Reviewing, Creativity, and Algorithmic Information Theory

Dan Brown[1] and Max Peeperkorn[2]
[1] Cheriton School of Computer Science, University of Waterloo, Canada
[2] School of Computing, University of Kent, Canterbury, UK
dan.brown@uwaterloo.ca, m.peeperkorn@kent.ac.uk

Abstract

We connect critical review and analysis of creative objects to a recent domain-independent creativity assessment framework by Mondol and Brown (2021a; 2021b). Reviewing is interesting for at least three reasons. Reviews are time- and space-limited, unlike other tasks. Reviews are a creative task about creative tasks, and that meta-creativity is interesting to consider theoretically. And reviews cause communication and learning; the various actors (the primary creator, the reviewer, and the reader) interact in complex ways. We show how Mondol and Brown's framework connects to the process of review, and show how topics like summarization, contextualization and learning fit within an algorithmic information theory frame. We also give some interesting examples, such as analysis of conceptual art and concert reviews, as computation tasks. We finish by showing that (as is often true of algorithmic information theory ideas) it is hard to fulfill our objectives with practical systems, due to uncomputability or intractibility issues.

Introduction

Reviewing, either of a creative product like a concert or novel, or of an academic product like this paper, includes several goals that the reviewer must satisfy. First, the goal is to provide an overall assessment of the quality of the product in the first place: is the paper good, should other people come to see the further offerings of the performance, and so on. Second, the reviewer summarizes the product: what is the relationship between the main characters of the novel, say. Third, a good reviewer offers contextual information, for example: how does the object fit within or expand its genre, or was the performance better than expected? Academic reviews also come with creative improvements and recommendations from the reviewer.

Reviewing is a computational task. With this lens, we see inputs, background knowledge, execution time constraints, memory use constraints, and output size constraints, and can identify what a critic does, what makes a good critic, why different critics will identify different aspects of a performance, how readers can interpret a review, and more.

Review is also creative. Readers may be delighted by a review's cleverness, or by its surprising insights. A reviewer may unexpectedly connect properties of the manuscript with the author's biography, or may examine an oeuvre in a novel way, or identify how the work expands knowledge.

We approach review via a computational creativity (CC) framework and focus on review quality and creativity. Additionally, we consider certain aspects of how reviews are created (in particular, their limited creation times), how reviews affect readers, and how reviews can form a communication channel between a creator and a potential audience. In emphasizing these perspectives on reviews, we reconstruct the 4P's framework (Rhodes 1961), adapted to CC by Jordanous (2016). That is, reviews are themselves a Product, created by a Producer, via a Process, for an audience (Press).

We focus our theory on the algorithmic information theory approach to computational creativity evaluation of Mondol and Brown (2021a; 2021b). That framework offers a genre-independent definition for key concepts like novelty, quality, typicality and more. However, our focus on review also extends Mondol and Brown's work, in that we focus on properties of the reviewer: what program is the reviewer executing? What background information does the reviewer have? How much time does the reviewer have in which to make a judgment? Does the reviewer have prior knowledge or expectation about who will be reading the review?

We also focus on the reader of the review: their background and ability to interpret the review. As such, our approach focuses on the review as a complicated communication intermediary between a creator and a consumer. For agents to talk about creative objects, we need to formalize how they convey their opinions. Thus, key to our understanding of review is the relationship between a reviewer and the reader. An expert reviewer may identify aspects of a creative object that a naive reader cannot understand; meanwhile, a naive reviewer may not be of use to an expert reader, who already understands the work and wants to assess if, for example, a concert of a beloved piece is worth attending. One example of a reviewer might in fact be the creator of an object, summarizing it as a teaser for potential readers; this still fits within our frame, especially if the creator is knowledgeable about potential readers.

Our goal here is to document how the Mondol and Brown framework can be applied to the overall task of critical review. We discuss what it means to be a "good" review or reviewer. And, we identify some serious challenges relating to the Mondol and Brown framework in the first place, given that their frame expects that the quality of an object is connected to its mathematical *logical depth* and *sophisti-*

cation, two hard-to-estimate quantities, and that the formulation by Mondol and Brown does not restrict the work of a reviewer in approximating the information found in an object. We connect our work to computational creativity by discussing previous efforts to shape critical review as a CC task, and to describe how individual reviewers might bring their own perspectives to the task of review. Our paper contributes to overall understanding of how artistic objects form a means of communication between producers and their audience, and the role intermediaries play in that communication act.

Background

Here, we give two different pieces of background to our project. First, we describe how review has previously been portrayed in the computational creativity literature. Then we give the necessary mathematical background for this overall work, focusing on the Mondol and Brown framework. We finish this section by giving the notational framework in which we embed the review task in terms of its various actors.

Review and CC

Few papers discuss *review* in computational creativity, but two ICCC papers (Fisher and Shin 2019; Roberts and Fisher 2020b) do. Both discuss an early effort by Stiny and Gips (1978) connected to algorithmic aesthetics, where they connect design directly with criticism, and so the full Stiny and Gips model consists of two parts, a structure for design algorithms, and a structure for criticism algorithms. In this paper, with the Stiny and Gips model, we specifically refer to its criticism component. Stiny and Gips base their model on Craik's general model of thought (Craik 1943), which consists of a *receptor* that produces a description of what it senses in the world, a *processor* that transforms the description for the *effector* to produce an observable response.

