Incorporating Algorithmic Information Theory into Fundamental Concepts of Computational Creativity

Tiasa Mondol
tiasa.ap.10@gmail.com

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

Abstract

Can we attribute creativity to an artifact by examining its computational history? What can be said about its value and novelty if the artifact is computationally laborious and interesting? Can the computation which gave rise to the artifact, help us interpret its artist? We look at these questions through the lens of Algorithmic Information Theory. Expanding on some of the advanced topics in this field: Kolmogorov Complexity and its conditional and resource-bounded versions, Logical Depth and Sophistication, we show that many of the standard criteria used in computational creativity follow naturally from these concepts. We also address the question of whether an artifact is a typical or novel creation of its artist, by first generating a good description of the artist's known works and then examining how fit this explanation is for the new artifact. Although using Kolmogorov Complexity is not without its challenges, its incomputability being the obvious one, we show that by rooting our analysis into the algorithmic complexity of an artifact we inevitably shed light on some of the fundamental concepts of creativity: typicality, novelty, and value of an artifact, the creative process, and the creator.

Introduction

A key question since the rise of computational creativity has been which criteria characterize an artifact as creative, and how an algorithm can evaluate these criteria (Lamb, Brown, and Clarke 2018). In this work, we use the lens of *algorithmic information theory* (Li and Vitányi 2019) to look for universal properties that indicate whether a work is of high quality or value, novel or typical within a domain or a creator's oeuvre. These commonly used criteria for creativity follow smoothly from several advanced concepts in algorithmic information theory. Quality is assessed by showing that there is a high amount of computational effort required by any short program whose output is a given artifact, while typicality and novelty are shown by looking for a program that can generate all members of a class of objects and assessing how random a new object is within that class. Another outcome of our work is that we can model the producer of an artifact by distilling down its core properties. Then based on the complexity and volume of these properties we can identify whether the artifact is a magnum opus of its artist.

Key to our work is a firm computational underpinning to some of the canonical ideas of creativity and to build it we carry out a more thorough exploration of Kolmogorov complexity and its adjunct ideas. In previous work, numerous authors have attempted to use two of the most basic ideas, the raw Kolmogorov complexity $K(x)$, and the conditional Kolmogorov complexity $K(x|y)$ of object x given y as a free input, as measures of the creative value of an object or of the similarities between two objects (Li and Sleep 2004; Ens and Pasquier 2018). While these measures are certainly important, their use in this manner is still potentially of concern: an object with low Kolmogorov complexity might or might not be trivial, and objects whose Kolmogorov complexity are close to their length (which is the maximum possible) are typically just random noise. Even looking for an object of "medium" complexity is insufficient: such an object might either be the product of substantial computational effort (which, in our telling, makes it a high-quality work) or could be trivial repeated patterns augmented with random bits to make it seem serious. Only by looking at the actual computational effort required to produce the object can we assess its quality; the raw value of $K(x)$ is insufficient. Similarly, while the conditional complexity $K(x|y)$ is important, more useful is to look at $K(x|M)$ where M is a model representing the non-randomness in x. Then a good model for x will be most successful in compressing x and classifying it into a category that is most appropriate.

Here, we begin the build out of this theory of creativity of computationally created objects and how to assess them. We begin with an introduction to algorithmic information theory, where we focus on the concept of *models* that gives a two-part description for members of a genre or a creator's oeuvre and how typicality or novelty can be assessed for an artifact. The concept of *logical depth* allows us to assess the computational effort required to produce an object: if an object with high compressibility has slow-running short programs, the most likely explanation for the object is that it required substantial effort on the creator's part to produce it, and it is valuable. Additionally, the concept of *sophistication* allows us to identify the inherent challenge of representing the producer of an object: if the only good models for an object are of high sophistication then it must have been created by a skilled creator. These concepts have not previously been extended to the domain of computational creativity. At the end of this paper, we describe the relationship between these concepts to existing efforts in joining algorithmic information theory with computational creativity and aesthetics; we also give, in Table 1, an algorithmic recipe for our own approach which follows naturally from the theoretical concepts presented in this paper. Although it is necessary to navigate the subsequent sections carefully to fully understand the ideas in this table and our critique of existing approaches, interested readers are encouraged to consult it for a helpful summary of the topics we present hereafter.

Kolmogorov Complexity and The Artifact

We begin by building a formal description method for a creative product by associating it with a Turing-computable function. Although it is difficult to regard an artifact like "The Mona Lisa" to be an output of something as inanimate as a function, such formulation provides us with a powerful theoretical framework to analyse an object computationally. We work with Turing machines[1] that compute partial recursive functions, which are only defined for some inputs and for which an effective algorithm with step-by-step instructions exists to compute it. The Turing machine upon receiving an input, computes the function on that input by manipulating the bits present on its tapes and halts with output x or runs forever if the input is undefined. Thus a finite binary program which encodes the Turing machine along with the input on which it halts with output x completes a description of the target creative product x. It is important to note here that when talking about generating a creative product on a Turing machine's tape, we are essentially reducing the vibrant object to a binary representation which may be a lossy depiction of the original product. Some creative objects are also a sum of their spatial and temporal contexts and cannot be perceived in isolation. For these cases, we assume that the target output x is a recognizable version of the original product which can be recovered from x within reasonable time bounds: for example, to generate a 2-dimensional $m \times n$ painting, we can imagine a Turing machine outputting the RGB values of its each pixel. More important for us is to maintain an objective notion of computability and description, as provided by the Turing machine model.

