Formalizing Human Ingenuity: A Quantitative Framework for Copyright Law's Substantial Similarity

Sarah Scheffler
sscheff@princeton.edu
Princeton University
Princeton, NJ, USA

Eran Tromer
tromer@cs.columbia.edu
Columbia University
New York, NY, USA

Mayank Varia
varia@bu.edu
Boston University
Boston, MA, USA

ABSTRACT

A central notion in U.S. copyright law is judging the *substantial similarity* between an original and an (allegedly) derived work. Capturing this notion has proven elusive, and the many approaches offered by case law and legal scholarship are often ill-defined, contradictory, or internally-inconsistent.

This work suggests that key parts of the substantial-similarity puzzle are amendable to modeling inspired by theoretical computer science. Our proposed framework quantitatively evaluates how much "novelty" is needed to produce the derived work with access to the original work, versus reproducing it without access to the copyrighted elements of the original work. "Novelty" is captured by a computational notion of description length, in the spirit of Kolmogorov-Levin complexity, which is robust to mechanical transformations and availability of contextual information.

This results in an actionable framework that could be used by courts as an aid for deciding substantial similarity. We evaluate it on several pivotal cases in copyright law and observe that the results are consistent with the rulings, and are philosophically aligned with the abstraction-filtration-comparison test of *Altai*.

1 INTRODUCTION

In 2021, the U.S. Supreme Court ruled on *Google LLC v. Oracle America, Inc.* [37], a case involving copyright of computer code. This ruling was the culmination of nearly a decade of litigation, and it exposed crucial ambiguities in interpretations of many fundamental aspects of copyright law surrounding the copyrightability of Application Programming Interfaces, the merger doctrine, and the dichotomy between ideas and expression in copyright law. While this high-profile case was ultimately decided based on the fair use doctrine, the fundamental ambiguities impact a growing number of copyright and computer software cases.

U.S. copyright law is complex, and accordingly any specific copyright case can turn on any of a myriad of considerations. Some copyright cases are straightforward matters of testimony or feasibility. Some cases depend on whether the defendant had access to the plaintiff's intellectual work when producing its But many cases turn on a "test" that compares the original and derived intellectual works to see if they are *substantially similar* in their creative *expression*.

These tests generally rely on two related concepts. First, they emphasize human expression in intellectual works (of literature, music, art, and other tangible media) as distinct from a fact, abstract concept, or scientific idea; only the *expression* is copyrightable [8]. Second, U.S. courts apply the standard of *substantial similarity* to decide whether infringement has occurred; at a high level, both literal copying or producing a derived work that depends too heavily on the original run afoul of copyright law. These two considerations are intervowen: Judge Learned Hand famously depicted the challenge as determining "when an imitator has gone beyond copying the 'idea,' and has borrowed its 'expression"' [72].

Alas, legal doctrine and case law are notoriously unclear on the definition of such substantial similarity. Judges and legal scholars have lamented about the lack of a consistent test. For example, Balganesh remarks that the "'substantial similarity' requirement['s] ...structure, scope, and purpose continue to confound courts and scholars," and that in spite of its "centrality ...to copyright law, its complexity renders it a virtual black hole in copyright jurisprudence" [9]. Lim makes a similar comment, calling substantial similarity "copyright infringement's black box" [60]. Roodhuyzen observes that "[t]here is a great amount of confusion among courts and commentators as to what the proper test is for determining whether two works are 'substantially similar' so as to constitute copyright infringement" [74]. And this is true in spite of the fact that there is no shortage of proposed tests by courts and scholars to systematize the substantial similarity requirement (which we will describe in §2). Indeed, Samuelson states that "it is problematic that there are so many different tests and so little guidance about which test to use when" [76] and Stanfield critiques that the tests "are not a means to determine similarity, but rather a means to explain a finding of similarity that is determined in such a way that defies clear explanation" [90].

Our objective. The goal of this work is to create, justify, and use a new computer science-inspired framework that can test whether one work is substantially similar to another. We emphasize upfront that having a test of substantial similarity does *not* eliminate the complexity of copyright cases, and it is neither our claim nor our goal to "automate" the legal system. On the contrary, our intention is to form a more objective, predictable standard of determining substantial similarity precisely in order to free up the courts' time to ponder more difficult questions, such as the access and fair use questions at the heart of *Google v. Oracle* [37].

From law to computer science and back again. So why the need for yet another test of substantial similarity? We offer three responses to this question, two from a legal perspective and one from a computer science viewpoint.

From a legal perspective, we claim that previous works tried to solve too onerous of a problem: constructing a test that judges or juries can use on their own to evaluate substantial similarity. Instead, we suggest a framework that leverages the adversarial system of justice: the litigants should put forward their best argument as to why copyright infringement did/didn't occur. We require that an

impartial party can easily determine which argument carries the day, but we do not require that the plaintiff and defendant's tasks are simple. The adversarial approach incentivizes both parties to present the best arguments they know, and applies a quantitative metric to weigh these arguments.

The question then arises: what quantitative metric is appropriate? Some natural candidates turn out unsuitable. Notions of conditional entropy and mutual information reason about random variables, rather than individual strings, and thus require modeling a counterfactual probability distribution — with little connection to copyright law. Common string difference notions, such as edit distance and earth mover's distance, compare strings at too superficial a level, and are trivially foiled by reshuffling, translating to a different language, etc. Instead, we utilize a *minimum description length notion* (specifically, conditional Levin-Kolmogorov complexity [51, 57–59]), which is both string-based and, by definition, highly robust to mechanical transformations and external context. (This metric is also, in principle, infeasible to compute precisely; however in the law it is perfectly acceptable for it to be challenging to craft a strong argument, and furthermore we argue that a "fair" upper bound is provided by the truthful history of producing the works.)

This brings us back to the legal perspective, to ascertain that the mathematical formalism is not merely elegant, but also adheres to the letter and spirit of the law. Indeed, we show that our test agrees with several landmark U.S. Supreme Court decisions in copyright law. Additionally, we discuss why our definition comports with concepts in U.S. copyright law such as the idea-expression doctrine, avoids the overturned "sweat of the brow" doctrine, and more.

Overview of our description-length framework. Our novel methodological framework reasons about substantial similarity (i.e., the amount of expression that was copied) by using description length. Specifically, given an original work x and an allegedly-copied work y, we require that the plaintiff and defendant present computer programs that generate y with and without access to x, respectively. Intuitively, the plaintiff's program P describes how she thinks the defendant infringed on her copyright by producing y from x, while the defendant's program R describes how she constructed y without access to x. Then, we define the *derivation similarity* as the difference in description lengths of these two programs P and R. This metric captures the advantage gained by utilizing the copyrightable elements of x in producing y. By using a Levin-style description length [57, 59], our framework accounts for both the program's description length and any brute-force searches performed.

The resulting definition (see Def. 3.3) has its roots in the abstraction-filtration-comparison method of *Computer Associates, Inc. v. Altai* [20]. It is is easy for an impartial judge to evaluate, and therefore we believe it can be used by the courts. In fact, our framework *requires* that its inputs be created by the existing court system. While our framework reduces the large, nebulous question of "what is derivative work?" into a collection of smaller, better-defined questions, they cannot all be handled using our computer-science metric. For instance, care and legal knowledge are required to identify the inputs to our system, namely, the copyrightable and non-copyrightable aspects of x and y; we require the courts to adjudicate this task (see §3.2 for details). This is consistent with the approach in *filtering* where the court (not a jury) performs filtration [70, §8.6.2] and often relies on expert testimony when doing so [35]. Once these inputs have been identified, our framework greatly simplifies the next step of *comparison* once the filtration step has been completed.

Our contributions. In summary, we make the following contributions in this work.

- In §2, we provide a brief taxonomy of existing copyright infringement tests (after providing a primer on copyright law for computer scientists).
- In §3, we contribute our quantitative metric of *derivation similarity*, which quantifies the extent to which one work is substantially similar to another.
- In §4, we validate and justify our frameworks by showing that its reasoning is consistent with the decisions in a wide variety of landmark court cases involving copyright law over the past century. Additional, hypothetical, stress-tests are discussed in Section 5.
- In §6, we provide a simple cryptographic algorithm that allows the plaintiff and defendant to compute our derivation similarity metric privately, so that only the result is revealed in the public record.

Limitations. This paper only considers copyright law in the United States; though measuring the degree of similarity between two works is a core principle in any copyright law, so the ideas might apply in additional jurisdictions. Our framework focuses on substantial similarity, and does not cover other aspects of copyright law such as the fair use doctrine or the Digital Millenium Copyright Act (DMCA) anti-circumvention measures. Also, our framework presently does not encompass randomized algorithms, and only partially handles hidden information or trade secrets [11]; see §3.3 and §6 for details. For these and other reasons, this quantitative framework should not be used as a method for determining whether there is copyright infringement overall; rather, it answers only the narrow question of whether two works are substantially similar.