In the Stiny and Gips model (based on Roberts and Fisher (2020b)), receptor function R takes object α and contextual information I_α to produce description δ. Processor P, in the Stiny and Gips model, is an algorithm informed by an aesthetic system, which is a set of algorithms that form the aesthetic criteria, and a memory of contextual information I_m. P takes description δ and contextual information I_m, and outputs, in addition to the original description δ, the best interpretation ι and numerical aesthetic evaluation ϵ. The effector E is a function that takes δ, ι, and ϵ to generate review χ.

Fisher and Shin (2019) identify review as a separate creative task and highlight that critics are part of larger creative ecosystems. They argue for the importance of computational critics and identify five desiderata for a critic. The computational critic should 1) understand the medium, 2) emphasize the authorial intent of an artifact, 3) reason about the creator's output and its relationship to the subject, 4) situate the artifact in social and historical contexts, and finally, 5) gauge the response of the readers and viewers to the critique. Besides the standard CC criteria for creativity (Boden 1992; Ritchie 2007), they analyze essential dimensions involved with critique: authority, authenticity, explainability, and interpretability. Additionally, because of the role critics play in society, they address the ethical concerns for when computational critics are implemented. The follow-up paper (Roberts and Fisher 2020b) further explores the Stiny and Gips model and adjusts their approach by formalizing their desiderata. In particular, they introduce a justification γ, as another output of the analysis algorithm P and an additional input to E. We extend their approach by focusing specifically on the computation that happens in both the reviewer and in the reader, and looking at properties of all of these agents.

Scientific review has been identified as a domain for implementing computational critics because it provides excellent grounding in a specific context (Fisher and Shin 2019). A first attempt at a computational scientific critic is *pReview* (Roberts and Fisher 2020a). A recent paper (Yuan, Liu, and Neubig 2022) discusses automating scientific review as a Large Language Model (LLM) task; that paper is more oriented around the practical LLM techniques to do this automation, and identifies some key desiderata.

Algorithmic Information Theory and Creativity

Our frame for analyzing creativity is the algorithmic information theory (AIT) framework of Mondol and Brown (2021a; 2021b). They give a Product-focused definition of basic concepts in creativity, including value, typicality, and novelty. In their framework, all of these concepts are based on properties of a Turing machine program whose output is the digital objects under study: an object s is *valuable*, for example, if there are short programs whose output is s, but where all of them require long runtimes to execute. Here, we give a brief introduction to the Mondol and Brown framework; interested readers are referred to the full paper (Mondol and Brown 2021b) for more detail. The standard textbook on Kolmogorov complexity (Li and Vitányi 2019) gives more complete definitions than those that follow.

Kolmogorov complexity We study digital objects, represented unambiguously. Given a universal Turing machine U that can generate any computable object s, the *Kolmogorov complexity* $K_U(s)$ is the length of the shortest input P^* for which $U(P^*) = s$. We ignore details of U, and often describe $K(s)$ without reference to U, speaking of the execution of P, not U. The runtime of program P is the number of Turing machine steps before P halts (and is infinite if it does not halt). The conditional Kolmogorov complexity $K(s|y)$, is the length of the shortest Turing machine which, on input y, outputs s: this quantity measures how similar s and y are, or how much knowing y allows us to compress the string s.

$K(s)$ alone is insufficient to identify if s is creative. Random sequences have Kolmogorov complexity very close to their length with high probability. The n-bit string 0^n has very low Kolmogorov complexity, at most $\log_2 n$. Both are not of creative value. Instead, Mondol and Brown use two other concepts in AIT as evidence of an object's creative value: *logical depth* and *sophistication*.

Logical depth The *logical depth* of s is the minimum runtime of programs with output s and with their length close to $K(s)$. Specifically, $ld_c(s) = \min_{P:U(P)=s,|P|\leq K(s)+c} \mathrm{runtime}(P)$, for some small parameter c. Objects with short, fast-running programs are not deep (they are highly and trivially compressible and decompressible). Objects with only long programs are not deep (they are random). Mondol and Brown show that high-quality objects can be compressed, but their decompression is slow: non-random parts of the object must be painstakingly reconstructed. Consider, for example, a painting where the positions of key objects are described by a short-to-describe algorithm that requires a long time to execute: there is structure in the painting, but it is hard to tease out. The logical depth model of value says that for an object to be of high quality, there must be substantial and complex work embedded in the object. By contrast, Schmidhuber (2010) has argued that beauty (which he treats as similar to quality) is connected primarily to being of short description, regardless of the required runtime of a generation algorithm.

Sophistication The *sophistication* of s comes from a two-part representation of digital objects. An object is defined by giving the class of objects for which it is typical (to its *model*, M), and the information required to describe the specific element in that class. (There is often deliberate ambiguity between M, the program for the model and $L(M)$, the class generated by M.) Valid models are typically restricted; one straightforward requirement is that models are Turing machines that halt on all inputs, or Turing machines that can generate any output. With such a restriction on valid models, then, the sophistication of a string is $soph_c(s) = \min_{(M,d):|M|+|d|\leq K(s)+c, U(M,d)=s} |M|$. It is the shortest model for which the two-part representation comes close to optimally encoding the important details of s. The model encodes the category of objects for which s is a *typical* example; typicality is of course a standard desideratum for computational creativity (Ritchie 2007).

Sophistication is not an easy concept. The restriction to models that are total functions removes the universal Turing machine U as a valid model, since (without it) $soph_{|U|}(s) \leq |U|$ regardless of what s is, since the universal Turing machine, run on the shortest program for S, will yield s. Instead, the model framework requires the identification of a computable class of objects with a relatively short description that includes s as a *typical* member, and then the details that identify s from all of the other class members.