We can enumerate the programs encoding Turing machines lexicographically by their increasing length (Li and Vitányi 2019, pp. 27-33) and such enumeration contains the shortest program to produce a target object x and will be the basis for defining its Kolmogorov Complexity. To run these programs, we employ an additively optimal Universal Turing machine (UTM) which accepts an optimal encoding of the Turing machine in the form of a program p, simulates it using optimal space and time and acts as a formal standard with which we compute an object's complexity (Li and Vitányi 2019, pp. 103-107). Thus, the Kolmogorov or descriptional complexity $K(x)$ of an object x is the length of the shortest program p that when run on a Universal Turing machine U terminates with the output x.

$$K(x) = \min_p\{|p| : U(p) = x\}$$

It is also desired that the UTM U upon reading exactly p from its input tape, will read no further and halt with output x. This setup is known as prefix-free coding where no program is a proper prefix of another. It lets us uniquely decode p to only one object and avoids ambiguous situations where both pq and p are valid programs. It also alludes to the algorithmic or universal a priori probability of an object: if the UTM U terminates with output x on $|p|$-length prefix-free inputs generated by fair coin toss, the probability of x's existence, also known as x's universal a priori probability is

$$Q_U(x) = \sum_{p:U(p)=x} 2^{-|p|}$$

By being the shortest, $K(x)$ contributes $2^{-K(x)}$, larger than any other program, to $Q_U(x)$ and thus is regarded to be the most probable causal source for the creative product x.

The Kolmogorov Complexity along with all its variants we shall present in this paper are incomputable; there is no function that computes shortest descriptions for all objects. A simple paradigm like: "Supply the UTM with inputs by lexicographically increasing length and the first program that halts with output x is its shortest program" does not work as it is not decidable which program halts and the computation can go on forever without any meaningful progress. While this might be disappointing, the Kolmogorov Complexity can be reasonably approximated (Vitányi 2020) and its incomputability is perhaps a reminder of the elusive and enigmatic nature of the creative phenomenon.

Conditional Kolmogorov Complexity, Two-Part Code and Models

It is often more useful to view one creative product in the light of another, like learning to describe Vermeer's "Girl with the Red Hat" painting while we already know about his "Girl with a Pearl Earring". It also illustrates a natural way of describing objects with a predisposition to prior knowledge or inductive bias. We formalize this notion with the conditional Kolmogorov Complexity $K(x|y)$, which is the shortest program to produce x on U when U is pre-furnished with object y. This generalizes the unconditional definition as $K(x) = K(x|\epsilon)$ and is more useful in understanding x's structure: if $K(x|y) \leq K(x)$, then x and y share commonalities and $K(x|y)$ represents the amount of idiosyncratic information left in x.

If the provided information y encodes nonrandom regularities that we recognise and associate with a class or category, then based on the $K(x|y)$ we can deduce whether or not x belongs to this class. For example, the "Girl with a Pearl Earring" belongs to a special class of paintings called "Tronie" which features unidentified subjects displaying exotic facial expressions or garments (Schütz 2019). Then the idea of a two-part code lets us decompose the painting into meaningful information- the part that makes it a tronie and individual randomness- the part that separates it from other tronies.

[1]We use Turing machines with a read-only input tape, one or more (a finite number) work tapes at which the computation takes place, and a write-only output tape. All tapes are one-way infinite, divided into squares,and each square can contain a symbol from a given alphabet $\{0, 1\}$ or blanks (Li and Vitányi 2019, pp. 27-33)

Formally, the shortest effective description of x can be expressed in terms of the length of a two-part code, the first part $K(M)$ describing an appropriate Model computed by a Turing machine T and the second part $K(x|M)$ describing the left-out irregularities or random aspects of x after M squeezes out its regularities:

$$K(x) = \min_{M}\{K(M) + K(x|M)\}$$

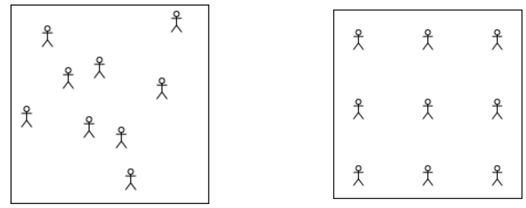
Here *Model* is a hypernym used to quantify the regularities in sets of objects and with which we recognize the regularities in x. The best model M encapsulates the useful or compressible information in x, while minimizing the total description length. In relevant research (Vereshchagin and Vitanyi 2004; Gacs et al. 2001), analysis has been mostly done with M denoting a finite set of objects. However, a model can also be a total recursive (Koppel 1995) or probability density function (Gacs et al. 2001), each formulation having its own properties and relevance. To start, let M be $\{x_1, x_2, \ldots, x_m\}$, signifying a history of observed phenomena. The cost of reconstructing an object x from this M on a UTM U comprises of a short program of length $K(M)$ to enumerate the set, while another $\log|M|$ bits to locate x in the set. In this setup, the shortest two-part code $K(M)+\log|M|$ can be larger than $K(x)$, even when $K(M)$ is small. Hence, a complexity restriction is imposed on $M : K(M) \leq \alpha,\ \alpha \in \mathbb{N}\setminus\{0\}$ so M may no longer describe a set losslessly and only captures its essence by exploiting the shared information among the objects. To illustrate this with an example: let $\{x_1, x_2, \ldots, x_m\}$ be different paintings of "crowded field". To transmit one of these through a channel with limited capacity α, one can transmit the indication that the painting is of a crowded field and the particular positions of people may be chosen by the receiver at random.