Relationship to prior works. This works joins recent scholarship on using concepts and modeling principles from computer science to reason about law and policy questions in a new light [5, 17, 18, 34, 44, 68, 69, 81]. Other classes of CS and Law cross-disciplinary research include developing and deploying new computer science tools for realizing social or legal goals (e.g., [11, 33, 36, 46, 49]), and policy discussions about the appropriate uses and limits of computer science technology in society (e.g., [4, 12, 28, 45, 53]).

Additionally, this work connects to a diverse landscape of court decisions and legal scholarship surrounding copyright law, which is too broad to do justice to in this space. The United States Congress' power to grant copyright protections is listed in Article I, Section 8 of the U.S. Constitution, with laws including the Copyright Act of 1976 [3] leading to its current statutory form that is codified within Title 17 of the U.S. Code [93]. There are hundreds of Supreme Court and appellate court decisions that have shaped the way that courts interpret copyright law (e.g., [7, 8, 20, 23, 25, 29, 31, 37, 42, 42, 43, 52, 62–65, 72, 75, 82, 85, 86, 89]); we describe several landmark cases in §4 and show that our framework agrees with the

decisions in all of these cases. Additionally, there are several casebooks (e.g., [15, 30, 39]) and law journal articles (e.g., [9–11, 54–56, 61, 71, 74, 76]) that describe and reflect the modern understanding of copyright law. In the next section, we explore the landscape of copyright law and the tests that have been previously proposed to make sense of the "substantial similarity" requirement.

2 BACKGROUND ON COPYRIGHT LAW

Title 17 of the U.S. Code §102 states that "[c]opyright protection subsists . . . in original works of authorship . . . In no case does copyright protection for an original work of authorship extend to any idea, procedure, process, system, method of operation, concept, principle, or discovery, regardless of the form in which it is described, explained, illustrated, or embodied in such work" [1]. Originality is a requirement — the original work must "possess some creative spark" [29, l. 345], but "the requisite level of creativity is extremely low; even a slight amount will suffice" [29, l. 345]. The copyright-holder holds the exclusive right to copy, distribute, or produce derivative works based on the copyrighted work [2], with some exceptions described in 17 U.S.C. §107-122 (including "fair use," which we do not discuss further here). Thus, the usual topic in a copyright case is to determine whether or not the *expression* in the defendant's work y was "derived from" the expression within the plaintiff's copyrighted work x (whereas reuse of *ideas* represented by those expressions is always permissible [1, 29]).

Sometimes this question is straightforward. Most instances when y is a literal copy of x are found to be copyright infringement (although if two authors truly did create the same work independently, there was no infringement [29, 65, 85]). However, most of the time y is not a literal copy of x. Many courts require that there be *substantial similarity* between the works to demonstrate copying, although there is no general agreement on what exactly this means (see e.g. [19, 32, 56, 67, 70, 76]). Each court applies the idea slightly differently, and the approaches to these methods have changed over time. We proceed to describe some of these tests for determining whether two works' expressions are substantially similar.

2.1 Copyright tests

We provide only a brief summary of the most important existing copyright tests here; for more details see [67, 70, 76].

The Ordinary Observer Test. One of the most widely-applied copyright tests, the Second Circuit's approach in *Arnstein v. Porter* [7] in 1946 has been adopted with minor variations by the First, Third, Fifth, and Seventh circuits as well [70, §3.1]. Its analysis combines two elements: The court must determine whether (1) the defendant copied from the plaintiff's copyrighted work, and (2) if that copying went so far as to be improper [7, l. 468]. In determining the first question, Judge Frank wrote in the case that extreme similarities between the works were suspicious, even if evidence of access to the original work was slight: "a case could occur in which the similarities were so striking that we would reverse a finding of no access, despite weak evidence of access (or no evidence thereof other than the similarities); and similarly as a to a finding of no illicit appropriation" [7, l. 469], and the latter question involves an "ordinary observer" to determine if the copying "resulted in substantial similarity" [21, l. 608]. Samuelson describes difficulty in applying this test because the key terms are used differently in the two parts of the test [76], and Lemley asserts that the the methods for resolving the two questions should be reversed: an ordinary observer could detect copying, but the question of whether or not that was improper could require expert testimony or analysis [56].

The Extrinsic/Intrinsic Test. Sid & Marty Krofft Television Productions, Inc. v. McDonald's Corp, a 1977 case from the Ninth Circuit, separated out the "extrinsic" elements like the work's type, materials, subject matter, setting, etc, from "the observations and impressions of the average reasonable reader and spectator" [86, 92], which they dub as the "intrinsic" portion of the test. While originally intending to compare similarities in ideas and expression respectively, those tests were later interpreted to refer to "objective" and "subjective" similarities (see e.g. [84]). The intrinsic or subjective part of the test is also highly similar to the "total look and feel" test from the same circuit in 1970, which essentially asks whether "the work is recognizable by an ordinary observer as having been taken from the copyrighted source" [75, l. 1110]. These approaches have been criticized for making it too easy to include uncopyrightable elements in the similarity analysis [76] and that the overall "feel" of a work may make sense in the case of a small work like a greeting card, but not in a larger work like a large body of computer code [67]. Nevertheless, the Fourth and Eighth Circuits both perform some variant of this test, and the Eleventh has done so for some cases [70].

The Abstraction, Filtration, Comparison Test. This test emerged from *Computer Associates, Inc. v. Altai* for the purpose of determining the similarity of computer code. In the framework, one begins by "dissect[ing] the allegedly copied program's structure and isolat[ing] each level of abstraction contained within it," ending with "an articulation of the program's ultimate function" [20, l. 707] (this abstraction step had previously been applied to other works like plays [65]). Secondly, one should conduct a "successive filtering method" [20] to determine whether the inclusion of a particular element of one level of abstraction was due to "idea" or "expression," and remove all the elements that were due to ideas, efficiency constraints, external compatibility, or from the public domain [20, l. 709-710]. Finally, one should compare what is left after the filtration process is done — all remaining similarities should be similiarities of expression.

Although this test was not without its critiques and wrinkles (e.g. [16, 26, 41, 77, 79, 88]), the general approach has stood the test of time and is the primary tool for the courts to rule on software copyright cases [70, §8.6]. It is used for all cases in the Tenth Circuit, and in the Sixth Circuit as well with a slight variation [70, §3.3]. The D.C. Circuit also practices a filtration step before reverting to something more akin to the ordinary observer test [70, §3.3.3]. Other courts have also adopted filtering tests like *Altai* into other settings like advertisements [73, 91], architecture [47], child safety locks [50], and dolls [22].

2.2 Where our framework fits in

In the abstraction-filtration-comparison approach of *Altai*, determining which elements may be "filtered out" is a challenging question requiring expertise in copyright law and the subject matter. However even if the "filtering" step was done perfectly, it is still challenging to rigorously state in the "comparison" step how much illegal copying occurred. This is often considered a "value judgement, involving an assessment of the importance of the material that was copied" [66, Vol. 3 §13.03]. The copied expression must be "important" to the copied work even though it may not be a large portion of it [70, §8.6.3].

Our approach provides a way to reduce the qualitative question of "how important" the copying was to a quantitative assessment that may be determined after the filtration step was completed. We do *not* take the approach of measuring the portion of the derived work that was based on copyrighted aspects of the original work, nor do we measure the simple distance between the works, nor the work required to generate those works. Instead, we use a metric based on program description length which measures the advantage gained when building the allegedly-copied work both with and without access to the original work (see Section 3). We argue that this method captures this "importance," and we hope that this will enable a simpler "comparison" step of the test, freeing up the court's time to focus on the more challenging step of "filtering."

3 DEFINING DERIVATION SIMILARITY

In this section, we describe our formal framework for modeling copyright questions, and we provide our mathematical test of *derivation similarity* that can be used to help determine whether copyright infringement occurred. To ensure that our eventual definition is built on solid computer science foundations, we begin in §3.1 with some formal definitions that measure the cost and complexity of clean room productions of the derived work. Then, we provide our framework for reasoning about copyright infringement in §3.2. Finally, we discuss the strengths and limitations of this definition in §3.3.

We emphasize that §3.1 is only needed to understand "the math." Readers who are focused on the high-level conceptual ideas contained within our copyright infringement definition can safely skip directly to §3.2.

3.1 Some Computer Science Formalism

In this subsection, we include some formal definitions about the cost of a Turing machine and the complexity of a string. This subsection is only needed to provide rigorous computer science foundations for the eventual copyright definition to follow in §3.2.

We begin by formally defining the "cost" of a producer algorithm P or reproducer algorithm R. We assume throughout this work that all algorithms are deterministic.

Comparable. We use $x \cong y$ to denote that x and y are *comparable*. In all cases we consider below, we define comparability using exact equality. That said, we leave open the possibility that one might choose a different notion of comparable strings (e.g., being within a certain Hamming distance) depending on the situation in an individual case. In the general case, we consider $\cong$ to be a binary relation on strings that need not be an equivalence relation; we only require that the relation is reflexive (i.e., every string is similar to itself).

