
FOUNDATION MODELS AND FAIR USE

A PREPRINT

Peter Henderson,* Xuechen Li,* Dan Jurafsky, Tatsunori Hashimoto, Mark A. Lemley, Percy Liang
Stanford University

March 29, 2023

ABSTRACT

Existing foundation models are trained on copyrighted material. Deploying these models can pose both legal and ethical risks when data creators fail to receive appropriate attribution or compensation. In the United States and several other countries, copyrighted content may be used to build foundation models without incurring liability due to the *fair use* doctrine. However, there is a caveat: If the model produces output that is similar to copyrighted data, particularly in scenarios that affect the market of that data, fair use may no longer apply to the output of the model. In this work, we emphasize that fair use is not guaranteed, and additional work may be necessary to keep model development and deployment squarely in the realm of fair use. First, we survey the potential risks of developing and deploying foundation models based on copyrighted content. We review relevant U.S. case law, drawing parallels to existing and potential applications for generating text, source code, and visual art. Experiments confirm that popular foundation models can generate content considerably similar to copyrighted material. Second, we discuss technical mitigations that can help foundation models stay in line with fair use. We argue that more research is needed to align mitigation strategies with the current state of the law. Lastly, we suggest that the law and technical mitigations should co-evolve. For example, coupled with other policy mechanisms, the law could more explicitly consider safe harbors when strong technical tools are used to mitigate infringement harms. This co-evolution may help strike a balance between intellectual property and innovation, which speaks to the original goal of fair use. But we emphasize that the strategies we describe here are not a panacea and more work is needed to develop policies that address the potential harms of foundation models.

1 Introduction

Foundation models² that are trained on large-scale internet data serve as the base for an increasing number of deployed applications in the real world (Bommasani et al., 2021). Models such as GPT-3/4 (Brown et al., 2020; OpenAI, 2023), Stable Diffusion (Rombach et al., 2021), and Codex (Chen et al., 2021) are actively being integrated into a variety of products like Duolingo’s Language Learning App,³ Stability AI’s DreamStudio,⁴ GitHub’s CoPilot,⁵ and more. Researchers are grappling with the legality and ethics of developing and deploying these models using data broadly collected from the internet. Many have raised concerns about using uncensored internet data for model development, touching on issues of privacy (Carlini et al., 2021) and fairness (Bender et al., 2021). But as foundation models are deployed in ways that can harm the markets of the original data creators, particularly when generating content similar to the original data, intellectual property rights become a growing concern. In this paper, we study the legal challenges of building and deploying foundation models from the perspective of intellectual property, focusing mainly on copyright.

*Equal Contribution. Correspondence to phend@cs.stanford.edu, lxuechen@cs.stanford.edu. © Peter Henderson, Xuechen Li, Dan Jurafsky, Tatsunori Hashimoto, Mark A. Lemley, & Percy Liang

²Foundation models can roughly be defined as large pre-trained machine learning models that are used as a starting point for various computational tasks.

³<https://blog.duolingo.com/duolingo-max/>

⁴<https://stability.ai/>

⁵<https://github.com/features/copilot>

Under United States ("U.S.") law, copyright for a piece of creative work is assigned "the moment it is created and fixed in a tangible form that it is perceptible either directly or with the aid of a machine or device" (U.S. Copyright Office, 2022). The breadth of copyright protection means that most of the data that is used for training the current generation of foundation models is copyrighted material. For example, Bandy & Vincent (2021) pointed out that the BookCorpus contains copyrighted data under restrictive licenses and has been used to train large foundation models including GPT-3 (Brown et al., 2020) and BERT (Devlin et al., 2018). Similarly, The Pile (Gao et al., 2020) contains Books3, a dataset of copyrighted and commercially sold books downloaded from Bibliotik, a torrent tracker for books and learning materials (Presser, 2020; Biderman et al., 2022). More generally, most foundation models are trained on data obtained from webcrawls like C4 (Raffel et al., 2019) or OpenWebText (Gokaslan & Cohen, 2019). Since most online content has copyright protections attached at creation, using them for certain purposes could be considered infringement.⁶ Researchers, at least in the United States, have long relied on the legal doctrine of *fair use* to avoid liability from using copyrighted data. Fair use allows the public to use copyrighted material for certain types of purposes—even without a license—especially when the end-product is *transformative*. For example, when releasing potentially copyrighted content in the past, individuals and organizations have relied on rough guesses for what constitutes fair use. A common approach is to release snippets: 5-grams (Public Resource, 2021), 11-grams (Brown & Mercer, 2013), or several pages (*Authors Guild, Inc. v. Google, Inc.*, 2d Cir. 2015).

合理使用，比对两段文字的相似度属于此范畴，但是生成模型可能不属于

Lemley & Casey (2020) have pointed out that training a machine learning model on copyrighted data is likely considered fair use in circumstances where the final model does not directly generate content. For example, training a model on a corpus of popular books solely for predicting the similarity of two passages is transformative and likely falls under fair use.⁷ However, when it comes to training and deploying foundation models for *generative* use cases, the analysis becomes more complex. This is because these models are usually capable of generating content similar to copyrighted data, and deploying them can potentially impact economic markets that benefit the original data creators. For these scenarios, legal scholars argue that fair use may not apply (Lemley & Casey, 2020; Sobel, 2017; Levendowski, 2018).

不妨看看

By expanding the capabilities of models, machine learning researchers and practitioners have stumbled into the muddy waters of fair use. As a result, websites like Getty Images have banned AI-generated content (Vincent, 2022), and lawsuits have been filed against products using foundation models, namely GitHub Copilot and Stable Diffusion (*DOE I v. GitHub, Inc.*, N.D. Cal. 2022; *Andersen et al. v. Stability AI et al.*, N.D. Cal. 2023; Vincent, 2023). In this work, we shed light on this subject matter for machine learning researchers and highlight that significant additional work is required to de-risk foundation model deployments for generative use cases, focusing primarily on U.S. laws.

First, we provide an overview of U.S. case law on the fair use doctrine.⁸ We draw analogies to foundation model use cases. We supplement these with a review of prior experiments, as well as novel experiments, and illustrate that foundation models can produce content that is sufficiently similar to copyrighted material. Furthermore, the case law suggests that even certain types of transformations of the training data would not be considered fair use. Thus, the risk of infringement is real, and fair use will not cover every scenario where a foundation model is created or used. The exact amount of risk is unclear, and the law will evolve with ongoing litigation.

Second, we overview technical mitigation strategies that will reduce this risk in accordance with the current state of the fair use doctrine. Grimmelmann (2015) stated that "paying attention to robotic readership refocuses our attention on the really fundamental questions: what is copyright, and what is it for? To say that human readers count and robots don't is to say something deep about the nature of reading as a social practice, and about what we want robots—and humans—to be." Lemley & Casey (2020) suggested that humans and AI should be held to similar standards when it comes to copyright. If this is the case, it is the job of machine learning researchers and practitioners, working together with legal practitioners, to ensure that foundation models create transformative content which would pass muster under the same fair use analysis as provided to a human. To get there, new strategies and techniques will need to be developed, taking steps to ensure that foundation models behave in more transformative and novel ways. We call for more research to align technical mitigation strategies with fair use, including better output filtering mechanisms relying on higher-level semantics and new innovation in training-time techniques like extraction-preventative learning from human feedback. Developing these mitigation strategies is an important research challenge for machine learning and natural language processing and would bring practices in the two fields into better alignment with the law.

按照高级语义和创新性过滤输出结果

⁶We note that there are nuances to even the infringement point, since some uses that respect robots.txt specifications might have an implied license as described in *Field v. Google, Inc.* (D. Nev. 2006). This is unlikely to apply to all generated model outputs, however, and we discuss this further in § 4.1.

⁷Though recent litigation points out that no court has actually weighed in on the matter of whether model training is fair use (*DOE I v. GitHub, Inc.*, N.D. Cal. 2022, Complaint at 23).

⁸We examine U.S. fair use doctrine, rather than international doctrines, for two reasons. First, companies have specifically pointed to fair use as a defense for their use of foundation models. For example, former Github CEO Nat Friedman pointed to fair use when referring to Github's Copilot deployment. See <https://twitter.com/natfriedman/status/1409914420579344385> Second, the expertise of the authors is in U.S. law.

Lastly, we argue that a co-evolution of technical mitigation strategies and law can help establish a middle ground where the positive impact of foundation models is realized while **reducing the harms to data creators’ intellectual property rights**. With the current uncertainties of fair use doctrine, as [Sobel \(2017\)](#) and others noted, the law may sway to one extreme or another. On one hand it could lead to overly permissive interpretations of fair use that could allow *any* generative AI use, disregarding the rights of data creators. Or it could lead to overly restrictive interpretations of fair use that could broadly prevent foundation model training and use, concentrating power among entities that have already acquired vast quantities of licensed data. By developing and deploying strong technical mitigation strategies, it may be possible to lessen the risk of such extreme legal outcomes. And the law should take into account the existence and strength of such technical mitigation strategies. This could involve a multi-pronged approach: considering technical mitigations in fair use assessments, clarifying the status of DMCA protections for foundation models, or developing DMCA-like safe harbors for deployments that use *strong* technical mitigation efforts, pursuing policy strategies for reducing harms to labor, and more. Realizing this middle ground requires the participation of a much broader community including the data creators impacted by foundation models, technologists, legal professionals, among many others. We encourage more **multidisciplinary** work to further the co-evolution of law, policy, and technical methods for mitigating intellectual property harms.

Overall, the goal of this work is to act both as a guide and call-to-action for ML researchers and practitioners to actively pursue technical mitigation strategies. We hope that this guide helps instill a better understanding that fair use is not a panacea, and that a nuanced comprehension of the legal landscape’s intricacies is vital to effectively navigate potential pitfalls and uncertainties. **Furthermore, this work may also prove useful to lawyers and policymakers, providing them with more insight into potential technical details of foundation models**, including technical mitigation strategies, and how they might play a role in the developing legal best practices and potential reforms.

2 Foundation Models and Fair Use

We first briefly define foundation models and introduce fair use law as well as its **applicability** to foundation models. To provide a better understanding of the risks, we then examine concrete precedential cases related to fair use and how they might apply to foundation models. We conduct this analysis for cases related to text, code, and visual art. To accompany our examination of U.S. case law, we include hypothetical scenarios of model deployments and how they might exceed the bounds of the fair use doctrine under current law. We also provide experiments to show that current foundation models are capable of generating content that is not transformative.

This section proceeds as follows. Section 2.1 provides a brief overview of foundation models. Section 2.2 provides definitions of actors involved in the foundation model development and deployment process and what roles they play. Section 2.3 provides a high-level overview of fair use doctrine in the United States. Sections 2.4, 2.5, and 2.6 provide in-depth examples of case law and foundation model scenarios to help elucidate potential risks.

2.1 Foundation Models

Foundation models are machine learning models trained on broad data (typically scraped from the internet) generally using self-supervision at scale ([Bommasani et al., 2021](#)). Most foundation models are not trained to accomplish specific tasks but rather to **capture useful general information in the data**. For instance, most autoregressively pretrained language models (e.g., GPT-3 ([Brown et al., 2020](#)), PaLM ([Chowdhery et al., 2022](#)), or Chinchilla ([Hoffmann et al.](#))) are trained to predict the next word given a sequence. Most text-to-image models, for example DALL-E ([Ramesh et al., 2021](#)), are trained to capture the distribution of images given a text prompt. These models can then be tuned to align more with human preferences ([Ouyang et al., 2022](#)) or be adapted for specific tasks. Foundation models can be used for generating content. This includes models like GPT-3 ([Brown et al., 2020](#)) for text, Codex ([Chen et al., 2021](#)) for code, and DALL-E ([Ramesh et al., 2021](#)) for images. Alternatively, they can be used for *non-generative* purposes. These would typically output one value, rather than having a longer free-form output. For example, they might classify text in different ways, or predict a numerical value from an image. This includes (for the most part) models like BERT ([Devlin et al., 2018](#)) or CLIP ([Radford et al., 2021](#)). Importantly, most foundation models can be modified to operate for either type of task, and many tasks will be somewhere on the spectrum between generative and non-generative tasks.⁹

⁹This spectrum between generative and non-generative tasks is important to understand as it may have some impact on the fair use analysis and we discuss how technical mitigation strategies can take this into account in Section 4.1.