If we formulate M as a total recursive function, the two-part code for x becomes $K(M) + |d|$, where $M(d) = x$ and $K(M)$ is the shortest program length that computes the function M. This is a more intuitive interpretation of models as M now can mimic a generator of objects which has embedded in them the structure that M signifies. This is also the foundation of the idea, *sophistication* which we discuss in more detail in a later section.

Model-fitness and Randomness Deficiency

The finite models $\{M_1, M_2, \ldots\}$ that satisfy the constraints $K(M_i) \leq \alpha$ and $K(x) \leq K(M_i) + \log|M_i| + O(1)$ are called algorithmic sufficient statistics (Gacs et al. 2001). These models allow description of x with only a small increase in complexity, and with a short-to-describe model. The task still remains to choose the one among the candidates $\{M_i\}$ that *best-fits* x and for that we look at the second element of the two-part code.

Recall that in the event where M denotes a set, it takes about $\log|M|$ bits to locate any $x \in M$. This amount is called the data-to-model code and is different from $K(x|M)$. Since $K(x|M)$ is the smallest program outputting x given M, it leverages M and x in a way that could be much smaller than just specifying an index. Going back to the paintings of "crowded field": if the people in x are ordered in a specific way, like a "military parade" then $K(x|M)$ could be much less than $\log|M|$.



(a) (b)

Figure 1: The crowd in figure 1a is more random than 1b

In this circumstance, x is not a *typical* element of M. If it were, then any randomly selected painting from M would be indistinguishable from x except for irrelevant details. The difference between $K(x|M)$ and $\log|M|$ is quantified by its randomness deficiency $\delta(x|M) = \log|M| - K(x|M)$. An object x is typical of M only when we can not significantly improve the conditional description of x given M than specifying its index in M, that is $\delta(x|M) \approx 0$. If M is a total recursive function which on some input d generates x, then M's fitness for x depends on the number of bits $|d|$ needed to indicate the input. Mathematically, if $K(M) + l_x(M) \leq K(x) + O(1)$ where $l_x(M) = \min\{|d| : M(d) = x\}$, then the randomness deficiency of x w.r.t. M is $\delta(x|M) = l_x(M) - K(x|M)$.

Typicality and Novelty

We shall now apply the notions of Model and Randomness Deficiency as discussed above to address the "typicality" and "novelty" of a creative product. Ritchie (2007) included "novelty" as one of the essential properties for assessing a product of a computer program exhibiting creativity. But arguing "novelty" or "originality" has an anthropocentric element to them, he added the property of "typicality" to measure the extent to which a produced item is an example of an artifact class. This section refines Ritchie's argument and formalizes a method for estimating "typicality" of an object. In addition, we discuss how "novelty" can be recognized with the help of a data-to-model code.

Previously, McGregor (2007) addressed this idea by defining the novelty of an object by looking at its information distance (Bennett et al. 1998) from each of a collection of objects of the same class, and choosing the minimum of these distances as the novelty of the new object. In the same paper, the author critiqued the idea by pointing out that the observer estimating novelty needs to have a perceptual frame in which to work. This approach can be fleshed out using the two-part code we use in this paper, and also by considering how computational agents update their perceptual model of the object by interacting with each other and with new artifacts; this latter subject is one we are currently exploring.

The question of how typical a creative object x is, with respect to a composer's oeuvre or a even broader artifact class, is really understood by its randomness deficiency $\delta(x|M)$. Here M models the apriori regularities or recognized properties of a representative set $S = \{x_1, x_2, \ldots, x_n\}$, which, following Ritchie (2007) we call the "inspiring set". We express M as a total recursive function such that there exist parameters $\{d_i\} : M(d_i) = x_i$ for $1 \leq i \leq m$. Let $l_x(M)$ denote the length of the first parameter d on which M halts with the artifact x. Thus as $\delta(x|M) = l_x(M) - K(x|M)$ decreases, x becomes more typical for M. Note that based on S, the typicality that x exhibits can be akin to either *H*-creativity (producing an idea/artifact which is wholly novel within the culture, not just the creator's oeuvre) or *P*-creativity (producing an idea/artifact which is original as far as the creator is concerned) (Boden 1991). However the analysis we present here is effective for understanding both, if S contains members that are consistent with the class that we are interested in. Then we have the following definition of typicality.

Definition 1 *Let M be a total recursive function and a minimal sufficient model of the inspiring set $S = \{x_1, x_2, \ldots, x_m\}$ such that there exist parameters $\{d_i\}$: $M(d_i) = x_i$ and $K(M) + |d_i| \leq K(x_i) + O(1)$ for $1 \leq i \leq m$. Then the typicality of an object x with respect to this model is as following.*

$$typicality(x|M) = -\delta(x|M) = K(x|M) - l_x(M)$$

When x is not in M's range, that is $\forall d : M(d) \neq x$, then $l_x(M) = \infty$ and we get the lowest typicality $-\infty$. If typicality is close to the maximum value 0, then there are no simple special properties that single x out from the majority of elements in S. Otherwise, we can pick a special subset Q of M, which has only the members with this property (like ordered positions of people in the "crowded field"), then x will be much more typical for Q, than it is for M.