Both of our previous examples of non-valuable objects have short models. A random string s with high $K(s)$ has as its model a constant-length "print" program p that just outputs its input, so s then has a two-part code of length $|s|+|p|$. By contrast, we need to be a bit more careful in describing a repeated pattern. Consider the string $s = 0^k$, where the number k has $K(k) \approx \log_2 k$ (that is, the binary string k is uncompressable). It is easily modelled by a constant-length program p that on input of a binary number x, outputs 0^x; then the two-part code (k, p) has total length very close to $K(s)$, so p is a good model for s. Neither of these strings, hence, is sophisticated: they both have short models.

By contrast, let s have fairly high Kolmogorov complexity, yet compressible by complex programs. It cannot be simply represented by its Turing machine representation as input to U, but we can build a model of similar size to $|p| = K(s)$ whose language only contains s: this machine ignores its input, and implements a universal Turing machine running p, which always halts and outputs s. The existence of this machine shows that the sophistication (assuming this machine is a valid model) cannot be substantially larger than $K(s)$, but it could be smaller if the two-part representation allows the model M to encode all of the compressible information in s, while allowing the data d to be uncompressible.

Key to the relationship between sophistication and value is the important part of the two-part representation, M, which represents the regularizable information found in all outputs of the model. By contrast, d, the data, is random information ultimately not relevant to the meaning of the model. A naive consumer of s may not be aware of the inherent information of s, and assumes s is less sophisticated than it is (mistaking s for more random data), or they may not understand s sufficiently to compress it fully. This challenge is the connection between review and the Mondol/Brown framework.

Compression, lossy compression, and generation The Kolmogorov complexity of an object s defines how much information is in s, by giving an optimal compression for s. If s is logically deep, no short program speedily generates s: the only speedy generators of s require longer descriptions. If we restrict to fast generators, we can only use models that do not fully understand the information found in s.

If we restrict to programs that are both fast and short, we cannot generate all of a logically deep string s. Instead, we can only lossily represent s. Let $s'_{t,n}$ be the closest approximation to s that we can obtain by fast, short programs: that is, $s' = \mathrm{argmin}_{s'} K(s|s')$, where s' searches over all machines of size at most n bits and with runtime at most t steps. If n is close to $K(s)$, and the runtime is kept smaller than the logical depth, then s' may be able to represent some surface features of s, but cannot identify the valuable pieces of s. However, if the runtime is kept moderate, and the programs must be short, we can still potentially explore some small piece of the logically deep core of s, which may offer some hint that the whole object is valuable.

Generation is also key to review. If we assert s is a "typical" example of a genre, that implies that a generator G for that genre, run on random inputs, would yield an object of similar quality and appearance. (We note that we will often use the more informal term "genre" to refer to the class of objects created by a generator, instead of the more common algorithmic information theory term "class"; in part, this is because we want to focus on creative objects.) For example, if G generates typical romance stories, then the parametrization might indicate the names of the characters or their occupations, as well as some arbitrary details about the story, then G could generate that new story. We only claim to *understand* s when we can describe such a generator; further, if on other random inputs, G's output does not fit the genre, then G is a bad representation of the class. Consider a general-purpose compression system, like the Lempel-Ziv

(Ziv and Lempel 1978) algorithm. It may compress s, but if run on a different input, it is likely to generate a completely different type of output than s. As such, it is not a good model for s.

Compression and generation are very difficult tasks to understand for simple objects because good generators of small objects must be much larger than the objects themselves; this yields a situation in which to analyze the quality of an object, we must instead consider a collection of objects of a type or make dramatic restrictions on what is a valid model; see Brown and Mondol (2021) for more details. Table 1 gives a summary of AIT concepts used in this paper.

Review and algorithms

In our formulation, A creates the object s, B uses s as one of its inputs and creates a review r of s, and C uses r to contribute to its understanding of s, and whether or not to further investigate s. For example, A might be a movie studio, creating a new movie; B writes a newspaper review of the review, and C decides whether the movie is worth going to on the basis of the review, in addition to learning from the review.

Different reviewers may discover novel aspects of s; for example, B_1 may focus on the brushstrokes of a painting, while B_2 focuses on the biographical details of the creator and B_3 focuses on just giving service journalism about the exhibition housing it. All of these might be more or less useful to readers; we will discuss this issue later in the paper.

Review as a multi-part CC task

A review of a creative object includes several parts (Fisher and Shin 2019). There is a summary of the object, situating it in the domain from which it comes: for a novel, perhaps talking about the characters and their relationships. The review will include the reviewer's assessment of *quality*, either in textual form, or as a numerical rating. It can include framing information: how the genre has changed, or what the reviewer adores or despises. The review may describe how the new object alters one's understanding of the field, or include biographical information about either the creator or reviewer. They can also suggest improvements; this is appropriate for academic papers, but could apply to any object: a recipe tester might identify missing flavours, or a musical reviewer might identify awkward lyrics or harmonies.