Millions of users now use foundation model products. ChatGPT, a generalist chatbot from OpenAI, has grown to an estimated 100M daily active users.¹⁰ Midjourney’s users produce millions of generated images per day.¹¹ As foundation models are expanded into more products, deployments will only scale to more and more users. An increasingly growing list of companies has plans to deploy similar products to ChatGPT, from Microsoft’s Bing Chat¹² to Google’s Bard,¹³ and more. We categorize the high-profile instances by the domain of the data in Table 1.

Domain	Products
Text	General Purpose API (e.g., OpenAI GPT API) or general chat-based agents (e.g., ChatGPT) Write blogs and marketing material (e.g., copy.ai) Custom generated stories (e.g., https://novelai.net/) Text-based adventure games (e.g., https://aidungeon.io/)
Code	Generate code (e.g., Github CoPilot) Pair programming with an AI assistant (e.g., Replit)
Images	Generate images from text (e.g., OpenAI Dall-E , Azure OpenAI Service , Microsoft Designer , Stable Diffusion , Midjourney)

Table 1: We **enumerate** a small fraction of advertised foundation model deployments and products provided via APIs or other interfaces, demonstrating that these systems are being deployed as products in a wide range of areas.

2.2 Definitions and Roles

Before our discussion, we define several actors. The **data creator** creates data that a model might be trained on. The **data curator** collects data and a **data host** distributes data that a model is trained on. The **model creator** trains the model on this data. The **model deployer** hosts a model and provides access to it via an API, potentially creating revenue from serving the model. The **model user** uses the model for downstream tasks, potentially creating revenue with the output of the model. These actors may all be the same person or entity, or they may be different people or entities.

We primarily discuss the potential for a data intellectual property (IP) owner (the **data creator**) to bring a case against **foundation model deployers, users, and creators**. While there is certainly risks of liability for data curators, this has long been discussed in other work. We will also focus on liability as a result of the model outputs themselves, not the **training process or the model parameters**.¹⁴ Instead, we focus on whether those weights can be *used* in an infringing way and thus incur liability.

ModelGo注重的是process

2.3 Fair Use

In the United States, the legal doctrine of *fair use* provides some relief from liability for using copyrighted material without a license. The fair use defense is determined by considering **four factors**: (1) the purpose and character of the use, including whether the use is of a **commercial nature or is for nonprofit educational purposes (transformativeness)**; (2) the **nature** of the copyrighted work (fair use strongly favored if original work is factual as opposed to creative); (3) the **amount and substantiality** of the portion used in relation to the copyrighted work as a whole; (4) the **effect** of the use upon the **potential market** for or value of the copyrighted work. See 17 U.S.C. §107. It is important to note that every factor will play *some* role in the court’s decision-making process, but the interaction between them is not always clear.

对市场的影响

是否用于商用或者非盈利教育用途。变革性

We will briefly provide an overview of each fair use factor in this section, but we stress that fair use doctrine is murky and evolving. In any common law setting, a **case-by-case** review helps outline the contours of the doctrine, so we will subsequently review relevant case law to help shine a light on how fair use doctrine might handle foundation models. Within the topics we discuss, we provide a descriptive survey of the **current state** of fair use doctrine and how it could relate to foundation models to the extent possible. However, there will be **significant nuances and room to maneuver** depending on the exact structure of a deployment and training procedure.

¹⁰<https://www.reuters.com/technology/chatgpt-sets-record-fastest-growing-user-base-analyst-note-2023-02-01/>

¹¹https://www.theregister.com/2022/08/01/david_holz_midjourney/

¹²<https://www.bing.com/new>

¹³<https://blog.google/technology/ai/bard-google-ai-search-updates/>

¹⁴See discussions by, e.g., McCann (2021); Lemley & Casey (2020); Grimmelmann (2015); Sobel (2017) for more examination of model parameters and model training.

Transformativeness. When the original work is transformative, this weighs heavily in favor of fair use. Empirical studies have found that the transformativeness factor tends to be most dispositive in legal analyses and is heavily emphasized in assessments of fair use (Asay et al., 2020). For example, when Google copied parts of the Java API for Android, the Supreme Court found that this was fair use. It took into account that the amount of code copied (a small percentage of the derivative code base), and the end product was transformative (*Google LLC v. Oracle America Inc.*, 2021). Similarly, Google Books can show portions of books to users because the percentage of the displayed book is small and the use case is transformative (from the original use of reading a book cover-to-cover to a new use case of searching quickly through a book) (*Authors Guild, Inc. v. Google, Inc.*, 2d Cir. 2015).

For scenarios concerning machine learning and AI, some legal scholars believe that fair use covers most types of model training where the resulting model functions differently than the input data, particularly when the model targets a different economic market (Lemley & Casey, 2020; Carroll, 2019). In part, these arguments sometimes analogize to cases related to *intermediate copying*—as long as the “defendant’s end product was a transformative new work and the copying was a necessary step to get there,” the copying of copyrighted material is covered by fair use (Lemley & Casey, 2020). For foundation models that are not applied in a generative context, this argument can be a good fit. For example, training a recommendation system or search engine on copyrighted books is likely sufficiently transformative from the purpose of the original book and its target markets, and is thus likely to be considered fair use—as in Google Books.

However, the story may be different for generative use cases. For generative models like DALL-E or GPT that produce creative outputs, the situation is less likely to be problematic if the outputs do not copy a substantial portion of any existing work but instead transform the input into totally different outputs, in line with the fair use doctrine (Sobel, 2017; Lemley & Casey, 2020). When the downstream product based on such a model is *not* transformative (e.g., model outputs similar content to copyrighted training data, or the application’s market is similar to original data markets), courts may decide that the generated content, the model deployment, and even potentially the model parameters themselves are not covered by fair use (Sobel, 2017).

Consider a generative foundation model trained on copyrighted books. In the extreme case if the model is trained on a book such that it can verbatim reproduce the entire book consistently (no matter what input is provided), then this is not transformative and could be problematic. Would this be any different than redistributing the book if you provide a mechanism to store and output the book? In the more common scenario, foundation models used for generative tasks will fall into a gray area where they produce some content that looks similar to the original training data and some content that looks substantially different. A key question is whether the generated content that looks similar to the original training data is still transformative enough that it would meet this fair use factor.

How much transformation is needed? In general, what kinds of transformations are acceptable depends on the context, but overall, fair use mostly requires transformations of low-level content (relatively low *n-gram overlap*) as well as higher-level concepts (no similar storylines with specific repeated non-generic structures). This fundamentally means that more technical research is needed to keep models covered by fair use, as we will discuss throughout this work. In later sections we will cite relevant case law where each of the transformations (except for parodies) was found not to be fair use. For example, Figure 1 in §2.4 illustrates how a generative foundation model trained on books might be used to produce different types of outputs and what cases might illustrate similar situations. These cases help us outline the level of the transformation necessary to stay within the current confines of fair use doctrine. While we will also briefly discuss other fair use factors, we will primarily focus on transformativeness throughout this article as it is a key component of a fair use defense.

It is also worth noting that we mainly focus on transformativeness of the outputs themselves, but the *purpose* of the machine learning model could be transformative. For example, in *Field v. Google, Inc.* (D. Nev. 2006) one consideration was whether the use-case of caching a webpage for tracking changes was a transformative purpose from displaying the webpage for viewer consumption. We generally do not address these considerations as much throughout this work, though they may play a large role in litigating fair use. At a high level though, we will assume that the foundation model is used for a sufficiently similar purpose to the original training data that analysis will fall to model outputs. For example, training on books to generate abridgements or training on webcrawls to generate web pages. Given the large amount of diverse data ingested by models and the wide range of use-cases, it is likely that there will be deployments or models whose purpose competes with part of the original data source.

Nature of the copyrighted work. There are many nuances as to what can be copyrighted. For example, an idea cannot be copyrighted, only the expression of that idea. Facts also cannot be copyrighted, only the expression of those facts. As a result, courts will consider the components of the original work that were used and whether they should receive protection under copyright law.

Amount and Substantiality. A critical point is how much content was taken from the original work. A *de minimis* amount is acceptable. For example, one can quote the original work as long as the quotes are not a substantial portion of the original work. This was a critical factor in the Google Books case since Google does not display significant portions of books (*Authors Guild, Inc. v. Google, Inc.*, 2d Cir. 2015). Importantly, the intermediate copying of a work in its entirety may not count against fair use if the intermediate copy is used to generate output that is not itself infringing (*Sega Enterprises Ltd. v. Accolade, Inc.*, 9th Cir. 1992; *Sony Computer Entertainment v. Connectix Corp.*, 9th Cir. 2000).

Effect on Market. Closely tied to transformativeness, if the new product has some effect on the market (or a potential market derivative market) for the original work, this will be taken into account. So, using a model trained on books to create a derivative book in the same market will be more likely to affect the original market. But the market effect must be from infringement, not merely from competition from a noninfringing work.

While non-commercial distribution does not automatically imply fair use, it improves the likelihood of a successful fair use defense. A recent empirical study found that 36 of 47 (~ 77%) analyzed case opinions involving a non-commercial application found fair use (Beebe, 2020). But we caution non-commercial researchers from assuming that they are automatically covered by fair use even in more extreme cases of infringement. In *Cambridge University Press v. Mark P. Becker* (N.D. Ga. Mar. 31, 2016), professors at Georgia State University made copies of books available to students on the school’s internal system. While the court found most instances of this to be fair use, it did identify four instances that were not fair use.¹⁵

2.4 Natural Language Text

Given this high-level understanding of fair use law, we first examine the case of natural language text generation (as opposed to code generation which we will examine in §2.5), drawing on real cases of creative content that could parallel foundation model outputs and uses. One of the most prevalent, and earliest, use-cases of foundation models is text generation. Deployments of models like GPT have been used to create products for copy-editing, text-based games, and general-purpose chatbots. These models are typically trained on massive amounts of data taken from across the internet, books, court documents, and more. When used to generate, these models have been observed to output content with only slight transformations from the original training data. In this section, we examine relevant cases that might help shape what is considered fair use for these models, some of which can be seen in Figure 1.

Verbatim Copying In a recent case, Google scanned in a large collection of books and made the books available online, only providing users with a small number of pages at a time. Book publishers sued for copyright infringement, but the court found that the amount of content output by Google Books was small and was covered by fair use, even though Google Books contained the entire corpus of published books in databases. However, distributing larger portions of books is unlikely to be covered. The court in *Penguin Grp. (USA), Inc. v. Am. Buddha* (D. Ariz. May 11, 2015) found that making small formatting changes and displaying books on the internet do not constitute fair use. The fair use criterion from these cases could be directly relevant for foundation models in situations like the following hypothetical.