Definition 3.1 (Conditional Levin-Kolmogorov Cost)**.** Let M, b, and y be strings in $\{0,1\}^*$. Let $U(M,b,t)$ be the output of a universal prefix Turing machine running program M for t timesteps, where M has access to input tape b. Then, the cost of running M on input b to get a string "comparable" to y is

$$C(M,z \mid b) = \min_{t\in\mathbb{N}}\{|M| + \lceil \log t \rceil : z \leftarrow U(M,b,t)\}$$
$$\tilde{C}(M,y \mid b) = \min_{z\cong y}\{C(M,z \mid b)\}$$

The first cost metric C is taken from the starting point of Levin's cost for universal search [57, 58] (which itself is a variant of Kolmogorov's complexity [51]) as presented in Def. 7.17 of Vityáni and Li [59], and adding a conditional variant as used on pages 412-413 of the same. Our new extended metric $\tilde{C}$ provides additional flexibility by allowing for the possibility of (re)producing a similar string that is easier to generate.

In our analyses, we will also sometimes be interested in the *minimum description length* of a program M that can output a string y, conditioned on some "free inputs" that M can freely use but is not charged any cost for receiving. Importantly, this complexity metric is well-defined for a single string y, and it does not require the existence (much less an understanding of) a distribution from which y is generated.

Definition 3.2 (Conditional Levin-Kolmogorov Complexity)**.** Let b and y be strings in $\{0,1\}^*$. Let $U(M,b,t)$ be the output of a universal prefix Turing machine running program M for t timesteps, where M has access to a tape b. Then

$$\mathsf{LK}(y \mid b) = \min_{M\in\{0,1\}^*} \tilde{C}(M,y \mid b).$$

3.2 Our Copyright Definition

In this subsection, we describe our abstract framework of the salient features in most copyright infringement cases, and we provide our main definition that separates the novel vs. copied creative expression in a derived work. Our framework considers an original work x and derived work y, each of which contain copyrightable elements of creative expression.

Our objective is to compare the innate expression required to create the supposedly-infringing derived work y both with and without access to the copyrightable elements of the original work x. In more detail, we want to understand the *ex post* question about how "simple" it can be to create y for a clean room actor working as effectively as possible, both with and without x. Our cost metric from §3.1 provides a way to do this. To avoid reducing our definition to a mere measurement of the "sweat of the brow," we also provide relevant facts and non-copyrighted works to the clean room actor for free.

We formalize this definition in two ways: as a standalone test that a court can apply on its own to determine the likelihood that

copyright infringement occurred, or (more interestingly) as an interactive game between the participants that is amenable to the adversarial system of justice used within courts in the United States. In the latter viewpoint:

- The plaintiff provides the smallest possible "Producer" algorithm P capable of independently producing something comparable to y, given full access to the original work x (and some other inputs we describe below).
- The defendant provides the smallest "Reproducer" algorithm R that produces a work comparable to y *without* using the copyrightable parts of x, i.e., in the manner that the defendant alleges that she initially created y.

By comparing the cost of P and R, our definition will measure the extent to which the derived work y intrinsically relies on the creative expression within x, as distinct from the independent expressive content that may also be present in y.

To focus on the copyrighted creative elements only, our framework also explicitly defines the following non-copyrighted materials such as basic facts or public domain works.

- The non-copyrightable aspects of the original work x, which we denote $\mathsf{noncopy}(x)$.
- The non-copyrightable aspects of the derived work y.
- A context comprising any other relevant works of interest to the plaintiff and defendant.

We collectively refer to these non-copyrighted works as the background $\mathsf{bg} = (\mathsf{noncopy}(x), \mathsf{noncopy}(y), \mathsf{context})$. Looking ahead, our definition will give P and R these background materials at zero cost when (re)producing the derived work y.

We require that the plaintiff and defendant agree upon the background material, before they attempt to use our definition. This might sound counterintuitive: after all, how can the parties agree on the non-copyrighted aspects of x and y when they are litigating whether y itself infringes upon x?

The idea here is that the plaintiff can scope her own claim: by conceding to the non-copyrighted aspects of x and y, the definition will hone in on the inherent expression contained within the remaining parts of x and y. The plaintiff's goal is to identify a concrete portion of y that bears a strong resemblance to (a portion of) x, in the sense that a clean room implementation of y would gain a substantial advantage by having access to the copyrightable parts of x. Conversely, the defendant's goal is to show that a clean room, armed with the background facts and works, could have created y almost as easily with or without x. (See §3.3 for more details.)

Our formal definition simply takes the difference between these two costs. We first present our empirical definition, which is an adversarial game between a plaintiff who produces a single producer algorithm P and a defendant who produces a single reproducer algorithm R. We call this notion *derivation similarity* to indicate that it is a quantitative attempt at measuring substantial similarity based on the difference in P and R's ability to derive y given their respective inputs.

Definition 3.3 (Empirical Derivation Similarity)**.** Let x and y denote the original and derived works, respectively. Given a producer algorithm P and a reproducer algorithm R, the *empirical derivation similarity* is defined as:

$$\mathsf{DerSimEmp}(x, y, P, R \mid \mathsf{bg}) = \tilde{C}(R, y \mid \mathsf{bg}) - \tilde{C}(P, y \mid (\mathsf{bg}, x)),$$

where $\mathsf{bg} = (\mathsf{noncopy}(x), \mathsf{noncopy}(y), \mathsf{context})$ contains the aspects of x, y, and relevant other works that the parties agree are non-copyrighted.

Our second definition is a standalone test that a court could apply on its own. It assumes an "optimal" plaintiff and defendant who provide the lowest-cost P and R algorithms.

Definition 3.4 (Theoretical Derivation Similarity)**.** Let x be the original work. Let y be the derived work which the plaintiff claims infringes on x. We compute the *theoretical derivation similarity* as:

$$\begin{aligned}\mathsf{DerSim}(x, y \mid \mathsf{bg}) &= \min_R \max_P \mathsf{DerAdvEmp}(x, y, P, R \mid \mathsf{bg}) \\ &= \mathsf{LK}(y \mid \mathsf{bg})) - \mathsf{LK}(y \mid (\mathsf{bg}, x))\end{aligned}$$

3.3 Discussion

In this subsection, we briefly describe several important concepts that influence the design of our mathematical definition as well as limitations of our framework.

Sliding scale of similarity. Our definition purposely does not provide a binary 'yes/no' test about whether copyright infringement occurred. Instead, the output of DerSim is a number in between the following extremes:

- 0 if x and y have no correlation.
- $\mathsf{LK}(y \mid \mathsf{bg})$, the Levin-Kolmogorov complexity of the string y, if $x = y$ (i.e., blatant copying occurred).

The goal here is to provide guidance to the courts to inform decisions about infringement. We leave it to the courts to decide in any particular case what the cutoff points should be for determining that x and y bear substantial similarity or striking similarity. In our application of the definition to prior court cases in §4, we describe why we think it is reasonable to conclude that DerSim is "too large" or "small enough" to conclude whether copyright infringement did or didn't occur.

Considerations when applying our definition. Our framework has some limitations that influence whether and how it should be applied in court cases.

First, we assume that all relevant background information bg is available to both the plaintiff and defendant. As a result, our definition cannot be applied in cases where one party has trade secrets or other hidden information that would facilitate its clean room (re)production.

Second, we remark that the context is vulnerable to "gaming" by a party to lower its conditional cost. For instance, the defendant might ask to include a 2-out-of-2 secret sharing of y in the context (e.g., two strings y_1 and y_2 that individually look random and devoid of creative expression, but with $y = y_1 + y_2$) in order to reproduce y at very low cost. We rely on the adversarial system of justice to call out such abuses, and we stress that our definition only applies after consensus has been reached on the background bg. Generally speaking though, the intent is for context to contain the union of the plaintiff and defendant's desired prior works, as long as they are not "gamed" in this way.

Finally, we note that the theoretical derivation similarity DerSim is uncomputable in general, because the Levin-Kolmogorov complexity has this same defect. For this reason, the main value of theoretical derivation similarity is in analyzing what we believe to be the likely outcome of court cases in our analysis in §4-5. If this test is adopted by courts, we recommend using empirical derivation similarity in practice.

Why minimum description length. Our derivation similarity metric reasons about the "amount of copying" by comparing the (minimum) description length of the plaintiff and defendant's programs. Here, we justify the use of a description-length metric as compared to other potential choices for the amount of copying, and then our specific choice of Levin-Kolmogorov complexity as a description-length metric.

