheriton School of Computer Science
University of Waterloo
dan.brown@uwaterloo.ca

Dan Ventura
Computer Science Department
Brigham Young University
ventura@cs.byu.edu

Abstract

We explore how the aesthetic lens of computational creativity can be used to aid in the development of ethical principles for artificial intelligence systems, and their application to real-world domains in which computers are expected to make reasoned, ethical judgments. In particular, we bridge two recent ICCC papers, one about how creative computers can design ethical principles, and one that uses algorithmic information theory as one component of the aesthetic value of the artifact. Our finding is that computational creativity ideas can enable the creation of novel ethical principles, but that the use of novelty, value and typicality measures in this space is quite challenging, and in particular, the algorithmic information theory objectives do not map smoothly to the goal of building fast ethical systems of provably high quality. We conclude with suggestions for making our approach usable in practice.

Introduction

AI systems, particularly those that inhabit the physical real world, make ethical decisions in response to either constructed dilemmas or ordinary scenarios all the time. This happens when they decide how to respond to human or other actors in need of help, but it also happens when a stock-picking robot decides which companies to invest in, or when an algorithm chooses which candidate to offer a job to, or (perhaps more) when the algorithm identifies which personal traits to look for in a successful candidate.

The all-pervasive nature of these ethical choices has caused "ethical AI" to become one of the current most active areas of research, teaching and progress in the area, with entire conferences devoted to engendering fairness (by multiple definitions), identifying properties of fair systems that are mutually incompatible, and reporting on situations in which an AI produces outcomes that are unfair, as when they make decisions that either confirm or exacerbate existing inequities, or when decisions are made by an AI for reasons that seem arbitrary. As such, concerns about training data bias, explainability, or the presence or absence of proxy variables that can be used to substitute for variables upon which discrimination is forbidden (such as height being a proxy for gender, or postal address as a proxy for race or income level) have also become major topics of research.

We argue in this paper that computational creativity (CC) has a message for AI ethics as well, but our focus is in fact that CC can produce ethical systems whose principles are themselves presented in a more aesthetic or satisfying way, and that the constrained exploration found in most CC systems can produce diverse systems of ethical principles, rather than reinforcing existing models for how robots or AI systems should interact with the world. Our argument bridges two recent ICCC papers: a paper by Ventura and Gates, which argues that considering AI systems as creative agents whose output artifacts are behavioral choices admits a natural approach for imposing ethics as an aesthetic filter on that behavior (2018); and the algorithmic information theory-based approach of Mondol and Brown, which seeks to use measures from that theory (basically, a few concepts from advanced Kolmogorov complexity) as indicia of high novelty and value (2021). A challenge with both of these papers is that they do not offer practical implementations.

Ventura and Gates consider the problem of ethical AI behavior on two levels. First, they propose a base-level system which considers potential behavioral choices and evaluates those choices via the lens of a normative ethics, which acts as an aesthetic and imposes ethical notions of novelty and value. They examine various classical normative ethical philosophies as possible behavioral aesthetics and conclude that the choice of *which* normative ethics should be used is a fraught one. In the most far-reaching part of their paper, they suggest building a meta-level creative system that creates abstract ethical principle sets and then focus on how to evaluate the ethical appropriateness of these meta-level artifacts, taking into consideration (meta-level) aesthetic notions such as novelty, utility, fairness, generalizability and comprehensibility.

Mondol and Brown also focus on quality and novelty, but their approach is much more abstract. For value, they indicate that an object is of high quality if it represents the output of a significant amount of computational effort (so called *logical depth*) or if it is an arbitrary member of a set described by a quite long program (also called *sophistication*). Objects with high logical depth are compressible; that is, they can be summarized with a very short program, but the program takes a lot of time to reconstruct the original object. Highly sophisticated objects are unusual, in that they show a large amount of internal structure, but describing that internal structure still requires substantially long descriptions; they are not just routine repetitions. Both of these measures are uncomputable. Another challenge is that if one is given a short, slow program that generates an object S, that alone is not proof that S is actually logically deep—another short program may also generate S, but do so in speedy run time, thereby demonstrating that S is in fact shallower than proposed. Similarly, just because there exists a long program for which S is a typical output (which might suggest that S is sophisticated) does not mean that there is not also a much shorter program that will also generate S as a typical output.

In the rest of this paper, we look into some of the properties of good ethical systems from the point of computational creativity, and explore ways in which Mondol and Brown's models of novelty and value can enter into the project of generating, comparing and evaluating ethical systems. We look into some of the contexts in which these systems might be used, and how creativity enters into the process of making ethical decisions as well. We explore how aesthetic judgments must be constrained here as well—just as a sonnet-generation system must create outputs that follow the rules and constraints of a proper sonnet, a system that builds ethical models must avoid horrifying outputs that may initially appear aesthetically pleasing.