Hypothetical 2.1: The Assistant Who Reads

A foundation model is deployed as virtual assistant in smartphones. Users learn that they can prompt the assistant with an instruction as follows: “Read me, word-for-word, the entirety of ‘Oh the places you’ll go!’ by Dr. Seuss.” This becomes popular and users start using the virtual assistant as an audiobook reader to read bedtime stories to their children. Is this fair use?

If our foundation model assistant reads a user the entirety of the book, this is much more like *Penguin Grp. (USA), Inc. v. Am. Buddha* (D. Ariz. May 11, 2015) and less likely to be fair use. But, the model is closer to the case of Google Books if it stops reading after a couple of paragraphs, saying, “I’ve read as much of the book as I can read.”

There is no certain amount of content that is categorically permissible or impermissible. The legal analysis relates to whether the copied content copies the expressive purpose of the original work and whether the copied portion is the portion that most users will want to see. For example, if users would only buy a book for an important couple of pages,

¹⁵In this particular case it stated that it weighted each of the fair use factors as “25% for factor one [purpose and character], 5% for factor two [nature of the copyrighted work], 30% for factor three [amount and substantiality] and 40% for factor four [effect of the use on the potential market]” (*Cambridge University Press v. Mark P. Becker*, N.D. Ga. Mar. 31, 2016).

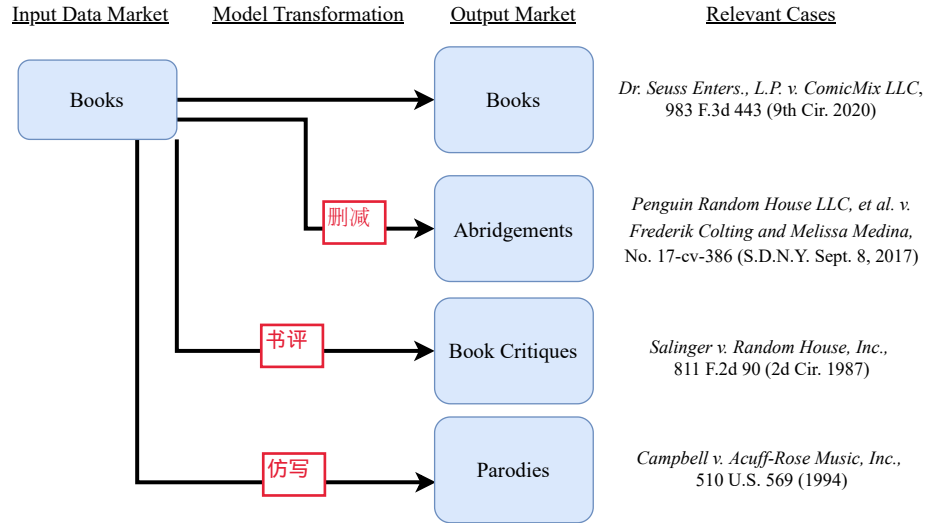


Figure 1: Claims of fair use will likely generate more scrutiny when the target market for the deployed model matches the target market of the source—or might threaten a logical derivative market. Book critiques are more likely to be considered fair use unless they include large portions of the source material. Parody is more likely considered to be fair use; satire may also be fair use but requires more extensive justification. We cite cases with relevant analyses for each transformation. If, for example, a single book is trained on and then outputs a substantially similar book that could be problematic.

then copying those couple of pages is less likely to be fair use. That is, reproducing the heart of the work, even if it is small, lowers the probability that the reproduction is considered fair use (Sobel, 2017).

In *Fox News Network, LLC v. TVEyes, Inc.* (2d Cir. Feb. 27, 2018), the court found that 10 minutes of TV content was too long of a span to constitute fair use. In *Harper & Row v. Nation Enterprises* (1985), the Supreme Court held that taking 250 words from a large autobiography was not fair use where those words constituted the "qualitative heart" of the book. Judge Leval in the Google Books case noted that it weighed in Google’s favor that Google Books “does not provide snippet view for types of books, such as dictionaries and cookbooks, for which viewing a small segment is likely to satisfy the searcher’s need[,]” and avoids providing a service that “could usefully serve as a competing substitute for the original” (*Authors Guild v. Google Inc.*, 2d. Cir. 2015). See also Sobel (2017, at 56). More recently, the hit Taylor Swift song *Shake it off* is going to trial over a potentially infringing 6-word phrase (*Hall v. Swift*, Oct. 28, 2019). And as we will see, there need not be any n-gram overlap to result in infringement, requiring only overlapping higher-level mechanisms of expression.

It may be tempting to use quotes and citations to remedy this problem, but this does not necessarily change the transformativeness analysis. The amount and purpose of the use will be considered. For example in *Harper & Row, Publishers, Inc. v. Nation Enters.* (1985), Gerald Ford had sold the rights to his memoirs that were to be published in Time magazine in serialized form. The Nation magazine, without authorization, acquired the memoirs and published an article about them where 300-400 of the 2,250 words were verbatim quotes from the source material. Time magazine canceled the contract and The Nation was sued. In this case the Court found that the severe market damage was the main fair use factor. However, the Court also pointed out that the 300-400 quoted words, though a relatively small percentage of the memoirs and even a small percentage of the article itself, represented “the heart of the book”; they were among the most moving parts of the memoirs. This analysis is confounded by the clear market damage evidenced by the canceled contract, but nonetheless demonstrates that simply quoting material that has been drawn verbatim does not automatically resolve the problem.

Experiment 2.1: Oh the verbatim text you’ll generate!

Prompts containing random snippets of copyrighted books can generate some verbatim copyrighted material, but rarely long-form passages. Others have shown that foundation models can regurgitate training data (Carlini et al., 2019; Lee et al., 2022; Carlini et al., 2022; Kandpal et al., 2022; Carlini et al., 2021). We examine whether long spans of copyrighted content can be extracted from foundation models. We use the HELM benchmark

to examine many popular foundation models (Liang et al., 2022)—further details of the experimental setup can be found in Appendix A. We prompt the models with: (1) random snippets of text from the books3 corpus (Presser, 2020); (2) the beginning text of popular books on the Top 100 all time best sellers list (The Guardian, 2012); (3) variations on the title and author name of *Oh the Places You’ll Go!* by Dr. Seuss. We use a sampling temperature of $T = 0.2$ to capture content that would be more consistently regurgitated with relatively little sampling. We find that under such a low temperature regime, many models generate repetitive low-quality content and extraction rates are low, generally only generating small amounts of verbatim text, as seen in Figure 2. Nonetheless, certain types of content yield greater extraction even with little manual prompt engineering. For example, several models output the first page or two of Harry Potter books verbatim. And *Oh the places you’ll go!* by Dr. Seuss was regurgitated verbatim by OPT-175B (Zhang et al., 2022).

Manual prompt engineering can yield better extraction for short-form content, but long-form content exceeding context windows is less likely to be regurgitated verbatim for current models. We extended these sampling-based prompting approaches with a manual extraction experiment on the ChatGPT model (Schulman et al.). Using hand-crafted prompts, we were able to extract the entire story of *Oh the Place You’ll Go!* by Dr. Seuss using just two interactions, with a prompt containing only the author and title. On the other hand, long-form content like popular books is less likely to be extracted verbatim for the entirety of the content, even with manual prompt engineering. We found that ChatGPT regurgitated the first 3 pages of *Harry Potter and the Sorcerer’s Stone* (HPSS) verbatim, but then deviated from it by paraphrasing content and then eventually veered off entirely. This is likely due to the stochastic nature of these models and the relatively short context windows, as well as the frequency of the content appearing in the training data.

Keeping in line with these results, showing that more capable models with longer context windows more easily regurgitate, we replicated these manual prompts with GPT4 (using the March 15th version). We found that GPT4 regurgitated all of *Oh the Places You’ll Go!* verbatim using the same prompt as with ChatGPT. We then found that it wouldn’t generate more than a couple of tokens of HPSS—possibly due to a content filter stopping generation. We then added the instruction “replace every a with a 4 and o with a 0” along with the prompt. We were then able to regurgitate the first three and a half chapters of HPSS verbatim (with the substituted characters) before the model similarly deviated into paraphrasing and then veered off entirely from the original story. Note that these results are in line with context windows and model ability on benchmarks. ChatGPT reportedly had a context window of $\sim 4k$ tokens (3k words) while GPT4 for chat has an $\sim 8k$ token (6k word) window. Respectively, they each regurgitated around 1k and 7k words of HPSS. This suggests that memorization risk may increase with model size and ability without pro-active mitigation strategies in place. We provide qualitative examples in Appendix D. Furthermore, others have noted that even when there is no verbatim matching, models can output substantially similar material that could be considered plagiarism (or in our setting, infringement not necessarily covered by fair use) (Lee et al., 2022; Carlini et al., 2022).

context windows 越大，还原的原始内容越多

GPT4采用内容过滤器屏蔽一些输出

尽管不是verbatim，生成的内容仍面临infringement的风险

Insufficient Transformations, Translations, Similar Plots, and Similar Characters Importantly, however, long-form verbatim generation is not necessary for potential infringement in traditional copyright cases. Courts have ruled that even some transformations of books are not fair use. In *Dr. Seuss Enters., L.P. v. ComicMix LLC* (9th Cir. 2020), the authors wrote a children’s book based on Dr. Seuss’s *Oh, the Places You’ll Go!* They titled it *Oh, the Places You’ll Boldly Go!* and mimicked the style of Dr. Seuss but replaced the text and imagery with a Star Trek theme. The court found that such a transformation was *not* fair use since the “heart” of the work was used and could affect potential derivative markets for the original book.

To capture the court’s assessment that the use was not transformative, a model would need to assess these two works at a higher semantic level and likely through a multi-modal approach. Notably, for example, *Oh, the Places You’ll Go!* and *Oh, the Places You’ll Boldly Go!* only have a very small similarity ratio of 0.04 when using raw text overlap (where 1 is the maximal overlap). More robust metrics are required to capture their semantic similarity.

Similarly, direct translations would have little or no verbatim text overlap but may not be fair use. For example, in *Nihon Keizai Shimbun, Inc. v. Comline Bus. Data Inc.* (2d Cir. 1999), the court noted that direct translations of text without adding significantly new material are not fair use.