Regarding alternative metrics: we chose to avoid measurements of cost based on running time because copyright doctrine is clear that the "sweat of the brow" necessary to generate the work is irrelevant to the question of copyright [29, l. 359-360]. Another seemingly-natural choice for measuring the "amount of copying" in a work is mutual information, which measures how many bits of information the value of one random variable tells you about a second random variable. The challenge is that mutual information applies to the probability distribution of all works that "could" have been created. But, courts only have access to the actual strings x and y rather than these counterfactual options, and even if convincing evidence could be provided about the distribution, there is not an established precedent for considering these counterfactual alternatives. This also rules out other distribution-based tools, such as cryptographic notions of indistinguishability.

We also chose to avoid more naive notions of string comparison like edit distance, for two reasons. First, edit distance is better suited to measuring the "portion" of the work that is copied, rather than its "importance" (see §2.2). Second, there is not a good way to incorporate the expression/idea dichotomy into these measures; that is, our framework must be able to identify that a large amount of expression is copied in a literally-copied novel, but a low amount of expression is copied in a literally-copied compilation of facts. This property would also make it challenging to adjudicate correctly on merger doctrine cases (see §4.2).

Among description length metrics: an alternative natural choice is to consider Kolmogorov complexity, which removes the additive $\lceil \log t \rceil$ term from Def. 3.1 and therefore only measures the size of the minimum-length Turing machine M. We opt to use Levin complexity instead, for two reasons. Philosophically, we want parties to provide an explicit description of their (re)production process; we don't want a test that incentivizes the parties to perform a brute force search that simply tries all possible options until they get the right answer. Practically, we don't want the existence of public-domain hash function digests to influence the substantial similarity test; see §5.2 for details.

Numerical stability. As with many computational, quantitative metrics, one challenge to consider when using our $\tilde{C}$ cost metric is numerical stability. That is: if the strings x and y both have very large Levin-Kolmogorov complexity, then it might be possible to "sneak in" extra effort into the programs P or R with very little net increase in their cost. Put simply: our derivation similarity metric doesn't work well when the costs on both sides are large.

Numerical instability harms the plaintiff, since by construction the defendant's task in Def. 3.3 is necessarily harder and it is the plaintiff's goal to ensure that this difference is noticeable. To resolve the numerical stability issue, we rely on the plaintiff's ability to declare aspects of x as non-copyrighted and thus available "for free" to P and R, even materials that might actually be copyrighted in practice, to scope the copyright claim down to the (hopefully small amount of) infringing material and mitigate this numerical stability concern.

4 ADHERENCE TO PRECEDENT

In this section, we apply our framework to several prominent historical copyright cases, both to demonstrate that our framework reaches the same result as the courts in all cases, and also to illustrate how reasoning with the framework works. We begin with cases that illustrate the *idea vs. expression* dichotomy in §4.1, which in our framework concerns what material belongs in $\text{noncopy}(x)$. We then proceed in §4.2 to analyze cases which illustrate our framework's approach to the *merger doctrine* — our framework does not require applying any special reasoning in this case; the best P and R in these cases will simply operate via reasoning that resembles the merger doctrine. We end by considering cases that use computer code and describing our framework's relationship to the *Abstraction-Filtration-Comparison* framework, in §4.3.

4.1 Idea vs. Expression

Feist Publications, Inc. v. Rural Telephone Service Co., 499 U.S. 340 (1991). Feist [29] famously resolved the seeming tension arising from the notion that facts are not copyrightable, but compilations of facts can be. It also corrected a misconception in lower courts by stating that copyright is *not* a reward for the "sweat of the brow" that went into creating the work. Rather, copyright only extends to *original* works, meaning those that are "independently created by the author, and possess[] some minimal degree of creativity" ([29, l. 345] citing [66, Vol. 1 §2.01]) and "[n]o one may claim originality as to facts" ([29, l. 347] citing [66, Vol 1 §2.11]). Thus a compilation of facts may be copyrightable, but only if they are arranged or selected in an original manner, making this protection "thin" [29, l. 349].

In this famous case, Feist Publications published a telephone directory that used several thousand entries of an existing copyrighted phone book, Rural Telephone Service Co., including 1,309 identical entries out of 46,878. The court found no "copying of constituent elements of the work that are original" [29, l. 361], since neither the raw facts themselves, nor Rural's arrangement and selection of those facts (name, town, and telephone number of each person in Rural's telephone service, listed alphabetically by last name), were original.

Recall that our framework compares the size of the smallest possible P and R that generate works similar to y — in this case, we take our similarity metric $\cong$ to be exact equality. Recall that P has access to all of x, and R has access only to the noncopyrightable aspects of x. In modeling the facts of *Feist*, because the facts and their selection and arrangement in Rural's phone book were not

original, nearly the entire text of Rural's phone book is a part of $\mathsf{noncopy}(x)$. Feist also copied four fictitious listings that Rural had included expressly to detect copying; although the court in *Feist* did not address this matter directly, most courts treat fictitious facts as facts and do not afford them copyright protections (see e.g. [24], though this approach has not been applied perfectly consistently [87]). In our framework we treat these as elements of $\mathsf{noncopy}(x)$, but not context.

The court did not rule on whether or not Feist's phone book is copyrightable — again, neither the facts themselves nor the arrangement of those facts were original, but since Feist had more freedom in how it chose which entries to include in its book in the first place. (Rural was a monopoly telephone provider in a local area and its phone book included only the phone numbers it provided plus the small number of fictitious entries — about 7,700 total. Feist covered 11 telephone service areas across 15 counties, for about 47,000 total.) To model this, as shown in Table 1, we could consider context to contain a much larger collection of facts arranged and chosen in an un-original way — perhaps every phone number, name, and address in the state (the sorting mechanism does not matter, it will change P and R's size by the same amount, but it is likely that sorting by county and then alphabetically will yield the smallest results). $\mathsf{noncopy}(y)$ contains a list of the counties from which the In order to select the facts to go into the Feist phone book, the P and R must generate the set of indices within context of the entries in Feist's phone book. Crucially, both P and R must do this — nothing in $\mathsf{copy}(x)$ aids this process.

Thus, the difference in description length between R and P is low and it is reasonable to conclude that no copyright infringement occurred.

Baker v. Selden, 101 US 99 (1880). In this case [8], Selden had published a book entitled *Selden's Condensed Ledger, or Book-keeping simplified* and several other books on the topic of his book-keeping method. He alleged that Baker infringed on those books' copyright by publishing his own book describing a similar book-keeping method, and especially using similar illustrations to describe the technique.

This case is key in describing the difference between copyrighting an expression of the system versus the system itself. They state "By publishing the book, without getting a patent for the art [the system], the latter is given to the public. The fact that the art described in the book by illustrations of lines and figures which are reproduced in practice in the application of the art, makes no difference. Those illustrations are the mere language employed by the author to convey his ideas more clearly. Had he used words of description instead of diagrams (which merely stand in place of words), there could not by the slightest doubt that others, applying the art to practical use, might lawfully draw the lines and diagrams which were in the author's mind, and which he thus described by words in his book" [8, l. 103]. Our analysis of this case, shown in Table 2, describes how since although the book-keeping method described in y is similar to the one described in x, once those book keeping methods are provided separately in $\mathsf{noncopy}(y)$ and $\mathsf{noncopy}(x)$, the remaining expression in x does not aid the creation of y.

Southco, Inc. v. Kanebridge Corp., 390 F.3d 276 (2004). In this case, Southco, Inc. had created a part numbering system to uniquely identify and describe characteristics of parts like screws. Kanebridge Corp. had labeled its screws both with its own part numbering system, and with Southco's numbering system, to describe that the parts were interchanbeable. The court ultimately found that the part numbers are not copyrightable, for two reasons. First, Southco had not claimed any copyright on the numbering *system* — in our framework, the system is in the context. And once the system is fixed, no creative options remain in picking the numbers for a particular screw — in our framework, $\mathsf{noncopy}(x) = \mathsf{noncopy}(y)$ contains a physical description of the screw, and $x = y$ is the part number for the screw itself, which could have been found by "looking up" the physical description in the parts numbering system. The second reason the numbers were not copyrightable was that they are akin to short phrases or titles, which were determined not to be copyrightable. We believe this to essentially set a bar for how much bigger R is allowed to be than P— if the difference involves a small amount of code to generate a "phrase," no copyright infringement has occurred.

4.2 The Merger Doctrine

Herbert Rosenthal Jewelry Corp. v. Kalpakian, 446 F.2d 738 (9th Cir. 1971). Herbert Rosenthal Jewelry Corp. designed and copyrighted a jeweled bee-shaped pin, and Edward and Lucy Kalpakian had "manufactured a jeweled bee alleged to be similar" [42]. The plaintiff especially focused on the arrangement of jewels on the original bee pin. However, the plaintiff was unable to describe exactly which aspects of the arrangement were original, nor what alternative arrangements would not infringe the copyright. The defendant's arrangement was "simply a function of the size and form of the bee pin and the size of the jewels used."

In our framework, this "function" can be done explicitly by the Reproducer — this can be thought of as a bin-packing problem, fitting jewels of different sizes into rows across the bee's surface. Note that even if the Kalpakians' arrangement of jewels had been identical to Rosenthal's, the maximum difference in the size difference between R and P is the size of the algorithm necessary to produce that arrangement.