Intuitively, a "novel" object with respect to a model should have high randomness-deficiency. But this property alone is not sufficient, as to define "novelty" we need a notion of unexpected or unique outcome of the corresponding model. The central motivation in our discussion has been to find the true source M that produced the object at hand. But suppose the true source is 100 coin flips and our data is $111 \ldots 1$. A model that identifies with flipping a fair coin as the cause of the data, is surely a bad model. However, in real-world problems, such as modeling creative products, the data can be just atypical or accidental for the model that actually produced it. In this case, the model might still describe the regularities in the object, but the extraordinary conditions or the data-to-model code that caused the model to output it, sets the object apart from other objects created by the same model. We liken a novel artifact to such data: it is an unlikely and original outcome of a model M that is a minimum sufficient statistic for the inspiring set $\{x_1, \ldots, x_m\}$ and there exist parameters $d_i, d : M(d_i) = x_i,\ K(M) + |d_i| \leq K(x_i) + O(1)$ for $1 \leq i \leq m$ and $M(d) = x$. The artifact x is still producible from model M, but the remaining distinctiveness d in x is an indicator that there are better models to produce x (consider the model M accidentally getting a set of ordered locations to place people in the "crowded field" example when location are being randomly generated). This unfitness is captured by its reduced kinship with other artifacts $\{x_i\}$ for which M *is* a model. Since, these objects are results of inputs $\{d_i\}$ and d to the model M which is the common denominator between the inspiring set and x, the extent to which x is novel is determined by how much information in shared between $\{d_i\}$ and d. Novelty of x is then essentially captured by the mutual information $I(\{d_i\} : d) = K(d) - K(d|\{d_i\})$ between d and $\{d_i\}$ (Li and Vitányi 2019, p. 249). The less $\{d_i\}$ informs of d, the more novel x is, reaching maximum at $I(\{d_i\} : d) = 0$. This could implicitly mean that the randomness deficiency $\delta(x|M)$ is large. But $I(\{d_i\} : d)$ measures the difficulty with which we figure this unfitness out even with the available information.

Definition 2 *Let M be a total recursive function and a minimal sufficient model of the inspiring set $S = \{x_1, x_2, \ldots, x_m\}$ such that there exist parameters $\{d_i\}$: $M(d_i) = x_i$ and $K(M) + |d_i| \leq K(x_i) + O(1)$ for $1 \leq i \leq m$. Then the novelty of an artifact x, producible from M with parameter $d : M(d) = x$, is*

$$\textit{novelty}\ (x|M) = -I(\{d_i\} : d) = K(d|\{d_i\}) - K(d)$$

Novelty thus can be compared to explorations undertaken by an artist or changes in a genre while not affecting their creative styles: learning and applying new techniques, exposure to new environments and influences, can bring about novel objects that capture the general spirit of the inspiring set but are difficult to be recognized in their light.

It is important to note here the distinction between a two-part-code (model and inputs) identified by an observer and the actual computation process followed by a creator. The observer may be updating an already-existing model based on new examples, or may in fact not be capable of building a computation structure as sophisticated as the one the creator used; for example, for a logically deep object (defined below as an object needing long computation by a short program), the observer may not even have time to run all of the steps the creator used, while updating or building a two-part code for the object. This takes us to the discussion of time-bounded complexity analysis and modelling the creator, which we discuss in detail in the next few sections.

Logical and Computational Depth

We now take into account the difficulty or resource with which p outputs x or transforms another object y into x. Indeed, the shortest program $p^{t,s}$ which computes some object x in a bounded time t and space s can be significantly larger than the shortest program p that has access to unlimited resources. Note that, programs using restricted time are more interesting to analyse than programs with restricted (polynomial) space but unlimited time, as they still can solve the hardest of problems (think about the program that generates the core ideas in this paper with only 8-pages to work with) (Li and Vitányi 2019, pp. 37-39), and it also helps quantify the number of steps taken by the program. Hence, we focus on programs p^t that generate a target object x within a bounded time t, while not being too inefficient in their use of space. The time-bounded Kolmogorov complexity $K^t(x)$ is then defined by

$$K^t(x) = \min_p\{|p| : U(p) = x \text{ in at most } t \text{ steps.}\}$$

If t is a shorter time span than the original time taken by the shortest program for x, then $K^t(x)$ suffers from $K^t(x) - K(x)$ redundancy or adhocness, that is $K^t(x)$ may have to store some information about x without meaningfully compressing it. This difference is known as the Computational Depth of x (Antunes et al. 2006).

$$cdepth_t(x) = K^t(x) - K(x)$$

As t grows, excluding the pathological cases (programs doing unnecessary computations), the non-randomness in x gets disguised by complicated manipulations or computations by the program. Bennett (1988) thus calls the time taken by the shortest of programs for producing x its Logical Depth.

$$ldepth_b = \min\{time(p) : U(p) = x \text{ and } |p| \leq K(x) + b\}$$

The minimum is taken among the available candidates to avoid selecting programs that despite producing the desired object, do so inefficiently, whereas other similar-length programs are faster. The term b, called the significance level of $ldepth_b(x)$, calibrates the added length and assigns an importance or confidence measure to the program: as b gets smaller, the program that witnesses $ldepth_t(x)$ becomes more likely to be the actual program that generates x. Thus the number of steps taken by this program is equally probable to be the time needed by x to evolve from its short description.

Value as Computational Effort

We now make a case for Logical Depth as a formal measure of value. We propose that what makes a creative object valuable is not its information content, but rather the amount of mathematical or creative work it relieves its receiver from repeating, which was plausibly done by its originator. A sequence that represents the outcomes of n coin tosses, has high information content but little value. Conversely, a book on algebra may list a number of difficult theorems, but has very low Kolmogorov Complexity since all the theorems are derivable from the initial few definitions and axioms. However, such derivations can be time-consuming and if we transmit only a short description containing the theorems of the book, a receiver has to spend a long time to reconstruct their proofs. Sending the entire book does not increase the information content transmitted, but now the receiver has all the useful information readily available. Thus value of an object does not depend on its absolutely unpredictable parts (information content), nor on its obvious redundancy (verbatim repetitions, sequence of 1's), but rather on what might be called its buried redundancy— parts reproducible only with difficulty, things the receiver could in principle have figured out on their own, but only at considerable cost in resources or computation (Bennett 1988).