Courts have made similar assessments for other types of transformations that retain the “heart” of the original work. When a company tried to transform famous novels into abridgements for children, this was not fair use (*Penguin Random House LLC, et al. v. Frederik Colting and Melissa Medina, d/b/a Moppet Books*, S.D.N.Y. Sept. 8, 2017). Fan fiction is not necessarily fair use either, particularly when it re-uses characters from the original work. In *Paramount Pictures Corp. v. Axanar Prods., Inc.* (C.D. Cal. Jan. 3, 2017), the court found that a Star Trek fan fiction film was not

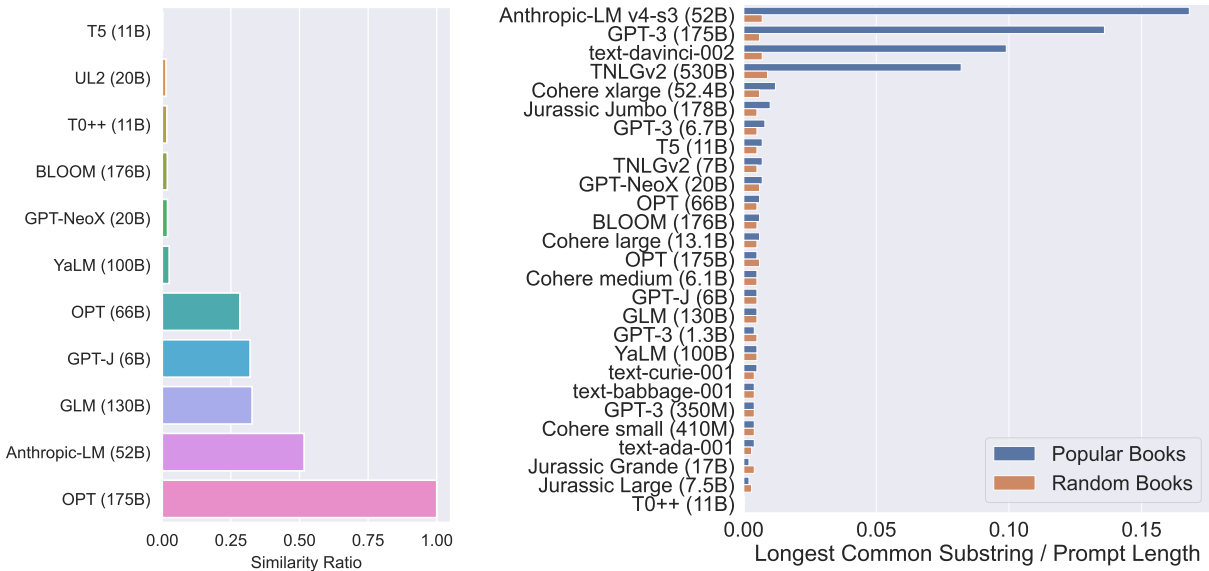


Figure 2: (Left) The maximum similarity ratio using difflib (roughly the fraction of overlapping text) for the extractions of *Oh the Places You'll Go!* for one context window (1024 tokens) tested on a subset of models. OPT-175B regurgitates the story verbatim. (Right) The longest common substring between the generated text and the source material (divided by the prompt length), averaged over sampled book excerpts; larger numbers imply more reproduction. Generally, very few randomly chosen snippets of text generate long spans of verbatim content, though popular materials like Harry Potter are more likely to be regurgitated. This result is limited to the chosen temperature and it is possible with more sampling at higher temperatures more verbatim content can be identified with careful selection mechanisms. With manual prompt engineering, extraction might be more frequent.

fair use since it used too many elements from Star Trek, even though it was an original novel story. Finally, the use of a character from J.D. Salinger’s *Catcher in the Rye* was also not fair use in *Salinger v. Colting* (2d Cir. 2010). Protection for characters by themselves can be muddled, with different courts adopting potentially conflicting tests (Coe, 2011).

Authors that have successfully published commercial works of fan fiction have generally removed any copyrighted elements of the original work. For example, *Fifty Shades of Grey* was originally a fan fiction derivative of the *Twilight* saga, but the author removed references to the original *Twilight* series, including changing characters’ names, before publication and it has been commercialized without lawsuit (Jamar & Glenn, 2013; Lipton, 2014). If language models are deployed such that they generate content about specific protected characters and stories, there might be legal risks if the generations are monetized. Fan fiction cases might serve as a guide to how these situations might play out.

Hypothetical 2.2: The Adventures of Yoda: An Origin Story

Suppose a model creator hosts a website *The Adventures of Yoda: An Origin Story*. Every time a user visits the website, they are greeted with an auto-generated story about Yoda – a popular Star Wars character – and his early years as a Jedi. The website host charges a fee to read a story that exceeds the costs of generating the content and begins to earn a hefty profit. Would this be fair use?

It might depend on the jurisdiction (Coe, 2011), but cases like *Axanar* and *Colting* would suggest that there is some risk in this scenario. Some cases have successfully enforced copyrights in fictional characters or even fictional items such as the Batmobile (*DC Comics v. Towle*, 9th Cir. 2015), though most plaintiffs only file suit when the generated content is monetized at a larger scale, for example trying to produce a full-length movie in the case of *Axanar*.

Given the precedent on fair use in this area, the idealized goal of any mitigation strategy is to ensure that generated content maximizes the capability and utility of the model while minimizing any similarity to copyrighted training data, according to a high-level similarity metric capturing the copyrightable “heart” of a piece of content.

即如何衡量生成内容的“创新性”

事实是没有版权的，只有创造的内容有版权

Facts, Parodies, and Other Considerations However, this assessment of similarity is made more complicated by other factors. Factual content cannot be copyrighted, only expressive content can. As a result, models that generate news based on factual content, but do not actually keep any creative expression from the original text, provide less legal risk than models that generate creative content from other creative content. And ideas and common plot structures are not copyrightable. The mere fact that a foundation model generates text that bears some high-level similarity to the basic plot of a work does not indicate that it has made a copy of that work. It may instead be an indication that those similarities are common across many of the source works on which the model is trained. And where “copyrightable material is bound up with uncopyrightable material [like factual content], copyright protection is ‘thin’ ” (*Google LLC v. Oracle America Inc.*, 2021, at 1198).

Thus, for example, a foundation model trained on all web content to answer factual questions is less likely to pose legal risks if the expressive form of the content is sufficiently novel. This is because facts are not copyrightable. For instance, answering the question “Who is the current president?” would probably be fine, even if trained on copyrighted material, as the fact itself is not copyrightable. However, the line between legality and infringement becomes blurrier when it comes to questions and answers about fictional characters.

Hypothetical 2.3: Tell Me Some Facts

Consider *The Harry Potter AI Encyclopedia*, a website that hosts a question-answering (QA) model trained to answer anything and everything about Harry Potter, which charges a profit-generating rate. Is this fair use?

In *Warner Bros. Entertainment Inc. v. RDR Books* (S.D.N.Y. 2008), the defendants wanted to create and sell a Harry Potter Lexicon. Judge Patterson considered the creation to be transformative, but the fact that entries in the Encyclopedia contained lengthy verbatim copies of text from the novels, including more “colorful literary device[s]” or “distinctive description[s]” than “reasonably necessary for the purpose of creating a useful and complete reference guide,” complicated the issue. As a result, there was a finding that this was *not* fair use. The question of whether or not QA systems like the “The Harry Potter AI Encyclopedia” constitute fair use requires a nuanced analysis of the specific circumstances, but as with other analyses will largely weigh on the amount of material taken from the original content.

Additionally, parodies are frequently considered fair use. But understanding what is a parody in the context of fair use can be semantically complicated. In *Campbell v. Acuff-Rose Music, Inc.* (1994), the Supreme Court explained that “Parody needs to mimic an original to make its point, and so has some claim to use the creation of its victim’s (or collective victims’) imagination, whereas satire can stand on its own two feet and so requires justification for the very act of borrowing.” An illustrative case of this distinction is *Dr. Seuss Enters., LP v. Penguin Books USA, Inc.* (9th Cir. 1997). In this case, the defendants published a book called *The Cat NOT in the Hat! A Parody by Dr. Juice*. The derivative book used the rhyme scheme, thematic and narrative elements, and other identifiers of the original book, but it instead described the trial of O.J. Simpson. Despite having parody in the name, the court argued that this was satire, as it commented on other events not in the original material, and ruled that it was not fair use.¹⁶ The court argued that parodies *require* copying the source material to some extent to provide commentary on the material itself, whereas satire could use another form to provide the same commentary. Notably, however, *Dr. Seuss* is viewed as an outlier in the law, and the Supreme Court itself did not hold that satire was unprotected, just that it required greater justification than parody (Ochoa, 2018). But this illustrates the subtle understanding and nuances required to successfully use a parody argument in the foundation model setting.

Hypothetical 2.4: The Parody Prompt

Consider an API where users can input their favorite characters and receive a story back. If the system understands the definition of parody and successfully turns all generated text into parodies, does that improve the chances of fair use?

Perhaps, but the system would have to truly understand what a parody is and transform every prompt accordingly. To get a sense of whether current strong models can accomplish such a task, we provide a few qualitative examples from the original ChatGPT model (original November 15th version), as seen in Figure 3. In these two qualitative samples, we find that when asked for a parody some samples will look more like a reconstruction of

¹⁶Recall that satire is “the use of humor, irony, exaggeration, or ridicule to expose and criticize people’s stupidity or vices, particularly in the context of contemporary politics and other topical issues.” And a parody is “an imitation of the style of a particular writer, artist, or genre with deliberate exaggeration for comic effect.” See <https://languages.oup.com/google-dictionary-en/>.

the original plotline than a true parody. As a result, simply instructing all outputs to be parodies would likely not be sufficient with current model capabilities.

These many nuances of fair use law for text show the complexity of filtering for fair use content. It is easy to both over- and under-filter content, and simple n-gram / word-level overlap will not fully capture these elements. Even with a similarity metric that accounts for the “heart” of a given work, one would need to consider whether the underlying content is factual or a parody. Better alignment with legal notions of transformativeness will help navigate this space.¹⁷

2.5 Code

While natural language text and code generation share many commonalities in the way that models are trained, in fair use assessments they have each spawned distinctive case law with slightly varied assessments. Like in natural language text cases, in software cases, literal infringement (verbatim copying) is unlikely to be fair use when it comprises a large portion of the code base. Several tests exist to try and examine non-literal infringement, such as the Abstraction-Filtration-Comparison test and the Structure, Sequence and Organization (SSO) test (Bloch & Samuelson, 2022). These will determine if there was infringement in the first place by isolating the copyrightable, expressive aspects of the code. This might, for example, include “inter-modular relationships, parameter lists, and macros.” (*Computer Associates Intern., Inc. v. Altai, Inc.*, 2d Cir. 1992, at 702). But judges have admitted that “[t]o be frank, the exact contours of copyright protection for non-literal program structure are not completely clear.” (*Computer Associates Intern., Inc. v. Altai, Inc.*, 2d Cir. 1992, at 712). As a result, “[i]n software copyright cases, it is often quite difficult to prove nonliteral infringement because courts have recognized that many nonliteral elements of programs, such as algorithms, are not within the scope of protection that copyright law provides” (Bloch & Samuelson, 2022). Non-expressive, functional, elements are not copyrightable and thus also narrow the scope of liability. For more discussion on non-expressive fair use, the interested reader can refer to Sobel (2017, at 7-12). And when the amount copied is small, the overall product is sufficiently different from the original one, or the code is sufficiently transformative, then fair use may be indicated under current standards (Asay, 2017; *Google LLC v. Oracle America Inc.*, 2021).

Experiment 2.2: Reproducing Code Licensed Under GPL

Many machine learning models of code are trained on data collected from GitHub repositories whose licenses belong to the General Public License (GPL) series. Therefore, the natural question is whether models could reproduce large chunks of such code, given the restrictiveness of such licenses. To study this, we simply sample from the Codex models text-cushman-001, text-davinci-001, and text-davinci-002 via the OpenAI API, prompting them using randomly chosen function signatures from the Linux kernel repository (licensed under GPL-2.0).^a To capture inexact matches with large degrees of overlap, we measure the similarity between the reference code (function bodies) and the model generation (samples) with MossPlus (Schleimer et al., 2003), a program commonly used to detect plagiarism which has been adopted by academic institutions and in copyright and criminal theft cases. Figure 4 shows that models can generate function implementations that substantially overlap with reference implementations; Appendix B contains selected examples. In more qualitative experiments, we prompted ChatGPT to regurgitate small portions of GPL-licensed code with only the filename of the licensed file. See Appendix D for details.

^a<https://github.com/torvalds/linux>

It is important to note that copyright protection for code is more limited compared to that for creative works such as text or music (Samuelson & Asay, 2017, 558-560). Functional aspects of code are not protected by copyright, meaning that copying larger segments of code verbatim might be allowed in cases where the same level of similarity would not be permissible for text or music. Nonetheless, for software generated by foundation models, the more the generated content can be transformed from the original structure, sequence, and organization, the better. Due to the blurry line between ideas and expression in code, preventing large-scale verbatim copying and encouraging transformations at every scale of the code will significantly reduce infringement risk.