Review as a computational and creative task

Each reviewing subtask is computational: the input to an algorithm is the object s, and the reviewer's process in moving from their own knowledge and mental state to the textual review is the execution of an algorithm. Further, review can also be seen as *lossy compression*: in summarizing a piece of music, one gives enough description that a reader has some better idea of the piece of music than they had before reading the review, and (if it is well-prepared) a better estimate of the quality of the piece of music than they had before reading the review. Formally, consider an object s under review, created by a creator A. Assume that A is a Turing machine computing a total function (A halts on all inputs), and $s = A(d)$, for some input d that comes from the outside environment of A. Program A is not known to any other observer. Reviewer B is also a Turing machine that is given s as input, and computes a review $r = B(s)$. B may have some framing information as part of s (the name of the composer of a song, the date of a performance, the title of a poem, or an artist's statement); in fact, it can be valuable to consider $s = (x, y)$, where x is the main object and y is its framing information (Charnley, Pease, and Colton 2012). Review r should include (possibly not well separated) each of the parts we have described: the summary, estimate of quality, and contextual framing for the object s. Then, the review r is sent to the reader C, potentially along with the same framing information available to B, and it is used by C to help update its understanding of the object s, without C being able to see the object s or its creation by A directly. See Figure 1 for a conceptual model of this process.

Conceptual model of review in AIT

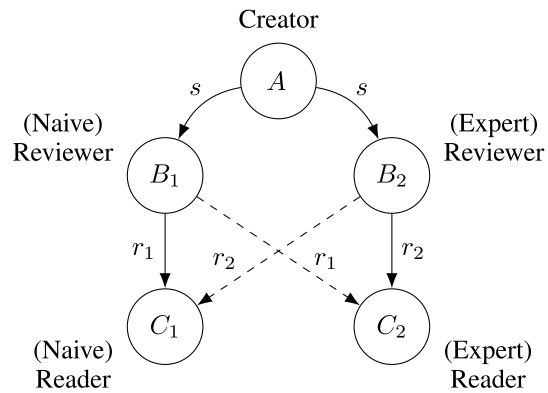

Figure 1: Here, s is the object under study, r_1 is a general review, and r_2 is an expert review. The usefulness of a review r to a reader C is dependent on its prior model of s.

Review is creative: in addition to identifying the properties of s, an expert reviewer can tantalize and educate their readers. Famous movie reviewers like Pauline Kael and Roger Ebert were considered masters of the craft, due to their encyclopedic knowledge of the movie industry and its customs, and other critics like Susan Sontag can join a diverse range of fields in giving context to s and why it matters. Hence, reviews can be assessed for typicality (is this recognizably a review?), novelty (does it identify unexpected things about s?) and value (does it show important aspects of s, or focus only on surface features of s?), and similarly, B can also be assessed (though the distinction between B and r can be hard to identify).

How review differs from other computational tasks

One other key condition governs review, and makes it simpler than other forms of critique (such as, for example, large literary biographies): reviews are created in a short time frame and must fit in a short space. One might be tasked with an 800-word review of a 2-hour concert, due three hours after the concert ends, or one might be handed an 8-page paper

AIT idea	notation	quick summary	meaning for reviews
K-complexity	$K(s)$	shortest program whose output is s	not useful
conditional K-complexity	$K(x\|y)$	how much information is shared between x and y	measuring novelty and typicality
logical depth	$ld_c(s)$	runtime of short programs for s	one measure of value
sophistication	$soph_c(s)$	length of shortest model for s	another measure of value
model	M	program M that outputs s on a specific random input d	way of describing s
good model	M	program M whose length is close to the sophistication of s	appropriate way of describing s

Table 1: AIT concepts used in this paper

and a 2-month deadline in which to review it. B is thus limited to Turing machines that always halt: they must halt in a restricted time t and with their output r restricted to a limit λ. This limitation is interesting: if $t < |s|$, the reviewer cannot examine all of s, and if $\lambda < K(s)$, then it is impossible to fully describe the object (in terms of describing a program that generates s in its entirety). Further, if $t < ld(s)$, then it is not possible to verify the correctness of a short program for s (let alone find that program in the first place!). If $\lambda < soph(s)$, then we cannot give a good model for s.

Another difference, which we explore later in the paper, is that review is connected to the reader of the review's preexisting understanding of the object under study. Review is a step in *learning*: if a new reader can be brought up to speed about the genre by reading good straightforward reviews of masterpieces of the domain that include descriptions of important milestones in the genre, this is a valuable service to that reader, as it allows them to build better models of both s and its genre. If the reader is already knowledgeable, then the needed review to help them better understand a complicated new piece of work is more complex.

Review and AIT

Review fits messily with AIT. A review with a short execution time cannot identify that an object is sophisticated, for two reasons. First, even if B knew the program and input that A used to generate s, it cannot run that program in a small amount of time. If s is sophisticated, then the program/data pair (p, d) that documents its sophistication is both long and slow to execute. Even if (p, d) generates s, a simpler program might also have generated it: the creator of s may have toiled to create s, but not made a valuable object.

Instead, all review tasks can only be approximated. The reviewer B can consider the object s within their expertise, and can find evidence for novelty and value, but cannot be guaranteed of success. As B specializes, they may be more prepared to find such evidence, but at the loss of broad applicability, particularly given that B has a time limit.

Aesthetic evaluation and AIT

To identify value, we estimate logical depth or sophistication. Novelty has a full description in the Mondol/Brown framework as well (Mondol and Brown 2021b), based on how much a new object differs from a corpus of members of the same genre. A novel object is both familiar (since it can be placed in an existing model), but also unfamiliar (it is distinguishable from other objects generated by the model in not just random, meaningless ways). This framework lets us not view a sonnet as "novel" when placed in a set of objects that are all paintings, for example. Quality (as novelty, typicality, and value) is estimated by describing what an object is in the context of a good model for that object and its class, and why it is exemplary of a sophisticated model.