This approach to value is obviously only about the object itself, not about its cultural significance, its ability to be understood by viewers, or any other social properties. Yet, despite this limitation, we are not the first authors to make this connection between the value of an object and its buried computational value; Vidal and Delahaye (2019), in particular, has cited exactly this same quantity in their proposal of an ethical mandate to protect artifacts that contain computational significance of the same sort.

A delightful example of a logically deep object is the characteristic sequence of the diagonal halting problem, χ, where each bit $\chi[i]$ is 1 iff the ith program halts. Despite its apparent importance, the n-bit prefix $\chi_n = \chi[0 \dots n-1]$ of χ is highly redundant with $K(\chi_n) = \log n + O(1)$. The intuition is we only need to specify the *number* of indices that contain 1. Once this $\log n$ number is known we can dovetail all the programs $p_0, \dots, p_{n-1}$ (Li and Vitányi 2019, p. 181) on an UTM and stop the computation once the desired number of programs have halted. Yet, this is computationally very expensive, taking at least as much time as the slowest program in the above enumeration.

A logically shallow object, on the other hand, has a fast-running program that is highly probable to be its source. Note that, defining logical depth as the runtime of the shortest program x^* does not constitute a stable definition since there might be a program of just a few more bits using substantially less time to generate x. A complex artwork may have a slow-running short program that is not much shorter than the print program that outputs the artwork literally.

$$K(x) \leq |p_{print}(x)| \leq n + O(1)$$

Here, the object lacks internal redundancy that could be exploited to encode them concisely and is logically shallow, as a print program which generates the object quickly, is almost as probable to be its origin as the shortest one. In a contrasting scenario: a painting that appears complex to its observer may have a short program to generate it. Colton (2008)'s "Art Exhibition: Dots 2008" describes a similar case where a painting of some random dots on the canvas is given two plausible explanations. One, "the dots are randomly arranged" and two, "the dots represent some friends of the artist and the colors and positions convey the artist's feelings about them". To a spectator, who has had a bounded time t to analyse the painting, the first explanation may seem most plausible in the form of time-bounded Kolmogorov complexity, $K^t(x)$. But knowing that there is even a shorter, if slower to execute description of the painting, $p_{artist} : K(x) \leq |p_{artist}| + O(1)$ available, the spectator is more inclined to accept p_{artist} as its most probable origin and assign to it a value as the computational effort $time(p_{artist})$ suggests. Thus:

Definition 3 *The value or quality of a creative product x is the minimum computational effort or time needed to produce it from a b-significant shortest description.*

$$value_b(x) = \{time(p) : U(p) = x \text{ and } |p| \leq K(x) + b\}$$

However, as we write about computation that is constrained by runtime or space, it is also worth considering certain complex artifacts that may have a short description but the only way to reproduce them from the description would require unbounded resources. As such, there is no other way to specify them than to spell them out bit by bit and if a creator were to claim the existence of such an artifact, no effective programmatic verification of this fact would be possible. The situation for such objects, as with Colton's Dots 2008 exhibit, is complicated: a Turing machine that knew about the personal relationships in Colton's artist's life might be able to much more sharply compress those paintings. But without the access to one, the observer is burdened with the verification of the artist's claim.

The Creative Process and its Non-randomness

We inevitably arrive at the questions of what constitutes an effective generation process for an artifact and in the presence of multiple plausible theories, which process is the most likely to have occurred. A fraudster or charlatan may claim that a complex-looking creative product is a result of a slow-running short program, thus artificially inflating its value. But, if the object has an equal-length fast program which involves no random steps in generating the object, then it is equally likely that the latter is the true generative process for the object; and if the object *can* be produced by taking a small number of non-random steps, then it is certainly possible that the fraudster program takes unnecessary pathological steps in order to seem serious.

This non-random non-trivial effort to generate an artifact is also stored in its subjective organization: Beethoven's "Für Elise" is aesthetically pleasing because it is organized in a certain way. If we rearrange its notes randomly to make different musical pieces, only a handful among the vast majority of resulting pieces will be deemed musically pleasing and perhaps only one will be as musically valuable as "Für Elise". Similar idea appears in the fiction of Borges (1998), "The Library of Babel", which describes a library all possible 410-page books on a 25-letter alphabet, and the librarians' attempts to discern which of the books were meaningful. Thus, the inherent organization of these products hold clue to the non-trivial and laborious processes that resulted in their existence (Bennett 1988).

Hence, the most plausible creative process is carried out by a program that is no more than b bits longer than the shortest and from which the work to reproduce x involves no unnecessary, ad-hoc assumptions except for the b bit redundancy. This necessary and non-random workflow for the production of x is what we assign credence to (recall the preference of the art-lover in Colton 2008's dots exhibit). If a short program p has a slow deductive reasoning process, it is not evidence against the plausibility of this program. In fact, if the product has no comparably concise programs to compute it quickly, it is evidence of the non-triviality of the generative process. A great work of autobiography is one example of this: if we just consider the written text as its acceptable representation, then its information content is really low (Shannon 1951). But the existence of such literature stands evidence of a profoundly-led life by the author and the significance of the events that happened in that lifetime.