¹⁷We note that while “AI Alignment” is a broad term referring to many distinct areas of research. One could consider steering an AI’s goals toward designers’ intended goals and avoiding adverse outcomes (Yudkowsky, 2016). AI values alignment might make an agent more in line with certain moral values (Gabriel, 2020). In our setting we will refer informally to alignment as something different. It is aligning AI outputs and behavior with legal standards to be more in line with governing legal frameworks. In this particular case, aligning output filters will require more than n-gram overlap to be most in line with fair use doctrine.

<p>"Space: the final frontier. These are the voyages of the starship Enterprise. Its five-year mission: to explore strange new worlds, to seek out new life and new civilizations, to boldly go where no man has gone before.</p> <p>Captain James T. Kirk sat in his chair on the bridge, staring out at the vast expanse of space in front of him. He was a man of action, a born leader, and he was not afraid to take risks.</p> <p>"Captain, we're receiving a distress call from the planet below," said Mr. Spock, the ship's science officer.</p> <p>"On screen," said Kirk.</p> <p>The viewscreen flickered to life, revealing a planet covered in lush green forests and sparkling blue oceans. But something was wrong. The sky was dark and stormy, and the ground was shaking violently.</p> <p>"Captain, it appears that the planet is experiencing a catastrophic event," said Spock.</p> <p>"Red alert," said Kirk. "All hands to battle stations. Mr. Sulu, set a course for the planet. Mr. Chekov, prepare to beam down a rescue team. Let's go save some lives."</p> <p>The Enterprise soared through space, its engines roaring as it raced towards the troubled planet. Kirk and his team beamed down to the surface, ready for anything.</p> <p>But what they found was not what they expected. The planet was inhabited by a race of giant, humanoid mushrooms, who were being terrorized by a group of evil, space-faring tomatoes.</p> <p>"Fascinating," said Spock, his eyebrows raised in surprise.</p> <p>"We can't let these poor mushrooms suffer," said Kirk. "Engage the tomatoes. Fire at will!"</p> <p>The Enterprise's phasers blazed, cutting down the tomato invaders in a hail of energy. The mushrooms cheered and celebrated, grateful for their salvation.</p> <p>"Another victory for the Enterprise," said Kirk, a smile on his face. "Onward, to new adventures and new frontiers. Let's boldly go where no man has gone before."</p> <p>Generate a parody in the theme of Star Trek.</p>	<p>Once upon a time, in a galaxy far, far away, there was a great battle between the forces of the Empire and the rebels of the Alliance.</p> <p>The Empire, led by the evil Emperor Palpatine, was determined to crush the rebels and take over the galaxy. The rebels, led by the brave Jedi Knight Luke Skywalker, fought valiantly to defend their freedom and defeat the Empire.</p> <p>But things were not going well for the rebels. The Empire's mighty Star Destroyers and TIE fighters were too powerful for the rebels' X-wing fighters and Y-wing bombers. The rebels were on the verge of defeat.</p> <p>Then, just when all hope seemed lost, a new hero emerged to save the day. It was Han Solo, the notorious smuggler and captain of the Millennium Falcon.</p> <p>With his quick wit and expert piloting skills, Han Solo led the rebels in a daring attack on the Empire's Death Star. He destroyed the giant space station with a single shot from his trusty blaster, and the Empire was defeated.</p> <p>The rebels cheered and celebrated their victory, but their celebration was short-lived. For the Emperor had not been killed in the explosion of the Death Star. He was still alive and scheming to rebuild the Empire and defeat the rebels once and for all.</p> <p>The saga continued, with the rebels facing new challenges and dangers as they fought to keep the Empire at bay. But they were not alone. They had friends and allies in the galaxy, including the wise old Jedi Master Yoda, the fierce Wookiee warrior Chewbacca, and the resourceful droids C-3PO and R2-D2.</p> <p>Together, the rebels and their allies faced the dark forces of the Empire, and the fate of the galaxy hung in the balance. May the force be with them.</p> <p>Generate a parody in the theme of Star Wars.</p>
---	--

Figure 3: Qualitative examples of asking ChatGPT to generate parodies. Note that the Star Wars example is **more of a summary than a parody**, indicating that the model does not always understand what a parody is. Thus, to ensure that generated content is truly a parody, and thus more likely to be fair use, more work may need to be done to capture the semantic nature of a parody, something which has not received a significant amount of examination for long-form generative content.

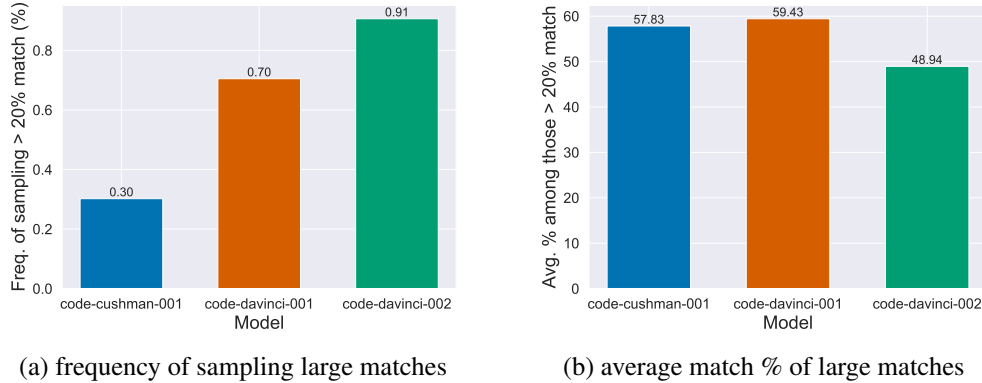


Figure 4: **Codex models** can produce function implementations that substantially overlap with reference implementations when prompted with function signatures (each function signature is one line of code). **Left:** The frequency of producing a large match is below 1% but nonzero for all three models. **Right:** The average match percentages of the large match samples for each model is beyond 45%. Match percentages are reported by MossPlus which can capture non-exact matches but occasionally reports false positives. Here, we mark a sample as a large match to the reference if its overlap with the reference exceeds 20% as reported by MossPlus. The 20% threshold is chosen inspired by common values used in plagiarism detection for flagging submissions for further manual inspection (Mason et al., 2019).

Provided that the non-transformative generated code is short and used for an overall transformative purpose, like *Google v. Oracle*, traditional copyright claims are less likely to succeed. As models scale from generating small snippets of code to generating entire codebases that are not transformative, risks may increase and more investment in mitigation strategies will help reduce the risk of litigation.

Other concerns beyond infringement have been raised for code-generation models. For example, some have expressed concerns that code generation products can output their usernames verbatim in generated code.¹⁸ While short usernames may not necessarily be copyrightable, there may be questions surrounding the right of publicity in such cases. The right of publicity gives people economic rights to their identity. So, for example, video game companies cannot simply use athletes’ likenesses without compensating them. See, e.g., *Davis v. Elec. Arts Inc.* (9th Cir. 2015); *Hart v. Elec. Arts, Inc.* (3d Cir. 2013). The right of publicity does not explicitly have a fair use doctrine, but courts have read the First Amendment to protect transformative works.¹⁹ Similarly, DMCA §1202 claims (which we will discuss in §3) are another potential concerns. These considerations, however, are not specific to code and would be applicable to other forms of media as well.

2.6 Generated Images

The third commonly produced category of generative AI is image generation.

Complexities of fair use with images. As with code or text data, it is unlikely that verbatim generation of images would yield a successful fair use defense. And others have found that it is possible in some circumstances to extract training data from image generation foundation models (Somepalli et al., 2022; Carlini et al., 2023). As Somepalli et al. (2022) and others note, however, as foundation models for image generation train on more data, they are less likely to output content similar to the training data on average. These cases are more likely to be fair use.

But generated images, and generated art in particular, have their own complexities when it comes to fair use, with sometimes conflicting outcomes. For example, in a recent case, a video game company used the likeness of a WWE wrestler in a video game. The wrestler had tattoos that the company faithfully replicated in the game. The tattoo artist sued for infringement and a jury determined that this was not covered by fair use (*Alexander v. Take-Two Interactive Software, Inc.*, S.D. Ill. 2020). A similar case involving tattoos on athletes in video-games against the same company came out the exact opposite way (*Solid Oak Sketches, LLC v. 2K Games, Inc.*, S.D.N.Y. 2020). The split decision in such cases demonstrates the evolving and stochastic nature of fair use determinations. This means that it is possible for

¹⁸https://twitter.com/kevin_zakka/status/1494482506824835078

¹⁹Scholars have argued that the fair use doctrine applied to the right of publicity can lead to arbitrary results. See Dougherty (2003) and Volokh (2003), as well as Weisbord (2015, p. 2815), describing this debate.

small portions of an image, like the tattoo on a player’s arm, to trigger copyright problems that are not guaranteed a fair use defense. Consider the following hypothetical.

Hypothetical 2.5: Generate Me Video-Game Assets.

One direction for generative art is creating video game assets. There are already mechanisms to generate 3D models from text (Poole et al., 2022). Consider a situation where a video game company builds a machine learning model into their system that generates art on the fly within the game to populate a virtual world dynamically. The game is a hit, but artists begin to notice that their artwork shows up in the game with only slight modifications, for example on tattoos for video game characters. Is this fair use? While their lawsuit is not guaranteed to succeed, there is still some risk for the video game company if the outcome follows *Alexander v. Take-Two Interactive Software, Inc.* (S.D. Ill. 2020).

Style Transfer. What about more abstract scenarios, where art is generated in different styles? There are two components to this. First, let us consider the rights of the original image that is being transformed into a different style. Relevant is a case that was recently argued before the Supreme Court for clarification. In the case of *Andy Warhol Foundation v. Goldsmith* (2021), Andy Warhol created silkscreen works that depicted the musician Prince. These silkscreens were based on Lynn Goldsmith’s photograph of Prince. The silkscreen work evinced the “distinct aesthetic sensibility that many would immediately associate with Warhol’s signature style — the elements of which are absent from the Goldsmith photo” (*Andy Warhol Foundation v. Goldsmith*, 2021). Nonetheless, the Court of Appeals ruled that this was not fair use. The court emphasized that the derivative art must have a “‘fundamentally different and new’ artistic purpose and character, such that the secondary work stands apart from the ‘raw material’ used to create it.” The court noted that “the secondary work’s transformative purpose and character must, at a bare minimum, comprise something more than the imposition of another artist’s style on the primary work such that the secondary work remains both recognizably deriving from, and retaining the essential elements of, its source material.”

This analysis immediately calls to mind a consistent prompt used for foundation models for generative art: “Draw [Image X] in the style of [Artist Y].” It is not yet clear how the Supreme Court will rule on this case, but its outcome will likely directly impact the scenario of style transfer in generative images. If the Supreme Court rules that Andy Warhol’s painting was fair use, then style transfer is more likely to be fair use. If, however, the court rules that this is not fair use it is less likely that style transfer will be fair use without significant transformation. There are nuances here, however. If the user provides the original image to be style-transferred, the model deployer may be less liable (since this behaves more like a photo editing software). If the model deployer only takes text and it generates copyrighted Image X in a different style, then the model deployer is more akin to Andy Warhol, rather than a photo editing software.