Summarization and AIT

To summarize effectively, one should identify the model from which an object is a typical example, and indicate the ways in which the model does or does not fully satisfy the non-random information found in the object. This relates to quality estimation: the critic identifies the model from which an object comes, and also the random parameters for that model. For example, "It's a Jackson Pollock painting, with the paint spills in these positions, with these colors." Again, novelty estimation is separate from the summarization process, and thus the task of review is not just summarization.

Contextualization and AIT

Contextualization is included in an object's model: if we know about a good model M for s, it cannot create surprise, since s is then a typical example of M's outputs. This requires that the model does not accidentally move outside the genre on random inputs. We discuss this topic in the examples later, but a straightforward example is Duchamp's "Fountain" (which was a urinal): obviously, a very short model can ignore its input and generate a sufficient description of the piece, but it will not generate any other Duchamp-readymades, nor distinguish them from fakes. By contrast, a much larger model that presents Duchamp's Dadaist background and the scenarios in which he worked might generate other examples of Dadaist conceptual art. Obviously, a short review cannot give a full description of such a model, but can present information necessary to describe how it differs from models, like "print" programs.

The role of the reader of a review

Our discussion so far has treated review as a very quickly created description of an object, focused on quality estimation and model identification. But reviews communicate: the readers of the review have their own tastes, their own goals, their own knowledge, and needs to extract from the review useful information about the object s.

This complex relationship requires unpacking. These three actors are not playing a complex game of Telephone: instead, a smart reviewer learns about both creator and reader to ensure that useful information gets passed between these two actors. B does not not merely build a short description of the model A used to create s, but also needs to pick the best way to describe this model, given what C already knows about objects like s. If C is an expert on the topic of objects like s, then the updates B needs to give to the internal model C has may be of a very different class than the model description given by B to a naive reader. See Figure 1 for a sketch of this process.

This observation highlights a key problem with the Mondol/Brown framework: it equates quality with logical depth or sophistication, removing the chance for a reader to have preferences and tastes. Their "complexity above all" frame does not allow for this person to like romance novels and that person to like conceptual art. Instead, for any object, the goal is to group it with objects of similar kind, and establish how well the new object extends that group: how much s expands the set of valid members and how much computation is built into its non-random parts. We must expand our understanding of the role of a reviewer and consider both learning and the Press component of the 4Ps of understanding creativity (Jordanous 2016).

Learning

A clear goal of a review is to allow C to better understand s and estimate its value. Our definition of value is about required computational effort and expanding our understanding the class of s, so, the B wants C to change its model of the class of objects s comes from. That is to say, C reads the review r, and as a result, its model for s, which C has not seen, changes, as does its assessment of the value of both s and its class. This is what is meant by learning: we modify our knowledge of how objects and concepts relate. If r does not allow C to either assess s or to alter its understanding of the class, then C did not learn anything useful, and the review was useless *to that reader*, though it might be useful to others. If B is writing without an audience in mind, then their optimal choice is to express observations that will be clear to any reader, implicitly assuming that C has no prior background in the subject under consideration, and that without encountering s or r, C will know nothing about s. If C is already an expert on objects like s, then B's job in writing a review is to focus C's interest on aspects of s that highlight quality or novelty, and on subtle reasons why s differs from previous members of its type.

Estimating the quality of a review Consider review r of object s. A natural way of assessing the quality of the review r is to ask how much r simplifies s: that is, what is $K(s) - K(s|r)$? But if the information of r consists overwhelmingly of random details about s, then having the review will not inform C about anything useful to assess s. Instead, the review must help build a more accurate model of the generator of s for r to have actually been valuable to C, or it must represent the logically deep components of s.

Review r can be even worse than just offering random information about the object under study: it can make false claims! Consider a review of a concert that misidentifies the set list, or a review of an art exhibition that gives false contextual information about the painter. The review provides no information to further the reader's understanding, and the model the reader C brings to the show may be less accurate (finding s more atypical) than before. We will consider this a bit further below when we look at the reader's experience.

To estimate the quality of review: If M is a good model of s, then by conditioning on r, we can estimate how many additional regularities are captured after observing r, which is $K(M) - K(M|r)$. Conversely, the irregularities of s are modelled with $K(s|M)$. We could fit the model M with r to estimate if the model improves as it captures more irregularities, that is $K(s|M) - K(s|M, r)$. We can separate information about s in r into two parts: those that connect to M and those that connect to its random input.

Alternatively, we can use the logical-depth frame to also explore the quality of a review: let $ld_c(s|r)$ be the minimum runtime of a program for s whose length is at most $K(s|r)+c$. This computes how much runtime is needed to give a short description of s given r. If $ld_c(s|r) \ll ld_c(s)$, then the review r captures critical information found in s, and is therefore a good review. If instead the required runtime has not changed much, then while r may capture information about s, it does not conveyed much of value.

Estimating the quality of a review to a reader Again, we must also consider the quality of a review to a specific reader. If C does not read Czech, then a brilliant Czech review will do nothing to help the reader. Moreover, there is insufficient information in r to teach the reader to read Czech. We must consider how reading r affects C, and see how information is transferred.

Let $K(M|C)$ be how much information C needs in order to create a good model M, without having seen the review r. If $K(M|C)$ is high, then s is complex and the reader is unprepared; if $K(M|C)$ is small, the reader is prepared. In both cases, r may help C change its model. Suppose that M_1 is a model that does a good job of explaining previous examples that C has seen, and that M is a model that explains both those previous objects and also s. $K(M|M_1)$ is the added novelty brought to bear by the creation of s. The key quantity under consideration, then, is $K(M|M_1) - K(M|M_1, r)$. That is, how much information is created in C by reading r that is relevant to M? A detailed review that establishes a small component of M, and how it changes as a result of s, may be of help to an advanced reader, but may contain much less overall information than a general review. However, that general review may offer little actual new content to a reader C.