Our approach is still theoretical—algorithmic information theory builds a collection of potentially theoretically sound, but fundamentally impractical, assessment tools for exploring the quality of objects like legal codes or computational systems, and designing algorithms to creatively explore the space of possible ethical systems will require a better sense of how to encapsulate both ethical systems and the dilemmas they face in a computational fashion. However, the combination of these two approaches offers the possibility of *incorporating aesthetics and ethics into a common framework, and developing a better understanding for how beauty and judgement can work together*. That said, there are key elements missing from "purely computational" frameworks—an aesthetics for ethics must also include a discussion of outcomes of ethical decisions, not just an analysis of their computational complexity and sophistication.

Two types of ethical system

Ethical decisions are made by computers or robots (or humans!) in response to specific situations. Here, we give two different formalisms for quite different scenarios in which agents make these decisions and describe how aesthetics can enter into the evaluation of this process. These two scenarios correspond to a large degree with the two levels of ethical CC agent treated by Ventura and Gates: the first considers the primary artifact to be a behavior (though we will see that this, in fact, may not actually be the locus of the creativity); the second considers the primary artifact to be analysis and judgement about behavior.

First, consider an agent residing in the real world. As a result of the state of that world, one piece of the process that decides what action the agent will take might be ethical—in response to a situation S, the ethical system P must quickly compute the best behavior b^* for the agent to perform. Or, P might generate a ranked list of behaviors $(b_1, b_2, \ldots, b_n)$, which information the agent uses in deciding what step to take next. In addition, each behavior may include a formal analysis A of why it makes a good choice. The key concern in this frame, though, is not interpretability; it is efficiency—for real-time decision-making, the system must compute $(b^*, A) = P(S)$ within a time bound t, or the decision will become moot. Nonetheless, for analysis of decisions that have been made by P, it is essential that its decisions are given along with some traceable analysis of where the decision b^* came from. Since P is fast, this can in theory be just a computation trace, which may be substantially unclear due to either deliberate or accidental obfuscation. Or, despite the fact that b^* must be computed quickly, it is possible that A may be computed reflectively and therefore much more slowly; indeed, many human justifications, both ethical and otherwise, may be computed this way as well (Bem 1972).

Whether or not A is interpretable and whether it is computed in real-time or post-hoc, it is arguably a much more significant output of P than is b^*, both from a creativity and from an ethical standpoint, especially if the set B of possible behaviors is well-defined and finite.[1] Nevertheless, in this instance, the agent can not be considered to be making deep ethical decisions, because it does not have time to do so; rather, it is quickly deciding how a (presumably) previously well-defined ethics P applies in situation S.

Second, consider the phenomenon of using legal codes to resolve disputes or trials. Here, there are two levels to the process: in the first, lawmakers must draft the law code C to be used as a basis for decisions. Traditional law codes, of course, are written in ambiguous natural language, and surrounding a code C will be existing jurisprudence and commentary C' upon which decisions can be easily hung.

Next, to respond to a dispute D, the judge must use reasoning based on the current law code C to produce an explainable outcome O for dispute D, such as a guilty verdict or a decision about financial damages (presumably from a well-defined and limited set $\mathcal{O}$ of possibilities), as well as a justification J for that outcome. As before, because $\mathcal{O}$ is (usually) a finite set, the justification J is the more creative task, as is building the way in which J comes from the law code and interpretations being used.

Both creative steps in this process are interesting from a computational creativity perspective: drafting C and drafting commentaries C' allows for one to explore questions of novelty and its appropriateness in a constrained space (we would not want to accidentally legalize murder as a "novel" innovation!), while at the same time, the choice of law code can enable simpler reasoning or force more complex reasoning in cases wherein the evidence of a particular case D is not well situated within the code C. As such, the "creativity" (particularly in the sense of novelty and value, but also in the sense of expressivity and conceptualization) of one choice of C or C' can have impacts on that of the other, and

[1] Classic ethical dilemmas, such as the well-known family of trolley problems, offer an agent two choices; which choice is made is never the point of proposing the dilemma, rather it is to elicit a justification for choosing that behavior (Thomson 2014).

they both have effects on O.

The difference between a law code C and commentary C' matters because law codes ought to be interpretable under a wide variety of situations, and in new contexts; for example, anti-discrimination law might not directly cite newly-protected groups, but expert commentary might offer arguments under an existing code that could be adapted to other groups than those previously identified.

We go into more detail about this mode below, but in this frame, an ethical decision process P is the creation (either in natural language, or in an interpretable format) of the pair $(O, J) = P(C, C', D)$.

Ethical decisions and quick actions

Many ethical dilemmas must be solved very quickly as part of an agent's participation in the world: should the agent interfere in an argument, call out a colleague for sexist behaviour, apply for an open job, choose one candidate for a job over another, and so on. So-called "trolley problems" (Thomson 2014), in which an agent must make a decision that will have negative consequences regardless of the choice made, also fit in this framework. These ethical dilemmas are encapsulated by a short description of the scenario and a straightforward decision that the agent has to make about the action it will take; perhaps this comes along with a predicted probability distribution over the states of the world that can result from the decision.

This formulation can easily turn into a Partially-observable Markov Decision Process (POMDP) if one simply optimizes expected long-term utility of the local decision at each step (Kaelbling, Littman, and Cassandra 1998). To avoid this potentially worrisome outcome, we note some ways in which this framing differs from the POMDP model.