Definition 4 *A b-significant creative process of a product x is simulated by the UTM upon input p such that $|p| \leq K(x) + b$, $U(p) = x$ and p takes the minimal non-random steps among all b-incompressible programs for x.*

Is it possible to convert a shallow object like a random string to something deep like the Tolstoy's "War and Peace"? Satisfyingly, this is answered in the negative by the slow growth property: a fast deterministic process is unable to transform a shallow object into a deep one, and that fast probabilistic processes can do so only with small probability (Bennett 1988).

Sophistication

We formalize a notion of "structure" or "projectable properties" in an object. In the previous sections, we have mentioned that the complexity and usable information in an object do not have a causal relationship; in fact they may very well be orthogonal properties of an object (Koppel 1995). We thus try to decouple the part of an object that is an aggregate of shareable properties from its accidental information with a two-part code. Earlier, when discussing models, we introduced total recursive function as a way of describing a set of objects with respect to which we examined x's regularities. Here we use a total recursive function slightly differently, to capture the structural information in an individual object x that shows evidence of some planning that went into x's generation. The utility of such formulation becomes evident through the following example: consider a total function $double(x)$ that on any input doubles the bits, $double(011) = 001111$, $double(101) = 110011$. Such properties are difficult to find in general, but work as an excellent compression scheme for the product. Hence the sophisticated part of an object is the size of such programs which stands for the non-random structure of an object.

The c-sophistication of an object is thus defined as follows (Koppel 1995).

$$soph_c(x) = \min\{|p| : p \text{ is total, a parameterization } d \text{ exists for which } U(p, d) = x \text{ and } |p| + |d| \leq K(x) + c\}$$

That is, x is sophisticated if the best model for x is a comparatively long program, not something as simple as the *double* function. The size of an optimal total recursive function along with the data may be c longer than x's shortest description. But in order to reduce c if we furnish p with properties that are accidental or exclusive to x, then it might fail to recognize objects that are similar or generated by the same source. Mondol (2020) demonstrated an example in symbolic-music-compression using context-free patterns. In order to capture core-properties of a music-piece, the author only considers the patterns that repeat most in the corresponding composer's oeuvre. This way the emphasis is put more on the inherent composition techniques of the sequence rather its distinct embellishments. The significance parameter c is thus interpreted as a confirmation of the description (p, d) before regarding extra structure represented by a longer program p'.

Attributes of the Creator

Sophistication is a natural way to measure how much information of an object we can throw away without losing the ability to query its properties (without false positives). We propose that such properties are also a measure of the creator's attributes embedded into the product's structure. Suppose that a music generator whose inner mechanism is unknown to the observer is broadcasting self-composed music.

If the composition obeys some simple rule such as repeating the same patterns or sounds maximally random, then we would not attribute craftsmanship to the source. If however, the composition exhibits complex structure, which is only possible through rigorous planning and meaningful exploration, we might suspect the existence of a skilled creator. Hence, sophistication is that quality of an object that sets apart the artist's talent from their fanciful impulses.

Definition 5 *The style or signature of a creator inherent in a creative product x is measured by the object's c-significant sophistication $soph_c(x)$. That is, we can say that the generator program p of length $soph_c(x)$ produced x on input d leaving out all but c bits of redundancy.*

It is worthwhile to note that both logical depth and sophistication are measures of meaningful complexity in an object; but while one uses dynamic resources (program time), the other uses static ones (program size). Thus the two measures are not necessarily correlated (Antunes and Fortnow 2009): the halting sequence χ is logically deep, but has low sophistication ($O(\log n)$). Rather, logical depth can be used as a structure finding mechanism. We formalize this with the converging hypotheses argument (Koppel 1995): consider the same music generator as above; as we observe more of its music, more structure becomes apparent, thus forcing revisions of hypotheses as to the generator's structure. Let x denote the complete music and x^n its observable prefix. At step n, we would like to find the hypothesized generator p_n and its parameterization $d_n : p_n(d_n) = x^n$ and $|p_n| + |d_n| \leq K(x^n) + c$. If for the previous hypothesis p_{n-1}, we find an input $d_{n-1} : p_{n-1}(d_{n-1}) = x^n, |p_{n-1}| + |d_{n-1}| \leq K(x^n) + c$ then we move onto a larger prefix without updating the hypothesis. Otherwise, we exhaustively search through all programs (not necessarily total) $p : |p| \leq n$ and data $d : |d| \leq n - |p|$ in order of increasing length and let them run for $ldepth_c(x^n)$ steps. We choose the shortest $p = p_n : |p| + |d| \leq K(x^n) + c$ that satisfies these criteria. Thus, in our model, as more composed music is observed, previous hypothesized generators are abandoned for one of two reasons. The most straightforward reason is that the subsequent parts of the composition is inconsistent (do not fall in the generator's range). In this case, the program is changed in favor of one which is less powerful (shorter, using longer data). The other reason for abandoning a program is that as more parts of the composition is observed, structure becomes apparent which was not previously so- that is, use of a more powerful, longer program results in a shorter description when including the required input to generate x^n. Such procedure might not give the smallest compression program for the music generator itself, but it increasingly describes the properties of initial segments of its generated music x, which can be used to compress the larger initial segments increasingly better as $n \to \infty$.

Non stochastic Objects (Masterpieces)

Finally, we discuss another remarkable outcome of the notion of sophistication: absolutely non-stochastic objects, whose complexity is mostly comprised of non-random structure (high sophistication and useful properties) and show that creative masterpieces fall into this category.