We may impose a time limit on C's execution as well. It is probably inappropriate to allow C enough time in understanding r to learn a new language. If we put a time limit t' on C's computation using r as an input, then let $M' = C(r)$ be the new model that C has after running for t' steps on input r; if $K(M|M', C) < K(M|C)$, then C has learned something useful about s from the review r in its short time.

C can also have knowledge about specific reviewers, and can learn to trust a particular reviewer B to give good advice. Our model does not really handle this circumstance, which has been brought to our attention by a peer reviewer (whom we hope enjoyed our work); however, C can identify which reviewers' overall quality estimates most track with its own choices. It is worth recalling that in the Mondol/Brown frame, since quality is an absolute quantity, there is less accounting for individual taste; we discuss this below when we consider limitations of our framework.

Communication through a naive reviewer A review can *still* be useful to an expert reader even if the author of the review is not an expert. B may explicitly notice features of s that are novel even to an experienced reader. If an artist has changed their colour palette, B may not know that the shift has happened, while highlighting A's colour choices. C can then update their model of A's work, even though B does not explicitly represent the change. A related example of this phenomenon might also be if someone noted the D-S-C-H motif in "Rejoice in the Lamb" (Britten 1943), which is an homage to Dmitri Shostakovich, without explaining the four-note sequence. Communication happens between A and C even though B is not aware of the content.

Creative reviews, creative reviewers

Reviews themselves are creative objects. A review r of a creative object s has *value* by describing s (giving information about the model that generates s, or about the logical depth and novelty of s). But it can have novelty and value in that process. Consider reviewers B_1 and B_2; if both identify elements of s showing it is of high quality, but those identified by B_1 are more often known by people in the audience than those identified by B_2, then to a typical reader, B_2's review will be novel, and as such have much more value than B_1's. A clever reviewer discovers new things to enjoy about a piece of creative work, and then shares that joy. Reviews can also provide delight to their readers in and of themselves; not only can they highlight the creativity and model-breaking natures of a new object, but they can just be objects of creative value in their own right. It is a challenge to separate these aspects of a review's quality from the overall analysis of creativity.

Creative reviews can also suggest improvements. AIT does not offer an easy way to give small corrections; if these corrections take up the bulk of a review, they will not change the underlying model very much (since one can use an existing model augmented with "at line X, change word Y to word Z" commands). Reviews that describe the underlying model for s that the reviewer B believes A has used could be made richer, and improve the overall value or novelty s.

We can also explore the creativity of the reviewer, not just the review; while the Producer perspective is not an obvious use case for AIT (which might be expected to focus on products, since it connects to properties of objects), one can either analyze the program that B executes in its review process, or one can focus on a collection of reviews by B, to see whether a single creative review is an accidental flash of genius or represents consistent excellence in a reviewer's work.

Examples and limitations of the approach

Here we explore some real-world examples of review and how they fit within our conceptual framework.

Conceptual music and art

Consider the iconic piece 4'33" by John Cage, in which the performers sit for four minutes and thirty-three seconds making no deliberate sounds. A performance of this work is hard to describe in a single object s, but the piece can be "summarized" easily. But if we look at it with an eye towards AIT, and in particular, towards sophistication, it is insufficient to model it as a "print" program whose input is "make no deliberate sounds for 273 seconds." This model ignores the awkwardness of sitting in a room with other humans where normally one expects to see music performed in the normal way. Instead, to properly summarize, one needs a model that, on random inputs, yields typical performance experiences for even this ostensibly simple piece, differing only in random details. Such a model is likely impossible, even for 4'33", to present in a short review. B must include descriptions, likely to improve C's understanding, enough to push its model closer to the truth of what the piece is. To describe the novelty and value of the experience, C's experience with conceptual music would need to have been pre-estimated by B (for a naive reader, this piece utterly alters their experience of what a concert is; a seasoned reader would understand what differentiates performances).

Similarly, looking at Duchamp's "Fountain" mentioned earlier. The short "it's an early twentieth-century urinal in an art gallery" review does not offer enough information on why this piece provokes such ire among gallery-goers, and certainly does not allow the reader to assess whether it improves or worsens the show. By contrast, a review that describes the state of early 1900s sculpture, and describes how "Fountain" expands the art gallery experience, is potentially a much better review, giving a reader a better sense of what other provocative conceptual art would be like. The "it's a urinal in a gallery" reader will be more prepared for other examples of other nouns to replace "urinal", but will not have a reason to understand scatological conceptual art in general; while a reader given a provocative review that describes "Fountain" might find, for example, Andres Serrano's 1987 work "Piss Christ" (a photo of a crucifix in a vial of the artist's urine) less surprising.

A conceptual artist's process can also be the focus of the works; consider Roman Opalka's paintings of the "numbers from 1 to infinity", where the artist's project was to paint consecutive numbers to represent the passage of time. Here, one might review either the individual paintings, or the process itself, in either case, one would again contextualize the creator's practice within the genre of conceptual art.

Finally, Sol Lewitt's work, which consists of short algorithmic descriptions of exactly how someone is to create the object. It is possible that a model that creates algorithmic art might be created by a reader, after reading a review. The review might describe several specific choices made by the person implementing the algorithm, to give a sense of what a different typical implementation looks like, and the complexity needed to properly understand Lewitt's work.