First, the computational power of the agent may be restricted to the extent that solving a full POMDP may simply not be possible; at each step, there may be too many possible outcomes to choose from to do a full calculation of expected long-term utility, for example. Fundamentally, the decision maker operates in a position of bounded resources: it cannot fully model other actors in the situation, it may not have a proper description of the immediate probability distribution (let alone the long-term outcomes of choices) resulting from the decision it is making, and the limits on its computation may restrict the functions it can optimize. This is essentially the same argument used as one explanation for "non-rational" human behavior (Simon 1955).

Second, even the utility itself is not an easily-defined function. Instead, the agent itself will learn its utility function by assessing the outcomes of situations, both those that result from its own decisions, and those it is shown while it is being trained. As a result, it is even possible that this utility function will be time-dependent.

In this framework, computational creativity mostly enters into the design of the agent's ethical system itself and the assessment of its qualities. In particular, we look for aesthetic qualities in the way in which the agent responds to situations (both those found in training data and those found in its own experiences): can the agent's decision-making be said to model principles, can it be summarized in a way that generalizes from pre-existing data, and can it be expressed in a compact and easily computed way? A high-quality system should also be unaffected by irrelevant changes in the input, which in fact will allow it to operate with more efficiency. We also look to novelty: does the summarization algorithm function differently from previous algorithms despite generalizing the same data? One way to see this, consistent with Mondol and Brown, is to say that the algorithm derived to do fast ethical decision-making is not "typical" of existing algorithms of that sort—knowing *how* those algorithms work will not offer much assistance in compressing the description of a new ethical decision-making approach.

These aesthetic principles of parsimony, generalizability, consistency and (perhaps to a somewhat lesser extent) novelty are what we view as core ideas of a speedy ethical system. Can they be adapted to an algorithmic information theory model of value?

Legal decisions

Law codes have existed for millennia. Historically, they began largely as criminal codes, identifying behaviours not permitted for residents of a city or nation and the consequences thereof; over time, they expanded to much larger codes accommodating trade issues, family law and so on. At various times, magistrates and judges are given the task of applying existing legal codes to evidence coming from specific cases; they must build arguments based on the case, the law codes, and previous cases and their commentaries. Having humans do this work requires huge effort and expense: legal scholars must be trained to build and present arguments, and judges must use enormous law libraries full of precedents and commentary to adapt their jurisprudence to the situations of a contemporary dilemma.

We view this process as a set of creative tasks. In fact, from an aesthetic point of view, there are three quite different tasks that occur when one uses a law code to resolve a case. The first is the codification of the relevant law code itself—summarizing a collection of arguments, traditions and customs into a short natural language formulation.

The second aesthetic task is perhaps less obvious: how to present the information of a case. If the case is presented in a way that is true, but which obscures the way in which the law attaches to the details of the case, it can require much argumentation to be built in order to properly describe the decision (O, J) that best resolves the dilemma.

And the third aesthetic task is the one that is perhaps most interesting, and which can be assessed in a variety of ways: the process by which a judge (computational or human or a combination of the two) can build from a legal code C and commentary system C', and from the evidence of a case D to an outcome O with a justification J. If judgment is to be made by a computer, then the task is in a sense one of using existing arguments from C', together with rules from C, applied to case D to create the decision pair (O, J). If (O, J) is easily derived from the evidence and the legal information, then we can say that the bulk of the effort in the case was already done in the creation of those processes (and in the training of the computational judge). If, instead, much hair-splitting must be done in the task of interpreting the evidence of the case, then we can say that the code was not well-matched to the evidence.

This offers one of our most tantalizing realizations: namely, that the computational task of coming up with judgments can be seen as finding an efficient function that maps evidence D onto judgments J by filling in details from J quickly. If the decision J is easily created from D, given (C, C') as a legal code and advice, then (C, C', D) is a good set of evidence and laws for the case. In particular, we can say that knowing C' can help us resolve the case more straightforwardly.

To make this more formal, consider a collection of evidence D. Suppose there is a small set of possible outcomes $\mathcal{O}$ defined by the legal code C for cases of the type of D. In order to resolve the case, we must come up with the $O \in \mathcal{O}$ that best represents how D interacts with (C, C'), and the explanation J that requires the least extra computation on top of what has already happened in the creation of (C, C').

Creating an ethical decision process, then, consists of choosing a good decision-maker P, but also "priming the pump" by ensuring that P is well adapted to the law code C; in particular, P should be able to come up with verdicts for most cases quickly, and the pair (O, J) should be easily computed (at least for most cases) given the input data (C, C', D). In the language of Kolmogorov complexity, this corresponds to saying that the conditional Kolmogorov complexity of the decision (O, J) is small, given (C, C', D).

In particular, we note that a legal code that requires us to build long, involved judgments for simple cases, or for which small changes to the evidence could force us into a completely different set of valid justifications, is not a good one. Rather, for most cases, the mapping from D to the pair (O, J) needs to be efficient; that is, the legal code is preprimed to make fast, straightforward decisions.