An absolutely non-stochastic object has neither minimal nor maximal complexity. Hence they are not typical outcomes of any total recursive program that exhibits low structure. Additionally, non-stochastic objects have no optimal programs that are of relatively small complexity; that is they exhibit high randomness deficiency or atypicality for a program $p : K(p) < K(x)$. Rather, these objects are typical outputs only of programs p that have complexity close to their own, $K(p) \geq K(x)$, indicating high sophistication (Gacs et al. 2001). The program part p of such an artifact thus showcases the creator's elaborate techniques, contents and creative properties that can be pioneering and replicated. When a non-stochastic object is an output of a unique highly sophisticated program, it depicts innovation; similar to the transformative effects of a masterpiece, it has the ability to push a medium or genre to new directions. Thus absolute non-stochasiticity is a pre-cursor to creative masterpieces.

Definition 6 *A creative masterpiece x is absolutely non-stochastic or highly sophisticated, that is, they exhibit low randomness deficiency (needing small additional data) only for total recursive programs p that have complexity close to their own, $K(p) \geq K(x)$. For programs p with $K(p) < K(x)$ they will either require large additional data or they will not be in those programs' range at all.*

If we were to partition a creative masterpiece into meaningful complexity and random noise, we will find that almost all of its complexity comes from useful incompressible properties. Moreover, the amount of such non-randomness is also significant. Thus non-stochastic artifacts reside in a Goldilocks complexity zone: Shen (1983) showed that these objects have complexity at least $K(x) \geq \frac{n}{2} - O(\log n)$. Thus a masterpiece, which is a product of its generative program p having relatively high complexity while being absolutely non-random, is an extremely rare phenomenon.

Re-examining Computational Aesthetics

With the computational tools we now have available in our armament, we revisit some of the existing attempts to incorporate algorithmic information theory into evaluating creativity. Among the more recent papers, Ens and Pasquier (2018) demonstrated a way to evaluate style imitation systems by comparing their generated artifacts with the reference artifacts. They analyse the statistical significance of inter-corpora artifact distance which is approximated with the normalized compression distance (NCD) $\frac{K(xy) - \min\{K(x),K(y)\}}{\max\{K(x),K(y)\}}$. The NCD is a very natural approach to measure how different two artifacts are (the numerator is the amount of information that x and y differ by), but as we have seen in the model section: when multiple artifacts are concerned, a more theoretically correct way is to contemplate a model that describes them well. Then two corpora are maximally different if their individual artifacts exhibit high randomness deficiency w.r.t. each other's model.

Another school of thinkers (Birkhoff 1933; Moles 1966) expresses aesthetic beauty as a ratio of order and complexity $\frac{O}{C}$. Moles (1966) approximates this ratio with relative redundancy $\frac{|x|-K(x)}{|x|}$ equating order with $|x| - K(x)$, the amount by which x can be compressed. While this might work for objects that lie at either extreme of randomness (strings formed by coin-toss or repetitions of bits), it fails precisely for objects that are in-between: e.g. for a highly sophisticated object this definition measures exactly the opposite of order, as such objects exhibit structure through almost all of their complexity. Kosheleva et al. (1998) propose quite a different way of approximating $\frac{O}{C}$ altogether. They equate complexity C with the time $time(p)$ a program takes and order with $2^{-l(p)}$. While the definition of order is similar to the former approach (both argue that the smaller $l(p) = K(x)$, the more order $2^{-l(p)}$ or $x - K(x)$ is there in the object), we now know that $time(p)$ is far from being a measure of complexity of an object; rather it marks the evolution-time of an object from a short program.

Schmidhuber (1997) has also given a framework for "beauty", a concept we are not considering much in our paper. His argument is that for each observer (which in our frame we treat as a model, of the total function variety), the most beautiful objects are the highest-probability objects. This suggests that the most beautiful objects are also the most typical ones, which is worrisome, since most models also have low sophistication (and hence, the most "beautiful" objects will be comparatively trivial). Schmidhuber's approach also does not allow easily for models to be adapted in light of new objects, nor in light of new explanations highlighting the complexity of existing objects.

A Formal Framework We can reformulate the aesthetic beauty ratio $\frac{O}{C}$ of an artifact x by defining order with $K(M)$, where M models the regularities in x leaving out $K(x|M)$ randomness and $K(M) + K(x|M) \leq K(x) + O(1)$. Based on our discussion, M can be built either from a coherent set of objects with which we want to recognize x or it can be its sophistication or useful properties that can be used to generate similar objects. The denominator or complexity is simply the raw Kolmogorov complexity $K(x)$. Then $\frac{K(M)}{K(x)}$ assigns highest aesthetic beauty to masterpieces and lowest to objects that exhibit low structure (random strings or sequences of 1's have low complexity models: fair coin-toss generators or printing $|x|$ 1's). Thus, in general, the algorithmic recipe presented in Table 1 can be followed for aesthetic analysis of computational creativity. We note the relationship between the entities in the first column of the table and three of the four P's of creativity (Jordanous 2016; Rhodes 1961); in a future work, we will integrate the fourth P (press) through analysis of criticism as a creative task in its own right.

Conclusion

We looked at some of the fundamental concepts of creativity through the lens of algorithmic information theory. We saw how typicality or novelty of a never-before-seen artifact can be measured against an inspiring set of already observed objects. Perhaps more importantly, we laid a groundwork for conceptualizing value and what it means to be an authentic creative process. We also highlighted a difference between an artifact's actual and apparent complexities: an observer or critic's time-bounded explanation $K^t(x)$ of an artwork x can be influenced by the real creative process of the artist; while they can also dismiss the fraudulent claims of a charlatan by seeing through the actual value of an artifact. Although this aspect of creativity will be expanded on more in a future paper, such interplay between artists and critics has been often absent from previous computational understandings of creative work. Additionally, the notion of sophistication lets us illustrate a creator's virtuosity present in their creative product. The input d to a generator program p that we called accidental information, can be thought of as the inspiration or an encoding of the surrounding environment that influences a creator program p. A particular delightful outcome of sophistication is its ability to describe masterpieces in the form of highly sophisticated artifacts, whose existence is only possible through a similarly sophisticated program. Although these concepts are not computable, they provide a reliable theoretical foundation upon which other models (e.g. machine learning) can be built and evaluated.