Searching for a relevant explanation for conceptual art, then, is a computational task of finding evidence for quality, valuable summarization, and contextual information. Even for "simple" conceptual pieces, there may be much to discover. Again, review is not merely lossy encoding of s: it is lossy encoding of s and how to understand it.

Concert reviews, short and long

A concert review offers an opportunity for both conveying important details about the performance (date, time, venue, set list) and also individual aspects about the performer and their genre. In a short review, the reviewer may still convey core performance details to naive readers, but these details are obvious to a well-prepared fan. A review that goes into detail about what was amazing might help out a novice reader to understand what demonstrates successful performances. While also focusing on specific details that made the concert different, allowing the expert to see why there is novelty and not just value in the performance.

A much simpler review (or preview) can also still yield important information about genre and quality. One of us once saw the Hilliard Ensemble, a Renaissance vocal quartet, describe their next piece as: "Late Tallis, early Byrd". These four words prepared the listener for the upcoming piece, while also making it clear that the singers are English (the piece they were about to present was French). Like so many other cases of short sentences, these four words provide much context and much model shaping to a prepared listener, and no context at all to a novice (they might even confuse one into believing the piece was English!).

Limitations of our approach

Fundamentally, our approach *describes* what reviewers need to do in the process of assessing quality, typicality, and novelty of a creative object, but the actual project of creating, within a time limit, a high-quality short review is complex.

Reviewing good objects is hard Reviewers assessing a sophisticated object face a challenging task: they must identify, in a short time, why the object comes from a large, slow model M. If B is itself highly sophisticated, it may be able to zero in on certain subfeatures of the object under study, and identify why these are consistent with the object overall being of good quality: for example, they could describe a single object in an exhibit, giving the reader enough information to better understand what to look for on a time-unlimited tour through an installation. In this way, the reviewer may spend serious effort to identify bits of information from the creator program A, but nonetheless, the review is productive, since $K(A|r) < K(A)$.

It is also possible that a reviewer unprepared to analyze the high-quality object cannot, in the short time allowed, explain the features of the object. In this case, while B fills r with true information about s, it will not be very useful to C. In the AIT sense, this means that the reviewer might describe some random bits d that are the input to the creator program A, not pieces of the structure of A itself. While these contribute to $K(s)$, they are not central to understanding s, and the reader of the object is not better prepared to encounter other objects of the type.

Since reviews are time limited, no critic is a general-purpose critic. Instead, a sophisticated, specialized critic, when handed an object it is well posed to review, can identify high-quality parts of the object efficiently, and is preset to describe the model in language that a reader will understand. A claimed general-purpose reviewer must spend some of its time chasing down blind alleys. We could theoretically handle this situation by creating teams of reviewers (the equivalent in computational terms of parallel programming), but the most sensible thing is to assign reviewers by expertise and by awareness of the identities of the readers.

Reviewing bad objects is hard Reviewing unsophisticated objects is also hard. If an object comes from a simple, fast model, it can still be extremely hard to verify in a short amount of time. A high-quality reviewer might have learned features of high-quality examples of a genre without sufficient awareness that these features are not the important items to find; if every good painting by a particular artist uses a lot of blue paint, it is easy to highlight this surface feature and incorrectly assigning high-quality to trivial new objects. Mondol and Brown (2021b) also discuss the *charlatan* phenomenon, when clever agents knowingly assign high-quality estimates to poor objects by describing complex programs with long runtimes to compute s; this can be seen as (fake) evidence of logical depth. Naive readers can easily be confused by such reviews into overestimating the object, while experienced readers still must identify errors in a review that claims a junky object is a work of genius.

An incorrect review r of a bad piece of art has almost no effect on a reviewer's actual understanding: the information gain of $K(M|C) - K(M|C, r)$ will be modest, since r gives minimal information towards M, a good model for s. Still, since a bad piece of creative work has low $K(M)$ to begin with, an incorrect review may accidentally give information toward an initial simple model; this is part of why we focus on the absolute number of bits in $K(M|C) - K(M|C, r)$.

AIT shows the challenge in review We end this discussion of AIT and reviews by noting that our message is not hopeful. Reviewers who can detect complexity quickly are rare, and we cannot verify high logical depth and sophistication in short runtimes since a short, fast program may also exist where a short, slow program has already been found. It *is* sometimes possible to properly explain why an object is trivial, but a critic can be caught up in enthusiasm for the trivial work of a beloved creator or a charlatan, and give an explanation that (falsely) highlights perceived complexity.

This difficulty comes down to the twin dilemmas of the Mondol/Brown aesthetic theory: objects are valuable if they embody much work, and such objects appear more trivial (random) to naive consumers until they are explained with appropriate models. (By contrast, the Schmidhuber (2010) approach focuses solely on K-complexity; it, too, is hard to tease out since simple strings may still appear complex unless one knows the short algorithm that explains them.) In a time-limited review, it is challenging or impossible to explore this work, and to properly explain these models. Instead, special-purpose reviewers and experts can only approximate this aesthetic lens.

Future work

We have described how reviewing fits with algorithmic information theory. The challenge is to make our insights practical: almost all areas AIT touches find either huge runtimes or uncomputable results. For example, identifying good models requires that not much smaller models work well, and that the model input is truly uncompressible; computing logical depth requires knowing $K(s)$ and the runtime of machines computing s. This is beyond a general-purpose algorithm with reasonable runtime for any interesting input, meaning that AIT largely provides an abstract restatement of a number of normal computational or human tasks. There are also a few specific concerns that connect to this specific application of AIT: notably, real-world algorithms for building seemingly creative objects seem huge, and that any form of embodiment can cause us to question what "the object under study" is, and whether it has a unique identity. Further work will address the queries below about practicality, parameterization, and embodiment.