Novelty and value in the context of ethics

Adapting traditional creativity measures to ethical systems and their products is a challenge. In particular, one principle that might be considered desirable in an ethical system is respect for precedent and tradition, which pushes these systems in a direction that moves away from novelty. Obviously, we still can look for new ways of reconsidering ethical dilemmas (either new ones or pre-existing ones), in the service of discovering a better way of improving people's lives, or in terms of mutually explaining a large number of compatible decisions. In this sense, the novelty of an ethical system is about the arguments it generates. As to value, the quality of an ethical system depends not only on the ostensible beauty of its philosophical tenets but also on objective observers' agreement with the decisions the system makes.

And of course, for scenarios in which the output of an ethical system is an argument or a legal code or a text decision, one can look at the overall quality of the text drafting, but of course, there is no value in a beautiful argument that creates a monstrous conclusion. In this sense, creativity may not always serve good outcomes, as when terrorists design novel forms of sabotage (Cropley, Kaufman, and Cropley 2008).

We can also look for some quality measures that are similar to those used by Mondol and Brown, which seek to encapsulate a collection of compatible ideas in a highly-compressible form with little internal redundancy. If generalizing these ideas can be done with a lot of effort, resulting in a short program that compresses the initial representation well, then this can be another indication of the value of the ethical system. Obviously, arguing about brevity alone is insufficient—an ethical system of the "kill all who oppose us" variety is clearly not a wise one despite its simplicity; rather, it is clear that wise ethics requires evidence of nontrivial thought from humans, or for computers, evidence of substantial computation.

Complexity-theoretic notions of aesthetic

Here we give a short introduction to the algorithmic information theory concepts Mondol and Brown use in their study of aesthetics. They use both *sophistication* and *logical depth* as measures of quality; we here focus on the simpler of these, which is logical depth. We also briefly summarize their approaches to novelty and typicality in non-technical language.

Basic Kolmogorov complexity concepts

A string s over a finite alphabet has Kolmogorov complexity $K_U(s)$ when this quantity is the length of the shortest input to a universal Turing machine U upon which U halts with the string s on its output tape. When U represents a programming language, $K_U(s)$ is the length of the shortest program in that language whose output is s; normally, we ignore the universal machine U and just speak of $K(s)$. There are a number of details about U that are also necessary (such as its accepting only a prefix-free language); we refer the reader to Li and Vitányi (2019) for full details.

The quantity $K(s)$ is uncomputable. In reasonable programming languages U, $K_U(s) \leq |s| + c$ for some constant c, since one can just write a program that prints out the symbols of s one-by-one. In general, $K(s)$ represents an optimal compression of the information found in s. The value of $K(s)$ is not correlated with the usefulness of s. A random string has $K(s) \approx |s|$, which is high. The string 1^n of n consecutive 1's has $K(s) \leq \log n + c$, since we can just write down the binary representation of the value n and then spit out that many 1s; this is a very low value of $K(s)$. (Certain values of n can be compressed far smaller than $\log n$; for these strings, $K(s) \ll \log n$.) And a string s of k random bits followed by $n - k$ 1's will have $K(s) \approx k + \log(n - k)$, which can take any value between $\log n$ and n. Knowing (just) the Kolmogorov complexity gives no way of distinguishing "useful" strings or "creative" strings from others.

Logical depth as value

Instead, Mondol and Brown move to estimate the value of a string s by its logical depth (Bennett 1988), the run time needed by short programs that compute s. Specifically, given a slip constant c,

$$d_{U,c}(s) = \min_{P:U(P)=s, |P| \leq K(s)+c} \text{time}(U(P))$$

that is, it is the minimum runtime of a program which generates s and whose length is within c of $K(s)$; again, both

U and c are often elided when they do not make the situation clearer. For simple strings, like those mentioned in the previous paragraph, $d(s) = O(|s|)$, because a `PRINT` program—in the case of the random string—and a linear-time FOR loop—in the case of the repeated symbol—will suffice to generate such strings (a simple combination of the two approaches suffices for the combination string). By contrast, a string s that contains the first n bits of a numerical constant that is hard to compute may be produced by a program whose length is a constant (plus a representation of the value n) but which takes a very long time to run; these are the logically deep strings. A short, slow program P whose output is a logically deep string s compresses that string very well, but an outside observer who does not have all the time needed for P to run will not be able to verify that it has a short program even if P is provided.

The overwhelming majority of strings are not even compressible, let alone logically deep (Li and Vitányi 2019). Mondol and Brown offer logical depth as one piece of evidence of the aesthetic value of a string; they propose that if a string is the output of a substantial, interesting piece of computation (or thought), then it is inherently valuable. One other component of this thesis is that as the length of s gets substantial, its availability to be compressed also grows; in particular, if s is the first n bits of a hard-to-produce constant, but the short, slow programs to produce that constant are longer than n bits long, then s is not logically deep—its shortest representation might in fact just be the program $\mathtt{PRINT}_s$. As such, logical depth is only meaningful as a function of long strings. By contrast, for long strings that come from repeated samples from a logically-deep creator, as the supply of these samples grows, the potential for finding repeated patterns and structures in those samples increases, and thus so may the possibility of actually finding a good compression method for such strings, including one that might require more complex algorithms than just "repeat this pattern k times". Logical depth is a *property* of the string s, but the evidence for it is the short, slow program that generates the string. Given such a program P, we can confirm that the string is deep by running all programs of length at most $|P|$ for the time that P takes to generate s to see if any of them can produce s in less time, but this is impractical (indeed, so might be running P itself).