Morrissey v. Procter & Gamble Company, 379 F.2d 675 (1st Cir. 1967). In *Morrissey v. Proctor & Gamble Co.* [64], Morrissey had written a promotion for a sweepstakes, and Proctor & Gamble Co. had later released a similar sweepstakes with some very similar text.

The court ruled that although Morrissey's expression did contain "original creative authorship," and "there was more than one way of expressing even this simple substance" [64, l. 678], the "uncopyrightable subject matter is very narrow, so that the topic necessarily requires ... if not only one form of expression, at best only a limited number" then copyright does not apply.

This would come to be a widely cited case establishing that the merger doctrine could apply to a case where there is more than one way to express some idea, but still only a limited number of ways to express the idea that are at least as reasonable [78].

Our framework is well-suited to handle the merger doctrine in this form. If there are only a limited number of ways to represent an idea, then those ideas may be enumerated by both P and R using only a very short code description, or possibly even enumeration

ENCODING: Assuming exact equality required. All entries converted to raw text, since aspects of the phone book such as the font or physical properties were not in dispute.	
x: Rural's phone book	NONCOPY(x): All entries in Rural's phone book, in alphabetical order
y: Feist's phone book	NONCOPY(y): The list of counties covered in Feist's phone book
CONTEXT: All phone numbers, addresses, and names in the state, sorted by county then by last name	
NOTES: Let L_{context} be a compressed list of indices of phone numbers in the context in y, with null characters separating the entries in uncompressed form. Let L_x be a similar alphabetized list of entries of entries within x (both real and fictitious), which as discussed are part of noncopy(x).	
BEST P: P must include the smallest possible compression L_{context} and L_x. Also includes code of constant length c to fuse the entries from context and noncopy(x) denoted in those lists, and write them in the output w in alphabetical order. This should take $\lvert L_{\text{context}}\rvert + \lvert L_x\rvert$ time. $\mathsf{LK}(y \mid (\text{BG}, x))$: $\lvert L_{\text{context}}\rvert \log \lvert L_{\text{context}}\rvert + \lvert L_x\rvert \log \lvert L_x\rvert + c + \log(\lvert L_{\text{context}}\rvert + \lvert L_x\rvert)$	BEST R: R must include L_{context} and L_x similar to P. Also includes code c' to copy the information from L_{context} and L_x into z in alphabetical order. Note that $c \approx c'$. The copying should take approximately $\lvert L_{\text{context}}\rvert + \lvert L_x\rvert$ time. $\mathsf{LK}(y \mid \text{BG})$: $\lvert L_{\text{context}}\rvert \log \lvert L_{\text{context}}\rvert + \lvert L_x\rvert \log \lvert L_x\rvert + c' + \log(\lvert L_{\text{context}}\rvert + \lvert L_x\rvert)$
OUTCOME: $\mathsf{DerSim}(x, y \mid \text{bg}) = (c' - c) \approx 0$. COURT OUTCOME: No infringement occurred.	

Table 1: Our framework's treatment of *Feist v. Rural*

ENCODING: Text and pictoral snapshots of the descriptions of each book and essay. Similarity taken to be exact equality.	
x: The exact text of "an introductory essay explaining the book-keeping system [in Selden's book], to which are annexed certain forms or blanks, consisting of ruled lines, and headings" [8, l. 100]	NONCOPY(x): The book-keeping methods described in x.
y: A series of books that describes a book-keeping method "similar … so far as results are concerned, but makes a different arrangement of the columns, and uses different headings" [8, l. 100]	NONCOPY(y): The book-keeping methods described in y
CONTEXT: An unrelated public domain book describing a book-keeping system.	
NOTES: Let $s(\text{bg})$ be the smallest possible algorithm which reads in the book-keeping methods in noncopy(x) and noncopy(y), and reads in the related book from context, and outputs y. Let $s'(\text{bg}, x)$ be the smallest possible algorithm which reads x in addition to the other inputs above. By assumption, conditioned on bg, y and x should not be much more similar than y and the unrelated book in context aside from the actual book-keeping method described, which is fully contained in noncopy(x) and therefore accessible to both P and R. Thus, we expect $\lvert s\rvert \approx \lvert s'\rvert$ and we expect their runtimes t and t' to be similar as well.	
BEST P: Runs $s(\text{bg})$ $\mathsf{LK}(y \mid (\text{BG}, x))$: $\lvert s\rvert + \log t$	BEST R: Runs $s'(\text{bg}, x)$ $\mathsf{LK}(y \mid \text{BG})$: $\lvert s'\rvert + \log t'$
OUTCOME: $\mathsf{DerSim}(x, y \mid \text{bg}) = \lvert s\rvert + \log t - (\lvert s'\rvert + \log t') \approx 0$ COURT OUTCOME: No infringement occurred.	

Table 2: Our framework's treatment of *Baker v. Selden*

ENCODING: Structured lists of jewel arrangements by type, size, arrangement, and color. Exact equality required.	
x: The arrangement of 19 white jewels on Herbert Rosenthal Jewelry Corp's jeweled bee	NONCOPY(x): The size and shape of x
y: A set of additional jeweled bees "in three sizes decorated with from nine to thirty jewels of various sizes, kinds, and colors" [42, l. 740]	NONCOPY(y): The size and shape of y
CONTEXT: A list of potential jewel colors and sizes	
NOTES: Let A be a compressed pattern of jewels on y. Let O be a compressed list of the indices of jewels that overlap between x and y. Conditioned on the idea of a jeweled bee, O is small as described by the court [42, l. 742]. We assume that the rumtime of algorithms placing the pattern A is approximately $\lvert A\rvert$ regardless of whether O is used or not.	
BEST P: The pattern of jewels on y must be hardcoded; in areas where there are overlaps with x, a strcopy can be used. $\mathsf{LK}(y \mid (\text{BG}, x))$: $\lvert A\rvert - \lvert O\rvert + \log \lvert A\rvert$	BEST R: The same pattern must be hardcoded, the only difference is that any overlaps with x cannot use copy. $\mathsf{LK}(y \mid \text{BG})$: $\lvert A\rvert + \log \lvert A\rvert$
OUTCOME: $\mathsf{DerSim}(x, y \mid \text{bg}) = \lvert O\rvert \approx 0$ by assumption COURT OUTCOME: No infringement occurred.	

Table 3: Our framework's treatment of *Rosenthal v. Kalpakian*

in the context, as appropriate. Moreover, our framework points to drawing a potential line of *how many* ways to represent an idea should be acceptable, since merger doctrine does not have to be treated any differently under our framework than any other copyright case.

4.3 Computer Code

There have been a number of rulings on what aspects of computer code are and are not copyrightable. *Apple computer, Inc. v. Franklin Computer Corp.)* [6] stated that object code (as opposed to source code) is copyrightable, though their rationale that copyright extends to works from which the expression "can be perceived, reproduced, or otherwise communicated, either directly or with the aid of a machine or device" may fall flat for obfuscated code. *Lotus Dev Corp. v. Borland Int'l* [62] additionally stated that menu hierarchies are not copyrightable, and *Sega Enterprises Ltd. v. Accolade, Inc.* [82] decided that dissasembly can sometimes fall under fair use, if the dissasembly only "reveals" un-copyrighted parts of the code and there is a "legitimate reason" for accessing this.

We examine some other cases in more detail.

Computer Associates International, Inc. v. Altai, Inc., 982 F.2d 693 (1992). Altai [20] introduced the *Abstraction-Filtration-Comparison* framework for determining whether computer programs are substantially similar. The test applies to the *original* work, not to the derived work. The test first describes the original program at different levels of "abstraction" ("At low levels of abstraction, a program's structure may be quite complex; at the highest level it is trivial" [20, l. 707]). Then, a "successive filtering method" [20, l. 707] separates copyrightable expression from non-copyrightable elements. For instance, elements dictated by efficiency or external factors (such as hardware specifications or compatibility requirements), or elements in the public domain, are removed. After this, whatever potentially-copyrightable elements remain of the original program are compared to the derived program.