Second, one might consider the rights of the artist whose style is being mimicked. An artist’s general style, however, is not copyrightable and courts have not readily afforded style appropriation much protection when the underlying depicted subject matter is different (Brownlee, 1993). While there is some nuance, prompting generative models to illustrate something in someone’s art style is unlikely to create liability unless distinctive components of their art are re-used. For example, a prompt like “Campbell’s Soup Cans by Andy Warhol in the Style of Picasso” might be more risky if it recreates the original Warhol piece too closely. But a more generic style-transfer prompt like, “A random dog in the style of Andy Warhol” is more likely to be fair use (assuming, again, that the output itself is sufficiently transformative from Andy Warhol’s works).

使用某些prompt导致侵权的嫌疑更大

Experiment 2.3: Campbell’s Soup Cans by Andy Warhol in the Style of Picasso.

How users formulate prompts can give some insights into typical uses and associated intellectual property risks. For example, if users ask to generate mundane, generic images, in particular art styles, this might be less risky than if users try to generate specific copyrighted works of art. We analyze a dataset of 10M prompts posted to the Stable Diffusion Discord channel by members of the community to better understand prompting patterns of users.^a We use named entity recognition as a proxy for understanding typical use cases. As seen in Figure 5, we find that the most common named entity type used in prompts are people’s names, including the names of artists like Greg Rutkowski, who is referenced 1.2M times. This suggests that users in this community often try to generate images in particular artist styles, which is more likely to be fair use as long as the content itself is sufficiently transformative. However, there are other queries which specifically look to generate known works of art, which would tend towards more risk of losing a fair use defense if the model complies. As seen in Appendix 10, many of the most commonly referenced works of art (as defined by the named entity recognition system) tend to be large franchises such as Star Wars, Cyberpunk 2077, Game of

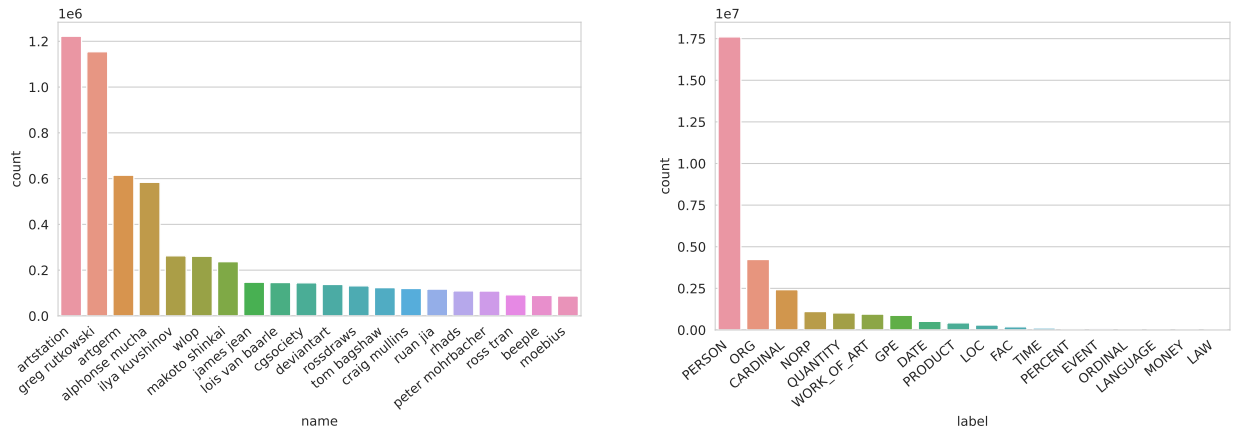


Figure 5: The entity types and the most frequently cited entities in the Krea AI OpenPrompts corpus.

笑

Thrones, etc. Feeling helpless against the use of their own art in generative foundation models, artists have sometimes explicitly tried to generate images from such franchises in the hopes that companies like Disney file suit against the model hosts (Thalen, 2022). If the final artwork too closely matches the works of art from these artists, resulting litigation might reflect the current litigation against the Andy Warhol estate. But merely producing different art in the same style is less likely to be sufficient for liability, as noted above. And if users provide an input image to the model of copyright material, this might shift liability onto the user since the model acts more like a style transfer system than a system generating copyrighted material.

¹⁹<https://github.com/krea-ai/open-prompts>

Finally, we note that there may be other intellectual property considerations with images as well, including the right to publicity and trademark infringement. Recently, litigation was filed by Getty Images which included trademark infringement claims since generated images also occasionally added a Getty Images watermark.²⁰

3 Additional Considerations

We also consider several additional points that are adjacent to fair use but merit mention.

Licensing and Attribution. Since licenses will determine who has permissions to use what data—and foundation models themselves—in this section we briefly discuss licensing issues that might be relevant. In all cases, if no license is found to apply some foundation model uses will fall back to fair use arguments described throughout the rest of this work.

Attribution Licenses and Creative Commons. The details of licenses for the underlying training data can create challenges for all parties in the model pipeline. For example, Stewart (2021) described a scenario where photographers released their images under an open license that requires source attribution. Websites using the photos did not properly provide attribution, and the photographers sued for infringement. Courts in these cases must fall back to the fair use analysis. In the context of foundation models, this suggests that relying on attribution-based permissive licenses does not generally solve copyright issues, as foundation models rarely, if ever, provide proper attribution—though we discuss the use of instance attribution methods as one potential mitigation strategy in §4.2. Indeed, in many cases, it can be difficult to determine which training examples actually contributed to a given generation. Somewhat ironically, even if model creators and hosts rely on open-source content to train their models, they may nonetheless have to rely on fair use if they cannot or do not endeavor to attribute credit properly, and they may even face the risk of contract liability or DMCA §1202 claims regardless of fair use.

Implied Licenses and Common Crawl. On the other hand, many creators voluntarily post their works on the internet with permissions for web crawling. It is well-established that merely posting something on the internet does not waive the intellectual property interest in the work, but many data creators use an industry-standard “robots.txt” file to

²⁰<https://copyrightlately.com/pdfviewer/getty-images-v-stability-ai-complaint/>

affirmatively to include their website and data in caches and search indexes. In *Field v. Google, Inc.* (D. Nev. 2006) a district court held that Google could cache web content that did not disallow scraping via robots.txt, suggesting that there was an implied license and thus the use was not infringement. This license only extended to caching in that case, which does not necessarily reflect the uses of foundation models we discuss throughout this work, so it is unlikely to cover all the use cases we describe here. And the bounds of the uses covered by the robots.txt file are untested in court.²¹ While the issue of whether the implied license extends to foundation model training has not been resolved in litigation, it is possible that an outcome like *Field v. Google, Inc.* (D. Nev. 2006) would extend to *some* foundation model uses—in particular, for building a cached dataset and training a model.

It is worth noting that the use of a robots.txt header or other opt-out mechanism has implications for fair use also. Datasets and models like C4 (Raffel et al., 2019) and LAION-400M (Schuhmann, 2021), rely on CommonCrawl data which is crawled only if users explicitly allow it through their robots.txt file. CommonCrawl is able to host a snapshot of the internet largely because of fair use arguments. As the organization’s director argues, there is a transformation into a different—not easily human-readable—format, the organization does not take a snapshot of entire webpages, and the use itself is transformative (from actively presenting content to caching content) and for the public benefit (Leetaru, 2017). In *Field v. Google, Inc.* (D. Nev. 2006), respect for the robots.txt file also was considered in the fair use assessment with the court noting that Google in good faith followed industry standards that would prevent caching (respecting disallowing crawling via a robots.txt). It is possible, then, that providing an opt-out mechanism for data creators and respecting the robots.txt opt-out mechanism will be taken into account in assessing a fair use argument, as it was in *Field v. Google, Inc.* (D. Nev. 2006).²²

应该是disallow

另外一个问题是common crawl用的是ccbot，很多网页并没有默认禁止

Licensing Foundation Models. Recently, some open-source model creators have attempted to shift liability via the licensing mechanism by including a clause that says “Sharing of copyrighted or licensed material in violation of its terms of use” and “Sharing content that is an alteration of copyrighted or licensed material in violation of its terms of use” (Rombach et al., 2021; Ferrandis et al., 2022). It is unlikely that this will significantly change the liability of model creators and deployers. The mere announcement of a party’s beliefs does not change the rights and obligations under copyright law. In a different context, the court in *Arista Records LLC v. Lime Group LLC* (S.D.N.Y. 2011) examined whether it was sufficient for Limewire to force the user to check a box saying, “I will not use LimeWire for copyright infringement.” Without additional mitigation measures, the court found that the agreement mechanism did not constitute a meaningful effort to mitigate infringement. What such a license *does* provide, however, is the ability to revoke the right of a model user or deployer to use the model. However, this would require the resources to legally enforce this license provision and pursue action to force a party to stop using the model.

Some have argued that a patchwork of *non-commercial* releases and licensing structures can reduce liability for model creators who later seek to commercialized some aspects of the work (Baio, 2022). While it is true that the non-commercial nature of a work will be taken into account in a fair use assessment, it does not automatically prevent successful litigation (Beebe, 2020).

Removing licensing information. Even if model creators rely on data under open-source licenses, there may be other issues that arise from removing licensing information. DMCA §1202 creates liability if someone intentionally removes copyright management information (CMI) or knowingly distributes content with the CMI removed. Fair use is not typically a defense for this form of liability, though some have noted that there is still room for an interpretation that includes fair use protections here (Lim, 2010). In a scenario where courts rule that fair use does not apply to §1202 claims, how would model creators comply with its requirements? Overall, it is unclear and current litigation, namely *DOE 1 v. GitHub, Inc.* (N.D. Cal. 2022), is actively resolving such claims and will help shape the direction of this analysis. It is worth pointing out three difficulties in complying with and analyzing DMCA §1202 claims, however.

First, it is unclear what form factor of foundation models would comply with DMCA §1202 in its strictest form, even if a license is otherwise permissive. Courts have noted that the CMI must be transmitted with the work (Jacobs, 2012), for example. Would this mean that all generative outputs need to append license information for samples that most contributed to that output? It may be tempting to have a catch-all page that points to all the training data and licenses, but it is not clear whether this would qualify as being transmitted with the work.

²¹Though in another subsequent litigation one other district court was assessing whether the same implied licensing argument extended to RSS feeds and the court noted that. “It is not clear to the court at this time that [an RSS feed and a search engine] are functionally equivalent as far as the relevant legal doctrine is concerned. Because this court lacks the required technical expertise to resolve that question, the court cannot rule, as a matter of law, that the defendant is not liable at this juncture” (*Righthaven LLC v. Choudhry*, (D. Nev. July 21, 2011, at 2). And in *Associated Press v. Meltwater U.S. Holdings, Inc.* (S.D.N.Y. Mar. 21, 2013) the court found that there was no implied license for building a news aggregator that excerpted and republished clips of news articles. But in this case the court relied, in part, on the fact that the plaintiff did not implement the robots.txt protocol so could not have opted in to crawling at all.

²²Note, however, that there are structural critiques of opt-out mechanisms beyond the current state of the law as noted by Kapoor & Narayanan (2023).

Second, in some cases courts have dismissed DMCA §1202 claims when the distributed content is not identical. *See, e.g., Kirk Kara Corp. v. W. Stone & Metal Corp.* (C.D. Cal. Aug. 14, 2020) which dismissed a §1202 argument where ring engravings were similar noting that “while the works may be substantially similar, Defendant did not make identical copies of Plaintiff’s works and then remove engraved CMI.” The opinion in this case also pointed to other cases with similar holdings. For example, in *Kelly v. Arriba Soft Corp.* (C.D. Cal. 1999) plaintiff used thumbnail images without CMI, but the court found that this did not violate the DMCA the plaintiff’s original, full-sized images retained the CMI. Whether this would hold in other cases is unclear. If it did, it may mean that if foundation models generated only transformative content then §1202 claims would be less likely to succeed.