Using logical depth as a proxy for the value of an object raises a number of concerns, not the least of which is that it does not constrain the object to be properly a member of the class it needs to belong to; the text of a book might be logically deep, but if it is a brilliant mathematical text written in German, it is still not a high-quality English novel. Part of our goal with this paper is to consider this question of constraints—if suitably constrained by precedent, genre and custom, can logical depth serve as a proxy for value? A logically-deep legal code summarizes a large collection of cases in a very tight package, and the added information and computation needed to resolve dilemmas can be small; by contrast, an arbitrary legal code will either be trivial because it is simple, or trivial because it is random.

Conditional Kolmogorov complexity as novelty

Kolmogorov complexity also offers the possibility of identifying whether a new creative product is truly novel: an object is novel if knowing other members of its class offers little information about the new object. To make this formal, the *conditional Kolmogorov complexity* of s given t, $K(s|t)$, is the minimum length of a program which, when given t on its input tape, generates s on its output tape and halts. If $s = t$, the program just copies its input tape to its output tape, so the program is of constant length; if s and t are unrelated, then the program just ignores t, and $K(s|t) = K(s)$. A simple generalization allows the identification of $K(s|T)$, where T is a set of objects. Of course, conditional Kolmogorov complexity is just as uncomputable as ordinary Kolmogorov complexity.

Given a collection of objects $T = \{t_1, \ldots, t_n\}$, Mondol and Brown argue that if $K(s) \approx K(s|T)$, then s is *novel* with respect to T: the items in T do not help in describing s. Of course, this idea of novelty will be represented as a spectrum; for example, in practice, any English text will help to some degree in compressing any other English text, even if they are not from the same genre at all. Ens and Pasquier (2018) and Takamoto et al. (2016), among other authors, have used this measure to cluster items and identify their style, using general compressors to approximate conditional and absolute Kolmogorov complexity.

Models as typicality

One could use the opposite of novelty to model *typicality*, but Mondol and Brown instead use the concept of a model: given a set $T = \{t_1, \ldots, t_n\}$ of objects, we can build a program P_T, which, when run on given inputs $\{r_1, \ldots, r_n\}$ generates the items of T, with $P_T(r_i) = t_i$ for all i. This is called a *model* of T. Models are a restricted class of Turing machines; one variety of restrictions requires T to be a computable set and P_T to be a Turing machine that halts on all inputs.

If the model is a good one, then for all i, $|P| + |r_i| \approx K(t_i)$, and the members of T are considered *typical* for P. A new object s is also a typical member of the class if there exists a good model Q of $T \cup \{s\}$ such that $|P| \approx |Q|$; that is, learning about the existence of s does not make us have to do much more to accommodate it. A simple example of this phenomenon is that the program $\mathtt{PRINT}()$, which on input r prints r, is a good model for random strings, but a highly repetitive string s would not be "typical" for that class, as $\mathtt{PRINT}()$ is a bad model for such strings, since $K(s) \ll |s| + c$. In algorithmic information theory, this framing may also give a probability distribution over members of the class of outputs of P (Li and Vitányi 2019), and can be used to model properties of the overall class, assuming one has a good model.

Domain-agnostic vs. domain-specific aesthetic

The complexity-theoretic aesthetic measures proposed by Mondol and Brown are *domain-agnostic*. That is, they are concerned with abstract notions of complexity that are independent of the domain to which they are applied, and thus they can in principle be applied to any domain—one can imagine encoding a joke, a recipe, a piece of music, a mathematical theorem, a drug design or a legal code as a string and then using logical depth as an abstract measure of its value. However, as elegant as this is, it clearly does not capture everything there is say about value, when it comes to jokes, recipes, music, theorems, drug design and legal codes. In particular, it does not consider *domain-specific* notions of aesthetic, which do not generalize across domains—jokes should be funny, recipes delicious, music catchy, theorems influential, drug designs effective and legal codes fair.

While there may be general debate about whether creativity is itself domain-agnostic or domain-specific, we argue that it is both, at least as far as aesthetics is concerned.[2] This means that it is critical to determine how to integrate the domain-agnostic with the domain-specific for a unified theory of aesthetic—how do we ground abstract notions of complexity in a specific domain? Specifically here, how do we do so in the context of ethical decision making? One way to think about this is that the domain-specific aesthetics naturally constrain the space of possibilities (it may not be acceptable to choose murder as a conflict resolution, no matter how sophisticated the argument supporting it); within that constrained space, domain-agnostic aesthetics can be used to drive the search. Another paradigm may be that of multi-objective optimization, in which an agent attempts to satisfy (or satisfice) both the domain-agnostic and the domain-specific aesthetic measures.