Creative Entity	Attributes	Algorithmic Information Theory Notion
Artifact	Typicality	Randomness Deficiency
	Novelty	Mutual Information between model parameters
	Order and Noise	Model and data-to-model codes
Creative Process	Non-randomness	Effective b-significant program
	Value (also of artifact)	b-significant Logical Depth
Creator	Skills and Style	Sophistication
	Masterpiece	Non-stochasticity

Table 1: An algorithmic recipe for Computational Creativity

Acknowledgement

The work of Dan Brown is supported by the Natural Sciences and Engineering Research Council of Canada. We are very grateful to the anonymous reviewers for a number of helpful suggestions in their thorough reviews.

References

Antunes, L., and Fortnow, L. 2009. Sophistication revisited. *Theor. Comp. Sys.* 45(1):150–161.

Antunes, L.; Fortnow, L.; van Melkebeek, D.; and Vinodchandran, N. V. 2006. Computational depth: Concept and applications. *Theor. Comput. Sci.* 354(3):391–404.

Bennett, C. H.; Gacs, P.; Ming Li; Vitanyi, P. M. B.; and Zurek, W. H. 1998. Information distance. *IEEE Transactions on Information Theory* 44(4):1407–1423.

Bennett, C. H. 1988. Logical depth and physical complexity. In *A Half-Century Survey on The Universal Turing Machine*, 227–257. USA: Oxford University Press, Inc.

Proceedings of the 12th International Conference on Computational Creativity (ICCC '21)
ISBN: 978-989-54160-3-5

Birkhoff, G. D. 1933. *Aesthetic measure,*. Cambridge, Mass.: Harvard University Press.

Boden, M. A. 1991. *The Creative Mind: Myths and Mechanisms*. USA: Basic Books, Inc.

Borges, J. L. 1998. *Collected fictions*. Penguin Books.

Colton, S. 2008. Creativity versus the perception of creativity in computational systems. In *AAAI Spring Symposium: Creative Intelligent Systems*, 14–20. AAAI.

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

Gacs, P.; Tromp, J. T.; and Vitanyi, P. M. B. 2001. Algorithmic statistics. *IEEE Transactions on Information Theory* 47(6):2443–2463.

Jordanous, A. 2016. Four pppperspectives on computational creativity in theory and in practice. *Connection Science* 28(2):194–216.

Koppel, M. 1995. Structure. In *The Universal Turing Machine (2nd Ed.): A Half-Century Survey*, 403–419. Berlin, Heidelberg: Springer-Verlag.

Kosheleva, M.; Kreinovich, V.; and Yam, Y. 1998. Towards the Use of Aesthetics in Decision Making: Kolmogorov Complexity Formalizes Birkhoff's Idea. *Bulletin of The EATCS* 66:166–170.

Lamb, C.; Brown, D. G.; and Clarke, C. L. A. 2018. Evaluating computational creativity: An interdisciplinary tutorial. *ACM Comput. Surv.* 51(2).

Li, M., and Sleep, M. R. 2004. Melody classification using a similarity metric based on Kolmogorov Complexity. *Proceedings of the Sound and Music Computing Conference (SMC'04)* 126–129.

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

McGregor, S. 2007. Algorithmic information theory and novelty generation. In *Proceedings of the Fourth International Joint Workshop on Computational Creativity, Goldsmith College, University of London, London, UK, June 17-19, 2007*, 109–112.

Moles, A. A. 1966. *Information theory and esthetic perception. Translated by Joel E. Cohen.* Urbana, Illinois, 1966: University of Illinois Press.

Mondol, T. 2020. Style recognition in music with context free grammars and kolmogorov complexity. Master's thesis, the University of Waterloo, Waterloo, ON, Canada.

Rhodes, M. 1961. An analysis of creativity. *The Phi Delta Kappan* 42(7):305–310.

Ritchie, G. 2007. Some empirical criteria for attributing creativity to a computer program. *Minds Mach.* 17(1):67–99.

Schmidhuber, J. 1997. Low-complexity art. *Leonardo* 30(2):97–103.

Schütz, K. 2019. *Vermeer: The Complete Works*. Taschen.

Shannon, C. E. 1951. Prediction and entropy of printed english. *Bell System Technical Journal* 30(1):50–64.

Shen, A. 1983. The concept of (α,β)-stochasticity in the Kolmogorov sense, and its properties. *Soviet Math. Dokl.* 28:295–299.

Vereshchagin, N. K., and Vitanyi, P. M. 2004. Kolmogorov's Structure Functions and Model Selection. *IEEE Trans. Inf. Theor.* 50(12):3265—-3290.

Vidal, C., and Delahaye, J.-P. 2019. Universal ethics: Organized complexity as an intrinsic value. In Georgiev, G. Y.; Smart, J. M.; Flores Martinez, C. L.; and Price, M. E., eds., *Evolution, Development and Complexity*, 135–154. Cham: Springer International Publishing.

Vitányi, P. M. 2020. How Incomputable is Kolmogorov Complexity? *Entropy* 22(4).