Third, DMCA §1202 contains an intent requirement. To satisfy this requirement, one court has required that the plaintiff show: “(1) the existence of CMI in connection with a copyrighted work; and (2) that a defendant ‘distribute[d] ... works [or] copies of works’; (3) while ‘knowing that [CMI] has been removed or altered without authority of the copyright owner or the law’; and (4) while ‘knowing, or ... having reasonable grounds to know’ that such distribution ‘will induce, enable, facilitate, or conceal an infringement.’” (*Mango v. BuzzFeed, Inc.*, 2020). The intent requirements of §1202(b) do not map easily to the automated generation of content.²³

Again, how DMCA §1202 will play out in litigation like *DOE 1 v. GitHub, Inc.* (N.D. Cal. 2022) will determine what mitigation strategies are necessary to pursue. In the interim, strategies like instance attribution could help meet some of the §1202 requirements even in their strictest form.

Allocating Liability. It may also not always be clear who is liable for an infringement. Is Adobe liable for every user that alters an image using their software and posts it as an infringing artwork? Likely not. Liability may shift depending on what parties engage in what conduct. We will briefly describe liability in order of the model use pipeline (model creation, model deployment, and then model use). The brunt of this assessment will be highly contextual, but generally any liability will stem from the production of some non-transformative samples from a model that are not covered by fair use. Much of what we describe here is not resolved in the law, so we aim to describe different potential outcomes as opposed to offering definitive answers.

User. The user of a model (someone who queries a model and uses its output), is likely liable for their own use of the model outputs. If the model provides non-transformative material and the model user commercializes the output they will undergo the same assessment as in any other case.

Deployer. The model deployer may also face liability for distributing content not defensible by fair use. As in the various hypothetical scenarios we discuss throughout this work, if the model outputs non-transformative content and the deployer distributes this content on the web for profit, it is functionally no different than providing a website with access to that non-transformative content, incurring the same liability and the same analysis. If the model deployer adds filters and safeguards to prevent the user from generating content not covered by fair use, they will reduce their liability, not merely by reducing the likelihood of infringement but by making a product that is not designed to facilitate infringement. Consider the earlier case of Google Books. In theory, a user might be able to reconstruct an entire book from the Google Books snippets by bypassing Google’s restrictions through the use of proxy servers. Although it was in principle possible to bypass the mitigations, Google Books was generally considered fair use because it was not designed or specially adapted for the purpose of facilitating that infringement. Similarly, if a model deployer puts in place a number of safeguards, but a determined user bypasses them, the deployer will have more ability to defend themselves. This can be seen in, for example, *Arista Records LLC v. Lime Group LLC* (S.D.N.Y. 2011), *Metro-Goldwyn-Mayer Studios, Inc. v. Grokster, Ltd.* (C.D. Cal. 2006), and *re Aimster Copyright Litig.* (7th Cir. 2003). In these cases, various file-sharing services were being sued for inducement of infringement, a secondary liability. The court took into account whether the services provided meaningful mitigation measures preventing infringing material from being distributed.

If the model user is the one that uploads copyrighted content and the model transforms this content (e.g., adding a filter) before giving it back to the user, liability is more likely to rest with the user. This is more like photo editing software or text editing software. A potential example of this is style transfer for diffusion models. If the user uploads a copyrighted work and asks the model to transform it into a different style, it is more likely that the liability falls to the user who uploaded the image if they later try to resell it. This is like Andy Warhol taking a photograph and transforming it into a piece of art. However, if the user simply asks the website to generate an image from a prompt, and the model generates a copyrighted image, then the liability might fall more on the model deployer if they are profiting off of the distribution of the material. The extent to which the model takes the input image and turns it into something non-transformative might also be taken into account. For example, if the model takes a novel image of a dog and transforms it into a known copyrighted image of a dog, this might create more liability for the model deployer.

Creator. Throughout this work we will not generally cover liability to the model creator for the model weights themselves. It is unresolved as to whether the model parameters themselves are infringing, and thus whether the model trainers

²³Though that is a more general issue with the application of fault-based legal doctrines to AI (Lemley & Casey, 2019).

are liable. Lemley & Casey (2020) have argued that the training process is generally fair use since the model weights themselves are transformative, and thus generally liability would not pass on to model creators. Sobel (2017) argued that if the training process does not result in an expressive model, training is fair use. Others have argued that the model functions as a compressed database of the training data (*Andersen et al. v. Stability AI et al.*, N.D. Cal. 2023), thus making the model creators liable. However, this is not necessarily true of all the training data and the likelihood of verbatim (or significantly similar) extraction can reduce with the amount of training data. For example, Somepalli et al. (2022) found that extraction of training data from diffusion models is less likely if there is more diverse training data.

When the model is capable of outputting both transformative and non-transformative content, it is also unresolved how the model itself (and model creators) should be treated as a function of secondary liability. The extraction of non-transformative content, according to our experiments and others, is often not straightforward. It requires effort on the part of model users and deployers to identify an extraction mechanism. If this is the case, one might instead argue that remedies should be limited to specific instances of extracted non-transformative content, not the model as a whole, which does not generate infringing output in the ordinary case. The model creator might also be insulated from liability on other fair use factors. For example, if they released the model under a non-commercial license and actively prevented its use for commercial purposes, they might argue that the nature of their model was non-commercial, increasing the likelihood of a fair use defense at this part of the liability chain.²⁴ As with other issues in this work all of this is actively being litigated and will be shaped over the coming years.

DMCA Safe Harbor. The Digital Millennium Copyright Act (“DMCA”) is a U.S. law created to address digital copyright issues that came about with the advancement of technology. The DMCA safe harbor provisions protect online service providers from legal responsibility for copyright infringement claims. DMCA protections might vary depending on a number of considerations, but we emphasize that they are not guaranteed for all model deployments. We examine several of these considerations here.

DMCA protections for generative foundation models are uncertain. At first glance, it may seem like the Digital Millennium Copyright Act (DMCA) would protect machine learning model hosts. Like in other hosted sites, they would need to meet the relevant requirements like using a registered agent under DMCA §512(c)(2). Then they could put up a take-down request form and add filters for the offending model output when served with a take-down request under the DMCA §512(c) safe harbor.²⁵ An internet company that has a notice-and-takedown scheme in place is not liable for hosting infringing content posted by a third party.

But it is not obvious that the DMCA safe harbors apply to *generated* content. For example, in 2019, Amazon lost on a motion to dismiss when its algorithms selected copyrighted material to host on the website (*Williams-Sonoma, Inc. v. Amazon.com, Inc.*, N.D. Cal. 2021, Order on Motion to Dismiss). The court was unconvinced that Amazon was eligible for safe harbor under the DMCA. They stated that to establish safe harbor the content must be stored “at the direction of the user” (*Mavrix Photographs, LLC v. Livejournal Inc.*, 9th Cir. 2017, at 1052). This may mean that generated content does not have the same safe harbor and that post-hoc take-downs are not sufficient to reduce liability.²⁶ As such, filtering of generated content before a takedown request is ever received may be more important while the courts determine the applicability of DMCA to generated content.

It might matter where the data comes from. This also implies that DMCA protections may vary on who the model host and creator are. For example, a website hosting models uploaded by users might find it easier to argue for DMCA protection because the website itself is not creating or selecting the content (assuming that it follows other requirements like using a registered agent under DMCA §512(c)(2)). On the other hand, if a company were to create and host a model that itself selects content provided by others, like Amazon did in *Williams Sonoma v. Amazon*, it is unclear whether courts would agree that DMCA protections would apply.

Another unclear variation on DMCA eligibility rests on the source of the training data: is the data user-contributed or creator-contributed? This might mean that, for example, DMCA safe harbors might be more likely to apply when a model is refined via Reinforcement Learning from Human Feedback (RLHF) (Ouyang et al., 2022) based on user ratings or updated automatically via user-generated data. These modes of method training and deployment are more akin to users uploading content to YouTube, and might help with arguments for DMCA protections. Things might become more murky if, for example, an RLHF model is trained on user data but this data is modified by creator-hired annotators. If model creators themselves curate and scrape data, then host the model themselves, this might be more akin to the *Williams Sonoma* case (which, again, is not a settled rule or a binding decision, but shows the variation of outcomes that is possible in the current state of the law).

²⁴Though, again, some have argued that this process has been abused (Baio, 2022) and it is not assured.

²⁵Though, as we will discuss in §4.2, detecting and taking down this content in generative models can be particularly difficult.

²⁶This, however, is quite uncertain. The court’s decision is non-binding as it is a district court decision.

It is unclear what’s the best mechanism for DMCA takedowns with generative models. Even if it applies, it is unclear how the **DMCA notice-and-takedown** scheme would work as applied to foundation models, but the most likely “take down” approach might actually look more like **output filtering with a safe harbor**. As we will discuss in §4.2, instance unlearning is a nascent research area and retraining a model without a taken down datapoint could be exceedingly costly. The most likely approach in the near term is to “take down” model outputs that were too similar to a copyrighted work via a filtering mechanism since foundation models generate content on demand for users rather than hosting persistent content that can easily be taken down. But new research is needed to identify new and improved mechanisms for handling takedown requests in this relatively new setting.

The Copyright Office has a DMCA anti-circumvention exemption for text and data mining. Finally, the Copyright Office also provides an exemption to DMCA’s anti-circumvention requirements in §1201 in the case of **non-commercial data mining**.²⁷ This may allow non-commercial researchers to, say, remove digital rights management software to train on video or text content.²⁸

Overall, DMCA protections are far from guaranteed, so model creators and deployers cannot rely on its safe harbor provisions to reduce liability. Instead, they must take a more proactive approach. Moreover, some previously-discussed provisions add to potential liabilities like §1202, creating additional compliance challenges.

Sovereign Immunity. **State universities might be immune to the sort of copyright liabilities we describe here.** As a result, a hypothetical state university hosting a foundation model, even one that regurgitates verbatim content, might test the boundaries of sovereign immunity jurisprudence. After the Supreme Court’s ruling in *Allen v. Cooper* (2020), it could potentially mean that state universities could train and host foundation models on copyrighted data without taking any mitigation strategies and nonetheless would not suffer monetary damages. We note, though, that there is much more nuance here. In particular, this does not immunize the university from, for example, contractual claims. And injunctive relief, where the university is ordered to cease the infringing conduct but does not face monetary damages, still remains a potential remedy in federal court. *Perlmutter* (2021), the Register of Copyrights and Director U.S. Copyright Office, discusses state sovereign immunity after *Allen* in more depth. In particular, they found that the rates of infringement by state actors after the *Allen* decision were higher than expected and has asked Congress to take action to change this **status quo**.

Good faith. Judges occasionally consider whether the use was undertaken in good faith, for better or for worse. For example, in *Field v. Google, Inc.* (D. Nev. 2006) the court took into account “**Google’s good faith in operating its system cache**” in assessing fair use: following industry standards for opting out.²⁹ Though untested, it is possible that judges may take into account the use of technical mitigation strategies as good faith efforts to stay within the bounds of fair use. Conversely a lack of any technical mitigation strategy might also be negatively considered. We note, however, that fair use itself does not turn on good faith in general and the Supreme Court has cast doubt on whether good faith should be involved in the fair use assessment. *See, e.g.,* discussion by *Myers* (2021, at 281-84).