Complexity-theoretic-based aesthetic in ethics

There are significant challenges with using the Mondol and Brown framework for identifying the quality of a creative artifact. First, and perhaps most distressingly, all measures used in their paper are uncomputable. Moreover, while their novelty metrics are largely just based on conditional Kolmogorov complexity between an object and others from an inspiring set, and can at least be estimated using standard compression algorithms like Lempel-Ziv (Ens and Pasquier 2018), the measures they identify for estimating the value of an object s largely relate to the internal complexity of that object; the only evidence of logical depth or sophistication is the creation of a slow short program whose output is s or a large model that generates s (as well as other objects) as a typical output.

As such, using computational complexity in any aesthetic scenario presents serious difficulties. However, this key challenge, ironically, is one of the strongest arguments in favour of the approach in the ethical domain: it recovers a fallacy often found in real human reasoning.

Charlatans and seemingly random decisions

A real annoyance, both in the real world and in computational artifacts, is claims that an object is serious and significant, when in fact it is arbitrary or random or trivial. This "The Emperor Has No Clothes" phenomenon is a serious risk of Mondol and Brown's formulation of value as sophistication or logical depth. For example, if P is a short program that first churns for $2^{|P|}$ useless steps, and then runs a very fast, very short, program P' whose output is a string x, then x will appear to be of high logical depth if we do not know about the program P'. Because in general it is impossible to know about the effects of a program without running it, programs of this sort are undetectable; indeed, as with the classic parable of the pot roast (in which a cook cuts off the ends of a beef roast before baking it for no reason other than that their parent did the same thing for a pan too small to hold a full roast) (Brunvand 1990), useless work might well be done by a contemporary reasoner because it arose in a benign former context and has never been discarded.

In our ethics framework, when the Emperor has no clothes, one of the objects under study for its aesthetic significance is assessed as having high logical depth or sophistication, by virtue of the long amount of research, study and preparation that has gone into its creation. But if that time has been wasted (by building circular logic, or by producing endless rehashing of the same case, for example, or by simply running a slow algorithm when a fast one might exist), the legal code C, or the decision outcome (O, J) may appear to be deep while not in fact being deep. (We note that detecting this scenario is difficult. For example, imagine if our standard for whether a string is logically deep or not is connected to polynomial runtimes. Then, if $\mathtt{P} = \mathtt{NP}$, there exists a fast compression algorithm for the binary string s_n that indexes graphs in a natural order $G_1, G_2, \ldots, G_n$ and has a 1 in position i iff graph G_i is Hamiltonian, which means that s_n is not logically deep; however, if $\mathtt{P} \neq \mathtt{NP}$, then no such fast compression algorithm exists, and s_n is logically deep.)

A different version of this problem occurs when the object under study was developed by a deliberately misleading agent. Here, the legal code C *appears* to be logically deep or of high sophistication: for example, we might even be able to run the short, slow program and create C with it. Such a program may still engage in some useless reasoning along the way of forming C, inserted by a charlatan who wants to make the code appear more serious than it actually is. Unfortunately, since in general it is hard (or uncomputable) to examine code for shorter or more efficient equivalents, it is also likely difficult to detect whether we have been deceived by a system that appears more complex than it actually is.

A similar problem arises when an extraordinary amount of detailed effort goes into planning how a system will respond to improbable scenarios. The object is *legitimately* logically deep and offers detailed guidance for how to handle the rare situation, summarizing challenging reasoning in a short package. Unfortunately, despite this potentially significant piece of work having been done, the author has hung it on a useless hanger. This situation is perhaps analogous to theological reasoning about the number of angels

[2] We hypothesize that this principle applies to creative process as well. That is, we hypothesize that there exists an abstract "core" creativity algorithm that is domain-agnostic and that can be specialized in domain-specific ways, rather like the notion of inheritance in object-oriented programming. However, we do not present arguments supporting that position here.

that can dance on the head of a pin—if this never observably happens, the system of reasoning is, in the domain-agnostic sense of Kolmogorov complexity, beautiful, yet useless.

Elegant is different than good

In addition to the concerns about seemingly random decisions, nothing stops an ethical system from being fundamentally monstrous except for external constraints pushing the decisions of that ethical system away from those terrible outcomes. In the previous subsection, we considered the case where a system appears sophisticated or logically deep, but is in fact not. However, one can also deploy algorithmic-information theoretic ethics in ways that are logically deep, but where the logical depth yields unhelpful results. For example, imagine a procedure P designed to decide cases about slavery, and outcomes of disputes involving enslaved people. If P is trained on a collection of cases and laws that start out with the presumption that slavery is valid, it might develop into a highly compressed program that encapsulates those cases and laws in an efficient framework. It might even use sophisticated reasoning to assert that one set of slaves should be freed and another subject to further bondage, generalizing from data about their cases and about existing similar cases. As such, P could appear to be typical and of high quality.

Further, P might not be like existing ethical systems in how it works, indicating that it is also of high novelty, in that knowing P does not given much help in building other pre-existing legal interpretation systems. However, none of these metrics—novelty, value, or typicality—pushes P to question the overarching unacceptability of the frame in which it operates. That is, P may be able to simplify, codify, and regularize the cases it decides, but if it starts with the requirement that it maintain the status quo, it may simply build a better evil system. It is unsurprising that this danger exists—it exists precisely due to the dichotomy of domain-agnostic vs. domain-specific notions of aesthetic.