Computer Associates (CA) sold a program entitled CA-SCHEDULER which was a scheduler for IBM mainframe computers. One component of CA-SCHEDULER was an interface called ADAPTER; instead of making system calls to the operating system directly, CA-SCHEDULER would call ADAPTER, which would correctly call whichever of the three possible operating systems used by IBM mainframe computers the main program was being run on. Altai, Inc. had a scheduler called ZEKE, which Altai wished to rewrite for use in a particular operating system. An employee at Altai offered a job to an employee of CA, Arney, to help write the new ZEKE version. When Arney joined Altai in 1984, he brought source code for ADAPTER, "in knowing violation of the CA employee agreements he had signed" [20, l. 700]. He also suggested that the best way to write multiple versions of ZEKE for different operating systems would be to have them call a common interface, without revealing that this idea stemmed from ADAPTER. Altai's version of the interface was called OSCAR, and ultimately about 30% of its code was copied from ADAPTER. A few years later, CA learned that much of the OSCAR code was copied, from ADAPTER, and contacted Altai. The company excised the ADAPTER-copied code and set several programmers who had not been on the OSCAR program before to rewrite it, excluding Arney from the process. The initial copying of ADAPTER into OSCAR was not in dispute. But CA continued to accuse the rewritten OSCAR of infringing the CA-SCHEDULER code for copying its "non-literal elements" [20, l. 701]. The court ultimately rejected CA's argument. After abstracting ADAPTER and filtering out the non-copyrightable elements, the code "presented no similarity at all" [20, l. 714]. The main elements at issue ended up being external constraints to ensure compatibility with the operating system.

Our framework essentially formalizes the latter two steps of the Abstraction-Filtration-Comparison framework. We also rely on manual abstraction or encoding of specific elements, and then filter the non-copyrightable elements into $\text{noncopy}(x)$. However, rather than "comparing" the remaining copyrightable info from x and y, we use a metric of description length to measure this instead. In our framework, x is the ADAPTER program and $\text{noncopy}(x)$ is the portions of ADAPTER that were made necessary by compatibility with the operating system. y and $\text{noncopy}(y)$ are similarly defined for the rewritten OSCAR program. The context contains a description of the operating system calls that both programs must adhere to.

5 TESTING THE LIMITS WITH HYPOTHETICAL EXAMPLES

In this section, we describe a few synthetic scenarios with new wrinkles that go beyond the precedent-setting cases from §4. Our objective in this section is to elucidate and stress-test elements of our definition, in order to further justify some of our modeling decisions.

5.1 Arrangements of non-original fragments can be original

In *Knitwaves v. Lollytogs*, the Second circuit stated that if they had to compare each individual element of the work instead of examining the work's "total concept and feel" [48, l. 1003], then taken "to its logical conclusion, [the court] might have to decide that 'there can be no originality in a painting because all colors of paint have been used somewhere in the past."' [48, l. 1003].

As the court goes on to point out [48, l. 1004], the fact that a work is composed of uncopyrightable elements does not preclude copyright over the *compilation* of those elements, as long as that compilation contains originality [29, l. 344]. In some sense any work can be broken down in this way — novels are "just" compilations of words or letters, digital images are "just" collections of pixels, and music is "just" arrangements of notes.

Thus, it is important that our framework ensures that, for example, representing a novel as "a collection of words" does not yield a smaller program P than hard-coding the novel itself does — we wish to ensure there is no mysterious advantage gained in representing a novel just because one has access to an unrelated reference text like a dictionary — or any other book, for that matter.

Suppose P naively read an index into the dictionary each time it needed to use a word. Depending on which words one chose to include, there are estimated to be between 400,000 and 1,000,000 words in the English dictionary, i.e. it requires approximately 19-20 bits to represent a word by its index into a dictionary in the context. Shannon used Zipf's law to estimate the entropy of an English word at approximately 11.81 bits [83], and this can be reduced with an improved estimate of the distribution of words [38].

In some sense this comparison is unfair to the dictionary approach — compression schemes use fewer bits to represent more frequently occurring words — in fact many compression schemes

ENCODING: Computer code	
x: The ADAPTER program	NONCOPY(x): Those parts of ADAPTER made necessary by compatibility with the operating system
y: The OSCAR program	NONCOPY(y): Those parts of OSCAR made necessary by compatibility with the operating system
CONTEXT: A description of the operating system calls that both programs must adhere to	
NOTES: By the court's determining, all parts of y were either within noncopy(x) or were dissimilar to x [20, l. 714]. Let C be the smallest possible program that takes context and noncopy(x) as input, and outputs y. By assumption, C is very similar to the smallest possible C' that creates y taking context and x as input. We assume they have similar sizes (s and s') and runtimes (t and t').	
BEST P: Runs C' LK($y \mid$ (BG, x)): $s'+\log t'$	BEST R: Runs C LK($y \mid$ BG): $s + \log t$
OUTCOME: $\mathsf{DerSim}(x, y \mid \mathrm{bg}) = s + \log t - (s' + \log t') \approx 0$ COURT OUTCOME: No infringement occurred.	

Table 4: Our framework's treatment of *Computer Associates v. Altai*

store their own dictionaries in this way [80, ch. 5]; in contrast our naive dictionary approach insists upon using the log of the number of entries in the dictionary to index within it. But the point is that the compression will not be improved by a dictionary that is not specific to the work in question. Dictionaries arranged in more advantageous ways compared to y can be considered part of the compression of y itself, and so no longer belong in the context.

5.2 Hash functions cannot be used as a shortcut

We imagine a setting where a plaintiff has copyrighted some work x, but places the hash $h(x)$ of the work in the public domain. This scenario is not theoretical; hashes of potentially-copyrighted files are often used in downloads to ensure that the file being downloaded is the correct one — the hash of the downloaded file is compared to a published hash to ensure that no corruption or tampering occurred during the download.

We imagine that h is a truly random function $h : \{0,1\}^* \to \{0,1\}^n$. (This assumption makes the analysis easier, and we do not expect our results to change meaningfully with a different choice of hash function model.) With overwhelming probability, each image will have infinitely many preimages. A machine that brute-forces messages in lexicographic order until it finds *a* preimage of $h(x)$ is quite small. However, being able to specify *the right* preimage (that is, x), would still require a significant description length, as we proceed to show.

Suppose we consider the binary length of the work ℓ to be part of noncopy(x), and that h hashes strings of that length (i.e., $h : \{0,1\}^\ell \to \{0,1\}^n$). We expect approximately $2^{\ell-n}$ possible preimages to hash to $h(x)$, assuming $\ell \gg n$. Indexing within this set does not save much space compared to needing to write out the entirety of x anyway. The set (and thus indices into it) remain large even when restricted to, e.g., strings with proper English grammar.

This shows that a pure description-length framework does not lose "much" derivation similarity by placing a hash of the copyrighted work in the public domain. Nonetheless, this example motivates us to adopt the Levin-Kolmogorov description length framework we defined formally in §3 rather than a pure Kolmogorov definition, for two reasons.

The first reason is non-technical: it is our intent for the Producer and Reproducer to bear some resemblance to the actual method an author would use when creating the work y from the various inputs. Running the program of "brute force search through the options until locating one that looks like y" does not meet this goal. This motivates the use of Levin-style complexity measures that were originally created in the exact context of brute-force search [57, 58].

The second reason is technical: we do not wish the brute-forcing method to aid *P even a little bit.* Thus, small reduction in size achieved by doing a brute-force search for a certain element is offset by charging for the brute-force search (the log of the machine's runtime). This has virtually no impact on efficient machines, and only becomes important in the setting of a machine as inefficient as brute-force search.

6 ZERO-KNOWLEDGE PROOFS OF DERIVATION SIMILARITY

In this section we consider the following question: what if the plaintiff and defendant want to submit to the court the result of an empirical derivation similarity test, but do not wish to reveal the producer program P and/or reproducer program R? This question is inspired by Bamberger et al. [11], who discuss the dilemma that occurs when "a certain computer program P... is the plaintiff's trade secret" and still the goal is that "the parties can determine similarity of the programs while keeping them secret." Their work pointed out the opportunity to use zero knowledge proofs in cases involving trade secrets. Our goal here is to highlight the value of zero knowledge in copyright cases as well, where some of the same dynamics may apply.

Value to the legal system. Before delving into technical details, we make three remarks about why it might be valuable from a legal perspective to reveal only the result of a derivation similarity test along with a proof of correctness, and why all parties might have incentives to participate.

Our primary notivation is the same as Bamberger et al. [11]: we are concerned about the possibility that the plaintiff and defendant might be less likely to litigate a copyright case if the (re)producer algorithms P and R reveal some "secret sauce" about the creativity in their methods. Our framework does not require P and R to be

publicly visible (only that the background information bg is available to both parties), so pursuing a privacy-preserving test is compatible with the concepts discussed in §3.3.

Second, even though we have focused in this work on litigation in the court system, we envision a private derivation similarity test to be a useful first step in private mediation proceedings. If these proceedings break down and parties move their case to the courtroom, then only a zero knowledge proof needs to be placed in the public record.

Third, even the "losing" party of our derivation similarity test may participate in its computation, both because it may not yet know the final result, and because the party has ample recourse to contest a copyright decision on other grounds even if the similarity metric is not in their favor. Recall that the goal of derivation similarity is to provide a predictable test of where the parties stand with respect to substantial similarity, precisely to allow the parties and courts to focus their litigation on other aspects of copyright law like fair use, DMCA anti-circumvention rules, whether the original work x is copyrightable in the first instance, and so on.