Connections to machine learning

Large language models (LLMs) and other large machine learning models show some of the complexity of review, as they require a truly enormous number of parameters before giving reasonable language creation. If such a a system is the "model" in the sense of our paper, then $K(M)$ will always be huge: it is the size of the code for the LLM; the data part of the two-part code is the input (prompt) to the model. This, though, requires extremely long strings s for $K(M) < K(s)$; such strings cannot be well explored by a time-limited reviewer. We still require much assessment to figure out for what kinds of input the AIT frame can work.

Embodiment

Another challenge with linking AIT and review is the phenomenon of sensory embodiment (Guckelsberger et al. 2021). Do B and C interact with the same object s? We have assumed that s is properly represented by a single digital string, but B and C may perceive it differently. B and C may either or both experience disability and inexactness in their ability to perceive s. If B watches a concert from the front row, the experience described in r may be inaccessible to C from a back row. If B is colour-blind, r cannot help C learn about the subtle choices in shading that A made.

Another way in which embodiment (and other concerns about physical and memory limitations) affects AIT and review is the finiteness and sequencing of memory. In AIT, $K(s|y,x) \approx K(s|x,y)$; it does not matter much in which order the two objects x and y appear. But encountering two different large and complex objects will have different effects if they cannot both be stored; the more recently-experienced object may have more details in memory, while the more distantly-experienced object may have had a more fundamental effect on the internal model the reader has of the category from which s, y and x all come. The key effect here, then, is *forgetting*, which we propose to discuss in future work.

Conclusion

We present review of creative objects as a computational creativity task, using Mondol and Brown's framework as a starting point. In our discussion, review is recast as quickly identifying features of a good model explaining an object. For a simple object, the features identified in a review can either be among the few interesting (non-random) aspects of the object, or might be simply surface features differentiating the object from similar ones, but which are also random. For complex objects, the best features to identify in a review tease out the complexity at the heart of the objects; unfortunately, these features can be extremely hard for a reviewer to identify in a short space and quick review period.

We have briefly discussed the ways in which targetting reviews for their audiences has an AIT formulation, and have also described why assessing creative and uncreative objects are both hard tasks. While our approach does not yield practical implementations, it gives a proper theoretical underpinning for a central task of the creative world.

Acknowledgments

The work of D.B. is supported by a Discovery Grant from the Natural Sciences and Engineering Research Council of Canada. The work of M.P. is supported by the University of Kent GTA Studentship Award, Prins Bernhard Cultuurfonds, Hendrik Mullerfonds, and Vreedefonds. We are grateful to Tiasa Mondol for useful conversations, and to a somewhat anonymous reviewer, whose peer review comments were extremely helpful in improving this manuscript.

References

Boden, M. 1992. *The Creative Mind*. London: Abacus.

Britten, B. 1943. *Rejoice in the Lamb. Op. 30*. Boosey and Hawkes.

Brown, D. G., and Mondol, T. 2021. On the problem of small objects. *Entropy* 23(11):1524.

Charnley, J.; Pease, A.; and Colton, S. 2012. On the notion of framing in computational creativity. In Maher, M.; Hammond, K.; Pease, A.; Pérez y Pérez, R.; Ventura, D.; and Wiggins, G., eds., *Proceedings of the Third International Conference on Computational Creativity*, 77–81. Third International Conference on Computational Creativity, ICCC 2012 ; Conference date: 30-05-2012 Through 01-06-2012.

Craik, K. 1943. *The Nature of Explanation*. Cambridge: Cambridge University Press.

Fisher, D. H., and Shin, H. 2019. Critique as creativity: Towards developing computational commentators on creative works. In *ICCC*, 172–179.

Guckelsberger, C.; Kantosalo, A.; Negrete-Yankelevich, S.; and Takala, T. 2021. Embodiment and computational creativity. In de Silva Garza, A. G.; Veale, T.; Aguilar, W.;

and y Pérez, R. P., eds., *Proceedings of the Twelfth International Conference on Computational Creativity, México City, México (Virtual), September 14-18, 2021*, 192–201. Association for Computational Creativity (ACC).

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

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

Mondol, T., and Brown, D. G. 2021b. Computational creativity and aesthetics with algorithmic information theory. *Entropy* 23(12):1654.

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 and Machines* 17:76–99.

Roberts, J., and Fisher, D. 2020a. preview: The artificially intelligent conference reviewer. In *2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 665–668.

Roberts, J., and Fisher, D. H. 2020b. Extending the philosophy of computational criticism. In *ICCC*, 358–361.

Schmidhuber, J. 2010. Formal theory of creativity, fun, and intrinsic motivation (1990–2010). *IEEE Transactions on Autonomous Mental Development* 2(3):230–247.

Stiny, G., and Gips, J. 1978. *Algorithmic aesthetics: computer models for criticism and design in the arts*. Berkley, California: University of California Press.

Yuan, W.; Liu, P.; and Neubig, G. 2022. Can we automate scientific reviewing? *J. Artif. Intell. Res.* 75:171–212.

Ziv, J., and Lempel, A. 1978. Compression of individual sequences via variable-rate coding. *IEEE Transactions on Information Theory* 24(5):530–536.