Small changes with big differences

Another unexpected problem with the domain-agnostic measures of value and novelty is that they can push the system to make tiny changes to its texts that may have dramatic overall impacts. For example, suppose that C is a criminal code that identifies the sentences for violating various laws; for simplicity, suppose that C_1 is a function that maps a crime c to *days* in jail $C_1(c)$. The code C_1 is essentially equivalent in complexity to another code C_2 that assigns the same number of *weeks* in jail to c as C_1 assigns days. (That is, $C_2(c) = 7C_1(c)$ for all c.) Yet these are fundamentally different. Similarly, and more alarmingly, imagine that C_1 is a civil code that describes how to identify which party legally owns a piece of property under dispute between two parties. If C_2 is a new civil code that results in exactly the reverse outcomes to that of C_1, then both C_1 and C_2 are essentially equal in all measures of complexity, just as a photograph and its inverse are.

The only way to avoid this problem is via precedent—we must prime the pump with existing case law, and only accept legal codes that are consistent with existing decisions. But this leaves us in the position we were hoping to avoid—novelty comes not through generalizing from existing situations, but by intentionally moving away from what is known.

Not all bad news

This litany of negative news about Kolmogorov complexity-based aesthetic might suggest that the whole endeavour is hopeless, but that is not the case. The fundamental idea still appears sound: a short legal code, or a simple, fast ethical system, which summarizes a large amount of case law in a small, efficiently-presented package, and which allows for the fast resolution of simple cases and a complex reasoning process in difficult cases, is exactly what is needed.

To be more specific, consider a legal question (C, C', D). If D is easily resolved, it should be the case that $K((O, J)|(C, C', D))$ should be small—that is, it should be possible to efficiently fill in the details of a proper judgment given the legal code and commentary, with very little extra information. Creating this extra information is, ultimately, the task of the judge, and the key observation is that if D is a *typical* case for (C, C'), this work of finding a good resolution for it should be efficient. By contrast, if D is an odd edge case, then the judge must perform substantial computation, creating much new information, in computing the outcome (O, J) of the dispute.

Fundamentally, then, an aesthetically appealing ethical system, particularly in our second frame, consists of a concise representation of complex ethical principles with an algorithm for quickly mapping them onto resolutions for dilemmas that commonly arise. Further, novelty search should enable the discovery of both algorithms and principles that, while they encapsulate similar information to those pre-existing, nonetheless use different methods; that is, knowing about existing algorithms should offer minimal capacity to predict a new approach.

Building an ethical system

As in Ventura and Gates, now comes the rub: how do we develop a system whose output is novel, valuable, consistent, transparent, and non-trivial? In no small part because of the challenges described in the previous section, we largely leave this question for future work and analysis.

As one possible avenue of exploration, as briefly suggested by Ventura and Gates, it may be possible to perform large-scale statistical simulations involving agents making decisions using the ethical system under scrutiny. Serendipitously, this is possible exactly because the agents are computational rather than human, and, interestingly, this empirical approach could apply to estimating both the domain-agnostic, information-theoretic aspects of aesthetic as well as the domain-specific, ethics-based aspects. For the former, one may be able to use statistical simulations to estimate information-theoretic measures similar to how Soler-Toscano et al. empirically estimate the algorithmic probability of short strings (2014). For the latter, such simulations may be used to estimate the likelihood of various individual outcomes, to perform differential analysis, or to model large-scale social outcomes, facilitating a comprehensive/empirical description of the system in terms of its effects.

For example, considering the case of real-time ethical decision making, we might construct a simulation of self-driving vehicles encountering ethically challenging scenarios. [3] Driving agents could be equipped with varying ethics systems and large-scale simulations could result in statistical measures of global or local utility (e.g., to estimate fairness). Or, agent behavior patterns could be analyzed for complexity as in (Zenil, Marshall, and Tegnér 2015) (e.g., to estimate logical depth).

For the case of making legal decisions, many of the same kinds of ideas might be applied. For example, a system for drafting traffic laws might hypothesize a traffic code and then perform simulations of varying traffic scenarios governed by that code to verify its generality, its clarity or its fairness statistically. Or, perhaps the complexity of the simulation may act as a proxy for the complexity of the code, in the information-theoretic sense.

The the main difference between the two scenarios is the perspective from which simulations are being run and who is making use of them for what: in the first case, if we are using simulations to evaluate something about the ethical system, it is because we are designing the system and wonder about its utility for the agent; in the second case, the agent itself is constructing the ethical system and is using the simulations as a part of its creative process.

Aesthetics of creating ethical systems

We have identified the creation of ethical systems as a fundamentally creative task, and considered the aesthetics of this task under two quite different formulations: building fast algorithms that find solutions to ethical dilemmas (as well as explanations for those solutions), and building slow algorithms that reason using law codes to find the correct answer to a serious case, and offer detailed reasoning to justify their decisions. We have suggested that aesthetic judgments appear at multiple steps in this process, and in particular, that good design of legal codes can enable efficient decision making and more transparent reasoning.