Privacy-preserving derivation similarity. From a computer science standpoint, the ideal tool to use here is *secure multiparty computation* [27], which allows two parties to perform a joint calculation based on hidden data while learning no more than (what can be inferred from) the result. In this case, the plaintiff has a secret program P, the defendant has a secret program R, both parties know (x, y, bg), and they want to calculate the empirical derivation similarity $\mathsf{DerSimEmp}(x, y, P, R \mid \text{bg})$.

Observe that the empirical derivation similarity is the difference of two numbers that plaintiff and defendant individually know. As a consequence, given only the result of the DerSimEmp calculation, each party will learn the Levin-Kolmogorov cost of the other party's program. Put another way: there is no point in trying to hide the $\tilde{C}$ metrics of each party from each other, even though there may be value in hiding this information from the general public.

Consequentially, there is a simple design for a secure computation algorithm for DerSimEmp: each party constructs a *zero knowledge proof* of the complexity of their own program. Concretely, the plaintiff publishes the claimed cost $c_P = \tilde{C}(P, y \mid (\text{bg}, x))$, a (possibly very loose) upper bound T on the runtime of P, and a zero knowledge proof of knowledge of the statement "I know a producer program P, a string z, and a time bound t such that (i) $z \cong y$, (ii) $U(M, (\text{bg}, x), t) = z$ where U_T is a universal Turing machine running in T steps, and (iii) $c_P = |P| + \lceil \log t \rceil$." Similarly, the defendant proves in zero knowledge that they know R that can create something similar to y with c_R Levin-Kolmogorov cost, but without free access to x. Then, the parties work together to produce a recursive zero knowledge proof about the value of $c_R - c_P$ while hiding the two values individually.

Using modern zero knowledge arguments, the resulting proofs, to be placed in the public record can be a few hundred kilobytes in size (e.g., using [40]). Furthermore, using incrementally-verifiable computation [13, 14, 94], the time to produce the proofs is linear in running time of P and R. This proof will hide the parties' programs while also allowing for public verifiability of both the result and the process of executing the derivation similarity test.

ACKNOWLEDGMENTS

We thank Ed Felten and Stacey Dogan for their helpful discussions and valuable insights about computer science and intellectual property law. Sarah Scheffler was supported by a Google Ph.D. Fellowship and the Center for Information Technology Policy at Princeton University. Eran Tromer was supported by the DARPA SIEVE program under Contract No. HR001120C00. Mayank Varia was supported by the National Science Foundation under Grants No. 1718135, 1801564, 1915763, and 1931714, by the DARPA SIEVE program under Agreement No. HR00112020021, and by DARPA and the Naval Information Warfare Center (NIWC) under Contract No. N66001-15-C-4071. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of Google, CITP, DARPA, NSF, NIWC, or any other organization.

REFERENCES

[1] 17 U.S. Code §102 - Subject matter of copyright: In general.
[2] 17 U.S. Code §106 - Exclusive rights in copyrighted works.
[3] 94th United States Congress. Public Law 94-553 – Copyright Act of 1976. http://uscode.house.gov/statutes/pl/94/553.pdf, October 1976.
[4] Harold Abelson, Ross J. Anderson, Steven M. Bellovin, Josh Benaloh, Matt Blaze, Whitfield Diffie, John Gilmore, Matthew Green, Susan Landau, Peter G. Neumann, Ronald L. Rivest, Jeffrey I. Schiller, Bruce Schneier, Michael A. Specter, and Daniel J. Weitzner. Keys under doormats: mandating insecurity by requiring government access to all data and communications. *J. Cybersecur.*, 1(1):69–79, 2015.
[5] Micah Altman, Aloni Cohen, Kobbi Nissim, and Alexandra Wood. What a hybrid legal-technical analysis teaches us about privacy regulation: The case of singling out. *BU J Sci. & Tech. L.*, 27:1, 2021.
[6] Apple Computer, Inc. v. Franklin Computer Corp., 714 F.2d 1240 (3d Cir. 1983).
[7] Arnstein v. Porter, 154 F.2d 464 (2d Cir. 1946).
[8] Baker v. Selden, 101 US 99 (1880).
[9] Shyamkrishna Balganesh. The questionable origins of the copyright infringement analysis. *Stanford Law Review*, 68:791–863, April 2016.
[10] Shyamkrishna Balganesh, Irina D. Manta, and Tess Wilkinson-Ryan. Judging similarity. *Faculty Scholarship at Penn Law*, 1185, 2014.
[11] Kenneth A. Bamberger, Ran Canetti, Shafi Goldwasser, Rebecca Wexler, and Evan J. Zimmerman. Verification dilemmas, law, and the promise of zero-knowledge proofs. *Berkeley Technology Law Journal*, 37, 2022.
[12] Steven M Bellovin, Matt Blaze, Sandy Clark, and Susan Landau. Lawful hacking: Using existing vulnerabilities for wiretapping on the internet. *Nw. J. Tech. & Intell. Prop.*, 12:1, 2014.
[13] Eli Ben-Sasson, Alessandro Chiesa, Eran Tromer, and Madars Virza. Scalable zero knowledge via cycles of elliptic curves. In *Proceedings of the 34th Annual International Cryptology Conference*, CRYPTO '14, pages 276–294, 2014.
[14] Nir Bitansky, Ran Canetti, Alessandro Chiesa, and Eran Tromer. Recursive composition and bootstrapping for SNARKS and proof-carrying data. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *45th ACM STOC*, pages 111–120. ACM Press, June 2013.
[15] James Boyle and Jennifer Jenkins. *Intellectual Property: Law & The Information Society—Cases and Materials.* 5th edition, 2021.
[16] John H Butler. Pragmatism in software copyright: Computer associates v. altai. *Harv. JL & Tech.*, 6:183, 1992.
[17] Aloni Cohen, Moon Duchin, J. N. Matthews, and Bhushan Suwal. Census top-down: The impacts of differential privacy on redistricting. In *FORC*, volume 192 of *LIPIcs*, pages 5:1–5:22. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021.
[18] Aloni Cohen and Kobbi Nissim. Towards formalizing the GDPR's notion of singling out. *Proc. Natl. Acad. Sci. USA*, 117(15):8344–8352, 2020.
[19] Amy B Cohen. Masking copyright decisionmaking: The meaninglessness of substantial similarity. *UC Davis L. Rev.*, 20:719, 1986.
[20] Computer Associates International, Inc. v. Altai, Inc., 982 F.2d 693 (1992).
[21] Concrete Machinery Co. v. Classic Lawn Ornaments, 843 F. 2d 600 (1st Cir. 1988).
[22] Country Kids 'N City Slicks, Inc. v. Sheen, 77 F.3d 1280 (10th Cir. 1996).
[23] Daniel A Crowe. Scope of copyright protection for non-literal design elements of computer software: Computer associates international, inc. v. altai, inc. . *Louis ULJ*, 37:207, 1992.
[24] Davies v. Bowes, 209 F. 53 (S.D.N.Y. 1913).
[25] Eckes v. Card Prices Update, 736 F.2d 859 (2d Cir. 1984).

[26] Walter A Effross. Assaying computer associates v. altai: How will the golden nugget test pan out. *Rutgers Computer & Tech. LJ*, 19:1, 1993.

[27] David Evans, Vladimir Kolesnikov, and Mike Rosulek. A pragmatic introduction to secure multi-party computation. *Foundations and Trends® in Privacy and Security*, 2(2-3):70–246, 2018.

[28] Joan Feigenbaum and Daniel J. Weitzner. On the incommensurability of laws and technical mechanisms: Or, what cryptography can't do. In *Security Protocols Workshop*, volume 11286 of *Lecture Notes in Computer Science*, pages 266–279. Springer, 2018.

[29] Feist Publications, Inc. v. Rural Telephone Service Co., 499 U.S. 340 (1991).

[30] Thomas G. Field. Fundamentals of intellectual property: Cases & materials, 2012.

[31] Financial Inform. v. Moody's Investors Serv, 808 F.2d 204 (2d Cir. 1986).

[32] Robert Fuller Fleming. Substantial similarity: Where plots really thicken. In *Copyright L. Symp.*, volume 19, page 252. HeinOnline, 1969.

[33] Jonathan Frankle, Sunoo Park, Daniel Shaar, Shafi Goldwasser, and Daniel J. Weitzner. Practical accountability of secret processes. In William Enck and Adrienne Porter Felt, editors, *USENIX Security 2018*, pages 657–674. USENIX Association, August 2018.

[34] Sanjam Garg, Shafi Goldwasser, and Prashant Nalini Vasudevan. Formalizing data deletion in the context of the right to be forgotten. In Anne Canteaut and Yuval Ishai, editors, *EUROCRYPT 2020, Part II*, volume 12106 of *LNCS*, pages 373–402. Springer, Heidelberg, May 2020.

[35] Gates Rubber Co. v. Bando Chem. Indus., Ltd., 9 F.3d 823, 835 (10th Cir. 1993).

[36] Shafi Goldwasser and Sunoo Park. Public accountability vs. secret laws: Can they coexist?: A cryptographic proposal. In *WPES@CCS*, pages 99–110. ACM, 2017.

[37] Google LLC v. Oracle America, Inc., 593 U.S. ___, 140 S. Ct. 520 (2021).

[38] Mario C. Grignetti. A note on the entropy of words in printed english. *Information and Control*, 7(3):304–306, 1964.

[39] James Grimmelmann. Patterns of information law: Intellectual property done right, 2017.

[40] Jens Groth. On the size of pairing-based non-interactive arguments. In Marc Fischlin and Jean-Sébastien Coron, editors, *EUROCRYPT 2016, Part II*, volume 9666 of *LNCS*, pages 305–326. Springer, Heidelberg, May 2016.

[41] Christopher Heer. The case against copyright protection of non-literal elements of computer software. *Intellectual Property Journal*, 18:1, 2004.

[42] Herbert Rosenthal Jewelry Corp. v. Kalpakian, 446 F. 2d 738 (9th Cir. 1971).

[43] Judi Boisson and American Country Quilts and Linens, Inc. v. Banian Ltd., and Vijay Rao, 280 F.Supp.2d 10 (2003).

[44] Samuel Judson and Joan Feigenbaum. On heuristic models, assumptions, and parameters. *CoRR*, abs/2201.07413, 2022.

[45] Seny Kamara, Mallory Knodel, Emma Llansó, Greg Nojeim, Lucy Qin, Dhanaraj Thakur, and Caitlin Vogus. Outside looking in: Approaches to content moderation in end-to-end encrypted systems. *CoRR*, abs/2202.04617, 2022.

[46] Seny Kamara, Tarik Moataz, Andrew Park, and Lucy Qin. A decentralized and encrypted national gun registry. In *2021 IEEE Symposium on Security and Privacy*, pages 1520–1537. IEEE Computer Society Press, May 2021.

[47] Jonathan Seil Kim. Filtering copyright infringement analysis in architectural works. *University of Illinois Law Review*, page 281, 2018.

[48] Knitwaves, Inc. v. Lollytogs Ltd.(Inc.), 71 F. 3d 996 (2nd Cir. 1995).

[49] John S. Koh, Jason Nieh, and Steven M. Bellovin. Encrypted cloud photo storage using google photos. In *MobiSys*, pages 136–149. ACM, 2021.

[50] Kohus v. Mariol, 328 F.3d 848 (6th Cir. 2003).

[51] Andrei N Kolmogorov. On tables of random numbers. *Sankhyā: The Indian Journal of Statistics, Series A*, pages 369–376, 1963.

[52] Kregos v. Associated Press, 937 F.2d 700 (2d Cir. 1991).

[53] Anunay Kulshrestha and Jonathan R. Mayer. Identifying harmful media in end-to-end encrypted communication: Efficient private membership computation. In Michael Bailey and Rachel Greenstadt, editors, *USENIX Security 2021*, pages 893–910. USENIX Association, August 2021.

[54] Guillaume Laroche. Striking similarities: Toward a quantitative measure of melodic copyright infringement. *Intégral: The Journal of Applied Musical Thought*, 25:39–88, 2011.

[55] Alan Latman. "Probative similarity" as proof of copying: Toward dispelling some myths in copyright infringement. *Columbia Law Review*, 90, June 1990.

[56] Mark A. Lemley. Our bizarre system for proving copyright infringement. *Journal of the Copyright Society*, 57, 2010.

[57] Leonid A. Levin. Randomness conservation inequalities; information and independence in mathematical theories. *Inf. Control.*, 61(1):15–37, 1984.

[58] Leonid Anatolevich Levin. Universal sequential search problems. *Problemy peredachi informatsii*, 9(3):115–116, 1973.

[59] Ming Li and Paul M. B. Vitányi. *An introduction to Kolmogorov complexity and its applications*. Texts and Monographs in Computer Science. Springer, 1993.

[60] Daryl Lim. Saving substantial similarity. *Fla. L. Rev.*, 73:591, 2021.

[61] Daryl Lim. Saving substantial similarity. *Florida Law Review*, 79, 2021.

[62] Lotus Dev. Corp. v. Borland Int'l, 49 F.3d 807 (1st Cir. 1995).

[63] Matthew Bender & Co., Inc. v. West Pub. Co., 158 F.3d 693 (1998).

[64] Morrissey v. Procter & Gamble Company, 379 F. 2d 675 (1st Cir. 1967).

[65] Nichols v. Universal Pictures Corporation et al., 45 F.2d 119 (1930).

[66] David Nimmer. *Nimmer on copyright*. Matthew Bender Elite Products, 2013.

[67] David Nimmer, Richard L Bernacchi, and Gary N Frischling. A structured approach to analyzing the substantial similarity of computer software in copyright infringement cases. *Ariz. St. LJ*, 20:625, 1988.

[68] Kobbi Nissim. Privacy: From database reconstruction to legal theorems. In *PODS*, pages 33–41. ACM, 2021.

[69] Kobbi Nissim, Aaron Bembenek, Alexandra Wood, Mark Bun, Marco Gaboardi, Urs Gasser, David R O'Brien, Thomas Steinke, and Salil Vadhan. Bridging the gap between computer science and legal approaches to privacy. *Harv. JL & Tech.*, 31:687, 2017.

[70] Robert C Osterberg and Eric C Osterberg. *Substantial Similarity in Copyright Law*. PLI Press, 2003.

[71] Gideon Parchomovsky and Kevin A. Goldman. Fair use harbors. *Faculty Scholarship at Penn Law*, 173, 2007.

[72] Peter Pan Fabrics, Inc. v. Martin Weiner Corp., 274 F.2d 487, 489 (2d Cir. 1960).

[73] R. Ready Productions, Inc. v. Cantrell, 85 F. Supp. 2d 672 (Dist. Court, SD Texas 2000).

[74] Nicole K. Roodhuyzen. Do we even need a test? a reevaluation of assessing substantial similarity in a copyright infringement case. *Journal of Law and Policy*, 15, 2007.

[75] Roth v. United States, 354 U.S. 476, 77 S. Ct. 1304 (1957).

[76] Pamela Samuelson. A fresh look at tests for nonliteral copyright infringement. *Northwestern University Law Review*, 107:1821, 2013.

[77] Pamela Samuelson. Functionality and expression in computer programs: Refining the tests for software copyright infringement. *Berkeley Tech. LJ*, 31:1215, 2016.

[78] Pamela Samuelson. Reconceptualizing copyright's merger doctrine. *J. Copyright Soc'y USA*, 63:417, 2016.

[79] Pamela Samuelson, Randall Davis, Mitchell D Kapor, and Jerome H Reichman. Manifesto concerning the legal protection of computer programs, a. *ColUM. l. reV.*, 94:2308, 1994.

[80] Khalid Sayood. *Introduction to data compression*. Morgan Kaufmann, 2017.

[81] Sarah Scheffler and Mayank Varia. Protecting cryptography against compelled self-incrimination. In Michael Bailey and Rachel Greenstadt, editors, *USENIX Security 2021*, pages 591–608. USENIX Association, August 2021.

[82] Sega Enterprises Ltd. v. Accolade, Inc., 977 F 2d 1510 (9th Cir. 1992).

[83] Claude E Shannon. Prediction and entropy of printed english. *Bell system technical journal*, 30(1):50–64, 1951.

[84] Shaw v. Lindheim, 919 F. 2d 1353 (9th Cir. 1990).

[85] Sheldon v. Metro-Goldwyn Pictures Corp., 81 F. 2d 49, 54 (CA2 1936).

[86] Sid & Marty Krofft TV Prods. v. McDonald's Corp., 562 F.2d 1157 (9th Cir. 1977).

[87] Cathay YN Smith. Truth, lies, and copyright. *Nev. Law Journal*, 20:201, 2019.

[88] Softei v. Dragon Medical & Scientific Communications Inc., 118 F.3d 955 (Fed. Cir. 1997).

[89] Southco, Inc. v. Kanebridge Corp., 390 F.3d 276 (2004).

[90] B. MacPaul Stanfield. Finding the fact of familiarity: Assessing judicial similarity tests in copyright infringement actions. *Drake Law Review*, 49:489–512, 2001.

[91] Transwestern Publ'g Co. v. Multimedia Mktg. Assocs., Inc., 133 F.3d 773 (10th Cir. 1998).

[92] Twentieth Century-Fox Film Corp. v. Stonesifer, 140 F. 2d 579 (9th Circuit 1944).

[93] U.S. Copyright Office. Copyright law of the United States (Title 17) and related laws contained in Title 17 of the United States code. https://www.copyright.gov/title17/, May 2021.

[94] Paul Valiant. Incrementally verifiable computation or proofs of knowledge imply time/space efficiency. In *Proceedings of the 5th Theory of Cryptography Conference*, TCC '08, pages 1–18, 2008.