We also briefly discussed the actual process of searching for such ethical principles. A key issue is that they must not be assessed solely on the basis of novelty, typicality and value as measured by (domain-agnostic) complexity, but (domain specific) characteristics such as fairness and real-world suitability must also be considered; failing to account for the latter creates the possibility of developing ostensibly beautiful philosophical models that are monstrous in the real world. While complexity-theoretic-based aesthetics can play a role in the development of ethical systems, these systems must still generalize from the decisions of extant judgment systems and case law, and they must display straightforward properties (such as consistency, explainability and generalizability) that are found in real-world systems.

Interestingly, this multiple-step process of looking for ethical answers, and then looking for ethical systems, suggests that we could also go out one further level, to the aesthetic analysis of the procedure with which we search for ethical systems. That is, we can have an aesthetics of ethical decisions and an aesthetics of ethical systems, but we can also assess the aesthetic value of the process of building systems that build ethical systems. Much as with the existing dilemmas of this paper, incorporating novelty, value, typicality and feasibility into such an assessment will likely not be an easy task.

Conclusions

In this work, we have looked at the question of ethical decision making and the design of ethical frameworks, to see if computational creativity offers different advice for this process than do other existing frameworks. We used the frame of Ventura and Gates, who first proposed this aesthetic understanding of the process of finding good ethical principles, and combined it with the domain-agnostic aesthetic value measures of Mondol and Brown, which focus on conditional computational complexity and efficiency of summarization as measures of novelty and value. We argue that we might use such an approach to aesthetics on both the process of making ethical decisions and the process of designing ethical systems, but in practice the challenges with computing these measures, and the potential that a decision maker might build something ostensibly aesthetically beautiful, but in practice monstrous, still remain. Putting this whole approach into practice will require much further work.

We note, finally, that while our motivation for considering these questions has been the question of developing ethical systems and the computational creativity of this question, there is ultimately nothing fundamentally ethics-based about many of our arguments; the same types of arguments likely hold for developing computationally creative systems that parent children or train pets, that make theological arguments, or that otherwise generalize reasoning from a large case set, or make quick decisions. We look forward to both generalizing these results and finding ways to make them more practical.

Acknowledgements

The research of D.B. is supported by an NSERC Discovery Grant.

Author Contributions

Both authors contributed to all aspects of the work, including ideation, narrative/position development and writing.

References

Bem, D. J. 1972. Self-perception theory. In Berkowitz, L., ed., *Advances in Experimental Social Psychology*, volume 6. New York: Academic Press. 1–62.

Bennett, C. H. 1988. Logical depth and physical complexity. In Herken, R., ed., *The Universal Turing Machine: a Half-Century Survey*. Oxford University Press. 227–257.

Brunvand, J. H. 1990. *Curses! Broiled Again!* W.W. Norton.

[3] For example, something like this: https://www.moralmachine.net

Cropley, D. H.; Kaufman, J. C.; and Cropley, A. J. 2008. Malevolent creativity: A functional model of creativity in terrorism and crime. *Creativity Research Journal* 20(2):105–115.

Ens, J., and Pasquier, P. 2018. CAEMSI : A cross-domain analytic evaluation methodology for style imitation. In *Proceedings of the Ninth International Conference on Computational Creativity, Salamanca, Spain, June 25-29, 2018*, 64–71.

Kaelbling, L. P.; Littman, M. L.; and Cassandra, A. R. 1998. Planning and acting in partially observable stochastic domains. *Artificial Intelligence* 101(1):99–134.

Li, M., and Vitányi, P. M. 2019. *An Introduction to Kolmogorov Complexity and Its Applications*. Springer Publishing Company, Incorporated, 4th edition.

Mondol, T., and Brown, D. 2021. Incorporating algorithmic information theory into fundamental concepts of computational creativity. In *Proceedings of the International Conference on Computational Creativity*, 173–181.

Simon, H. A. 1955. A behavioral model of rational choice. *The Quarterly Journal of Economics* 69(1):99–118.

Soler-Toscano, F.; Zenil, H.; Delahaye, J. P.; and Gauvrit, N. 2014. Calculating kolmogorov complexity from the output frequency distributions of small turing machines. *PLoS ONE* 9(5):e96223.

Takamoto, A.; Umemura, M.; Yoshida, M.; and Umemura, K. 2016. Improving compression based dissimilarity measure for music score analysis. In *2016 International Conference On Advanced Informatics: Concepts, Theory And Application (ICAICTA)*, 1–5.

Thomson, J. J. 2014. Killing, Letting Die, and The Trolley Problem. *The Monist* 59(2):204–217.

Ventura, D., and Gates, D. 2018. Ethics as aesthetic: a computational creativity approach to ethical behavior. In *Proceedings of the International Conference on Computational Creativity*, 185–191.

Zenil, H.; Marshall, J. A.; and Tegnér, J. 2015. Approximations of algorithmic and structural complexity validate cognitive-behavioural experimental results. `https://arxiv.org/abs/1509.06338`.