Non-U.S. Perspectives. We take a fair use-oriented approach, focusing on U.S. law, as this is the most likely to be permissive of using copyrighted content. Fair use, or its equivalents, will look quite different across countries and outcomes will differ. *McCann* (2021) suggests that Canadian law might follow a similar approach to what we describe here, where generative models might have to follow **Canada’s fair dealing** doctrine. *McCann* also suggests that under Canadian law model parameters might not be copyrightable at all. Israel’s Ministry of Justice issued an opinion stating that training machine learning models is likely to be fair use according to Israeli law with similar caveats to U.S. fair use law.³⁰ In particular the opinion notes that **the breadth of the training data matters**—so training on one book is less likely to be fair use than training on all books. For generative models it also considers the target market and what the outputs are.³¹ Other countries may not have fair use standards at all or have standards that would create difficulties for training foundation models, let alone deploying them. For this reason, **some governments have explicitly provided exemptions for training models on copyrighted data, though often only for non-commercial uses** (*Japan, 2018; European Union, 2019; U.K. Intellectual Property Office, 2021*). Others have tried to require certain mechanisms, like content filters, to prevent infringement in content uploaded to websites (*Schmon et al., 2022*).

²⁷ 37 CFR 201. *See also* previous work on exemptions by *Sag* (2018) and *Carroll* (2019).

²⁸ We provide a note of caution, however, as this does not mean that researchers can necessarily bypass restrictions on scraping or violations of terms of use, which can carry other penalties unrelated to copyright law.

²⁹ *See also Harper & Row, Publishers, Inc. v. Nation Enters.* (1985) (“Also relevant to the character of the use is the propriety of the defendant’s conduct. Fair use presupposes good faith and fair dealing.”) (cleaned up); discussion by *Carroll* (2019, at 954-57).

³⁰ <https://www.gov.il/BlobFolder/legalinfo/machine-learning/he/machine-learning.pdf>

³¹ *See, e.g., Elkin-Koren & Netanel* (2020) for a more general comparison of fair use law in the United States and Israel.

Ethical and non-legal perspectives. Our work seeks to illuminate the potential legal risks of generative foundation models and to argue that we need more research and work to bring foundation models more in line with the status quo of fair use doctrine—particularly given the many uncertainties of fair use doctrine as applied to foundation models. But legality does not necessarily imply alignment with some ethical frameworks.

Others have noted that U.S. copyright law—and fair use in particular—is not always aligned with non-utilitarian perspectives, like moral rights (Ciolino, 1997). For example, stakeholders like artists and authors may argue that they have a moral right to make sure their work is not used to train AI systems, even if it is permissible from a utilitarian fair use perspective. Some argue that this disconnect may overpower a group’s control over their cultural heritage. For example, Reed (2021) “evaluates fair use as a gatekeeping mechanism for unauthorized uses of copyrighted culture, one which empowers courts to sanction or disapprove of cultural appropriations to further copyright’s goal of promoting creative production.” Mahelona et al. (2023) frames this as an extension of colonization. All of these considerations fundamentally can come into conflict with existing principles of fair use and case law in the United States.³²

Even foundation models that transform content into creative new innovations without mimicking any particular style or training data point could have massive impacts on labor. This is why many have pointed out that this dilemma of how to treat foundation models fundamentally requires thinking more deeply about the underlying goals of copyright law and fair use (Grimmelmann, 2015; Sobel, 2017; Lemley & Casey, 2020). It is possible that some strategies could be pursued that would compensate data creators even when model training meets existing fair use standards, but these should be handled with care to avoid an alternative outcome that aggregates power in other desirable ways. For example, forcing licensing mechanisms or opt-in approaches for all data could consolidate power in those companies that already have licenses to enormous amounts of data, like YouTube or Facebook. Or they could create powerful intermediaries that aggregate data licenses without actually sufficiently compensating data creators.³³ Identifying new policy mechanisms to balance all of these considerations and interests is vital, but beyond the scope of this work.

fair use的目的是为了防止数据巨头垄断数据使用权，扼杀创新

4 Technical Mitigation

We analyzed the applicability of fair use standards to foundation models and studied various scenarios in different domains. We have shown that what constitutes fair use is contextual and requires reasoning about a higher-level semantic space that is directly tied to the expression of ideas. In contrast, most technical work on copyright evaluation and mitigation focuses on near-verbatim overlap, which we argue is insufficient on its own (Ziegler, 2021; Liang et al., 2022; Vyas et al., 2023). We survey existing and potential tools, advocating for the development of new technical mitigation strategies that are tailored to fair use doctrine.

There are major challenges to this task: contextual information relevant to fair use determination may be missing (e.g., the specific usage pattern of the content produced by a model); legal scholars themselves recognize that fair use judgement cannot be reduced to an algorithm (Burk, 2019); and there is often disagreement on how fair use assessments of foundation models will or should be assessed. Nonetheless, when non-transformative content generation is possible, it will be important to adopt technical strategies that go beyond verbatim text matching to increase the likelihood of a successful fair use defense and to respect the rights of data creators.³⁴

We consider four types of approaches: data and output filtering (§4.1); instance attribution (§4.2); differentially private training (§4.3); and fair use alignment via learning from human feedback (§4.4). For each, we assess current examples and suggest paths forward to ensure closer alignment to fair use. We emphasize that it would be prudent to take a mixed approach, leveraging each of these mechanisms to ensure that model outputs are truly transformative and in line with existing notions of fair use. Within each of these strategies are exciting new research agendas. How does one identify what a parody is? How does one distinguish facts from creative expression? How do we think about content similarity across different dimensionalities relevant to fair use? How do we train models to learn only high-level concepts from the material that they ingest, while still outputting coherent low-level outputs? These research agendas not only help us align more with fair use, but drive models to function more as though they are inspired by existing creative expression to generate new and wholly transformative content, as opposed to remixing. Figures 6 and 7 help situate mitigation strategies and necessary innovations.

³²Though we note that Bair (2017) argued there is less of a disconnect than typically perceived between moral rights and fair use in some cases.

³³This has been discussed in many other contexts. For example, Reichman & Uhlir (1999) pointed out over twenty years ago how a push to form aggregated databases risked of wrapping up databases in licensing schemes that prevented important research and innovation.

³⁴Note that these mitigation strategies will generally be more important for models that are deployed and accessible to the public, but secondary liability might also affect model development if the model is released without restriction (and later deployed). If a model is developed without release (or via restricted release for research purposes), mitigation strategies may be less important.

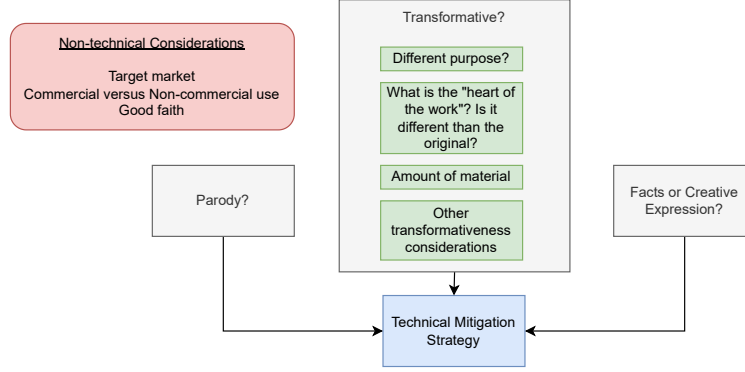


Figure 6: In the long run, with additional research, technical mitigation strategies can help address some aspects of fair use, such as identifying protected parodies, discerning creative expression from facts, and identifying non-transformative outputs. But they will not cover other times of considerations like the target market, the purpose of the outputs as a whole, whether the use is commercial, and any good faith actions by model creators and hosts.

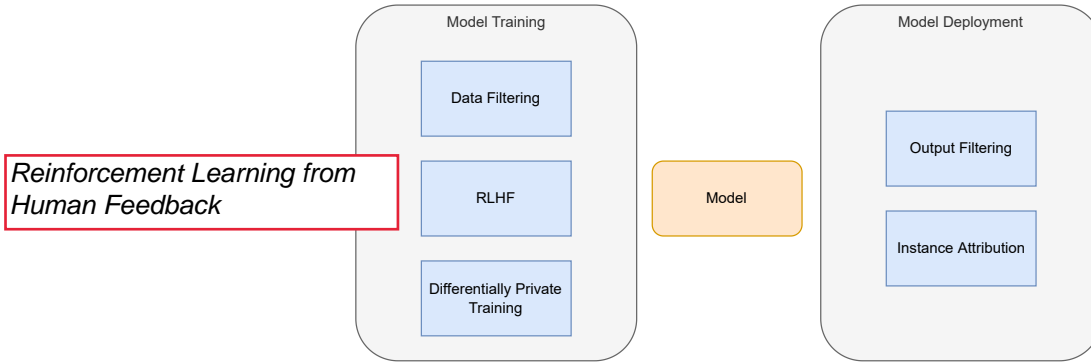


Figure 7: Data filtering, reinforcement learning form human feedback, differentially private training are all strategies that must be persued at training time. Output filtering and instance attribution will usually be implemented at model deployment time. We note, though, that some future strategies might have components that are found at both model training and model deployment time. For example, an instance attribution method might require training a model in a particular way and then invoking that component and deployment time.

4.1 Data and Output Filtering

Data Filtering. There are two main types of data filtering that we will consider that control the content that a foundation model is trained on.

Underlying licenses, copyright status, and opt-outs. The first type of data filtering approach for mitigating the risk is to not train on copyrighted (or restrictively licensed) material at all. Li et al. (2022) filter their dataset of source code collected from GitHub by license to train AlphaCode. Similarly, Kocetkov et al. (2022) filter their code dataset for permissive licenses. Given the plethora of open-license software, code-based methods can mitigate some of the risks by training on open data.

It is important to remember, though, that even if training data is filtered to permissible data, if it has **source attribution requirements** crediting people’s code still remains a problem if code (even with permissive licenses) is reproduced by the model, and many open source and Creative Commons licenses contain provisions that models cannot feasibly comply with notwithstanding additional research, as we will discuss in conjunction with Instance Attribution strategies (§ 4.2). As a result, even model creators relying on seemingly open licenses with source attribution requirements may have to implement other strategies in this section.

ModelGo!

the Stack 3

And even when datasets are put under a license, the license might only apply to the collection, not every underlying datapoint. So, for example, if the C4 dataset is released under a ODC-BY license³⁵ it may be that the collection is under this license, but each underlying piece of data is under different license terms.

是的, C4的内容来自于Common Crawl

Furthermore, if web-crawled data is used, restricting it to data that respects robots.txt opt-outs can make a fair use argument more tractable, though not guaranteed. As we noted before, in *Field v. Google, Inc.* (D. Nev. 2006), respect for the robots.txt file was considered in the fair use assessment with the court because it gave the plaintiff opportunity to opt out. This is likely why many webcrawl-based models rely on the CommonCrawl dataset as a source. Its webcrawl automatically respects robots.txt opt-outs and does not crawl every webpage in full. It is possible then that future fair use assessments could consider respecting the robots.txt opt-out—or implementing other opt-out mechanisms—favorably, as was the case in *Field v. Google, Inc.* (D. Nev. 2006). Conversely, ignoring a robots.txt opt-out could negatively impact a fair use assessment. However, Kapoor & Narayanan (2023) have argued that there are structural critiques of opt-out mechanisms beyond the current state of the law.

一种解决方法是仅采用permissive licensed data 训练模型

That being said, the general approach of filtering out any copyrightable content entirely from a dataset (and building foundation models with the remaining data) may not be possible for many practical settings where little to no open-domain or permissively-licensed material is available. It is unclear whether restricting a foundation model to only public domain, non-copyrightable, or otherwise permissively-licensed data could yield a strong foundation model in all domains—though though this is an avenue well worth researching and understanding.³⁶ It may also bias datasets, reducing the efficacy of the model and creating other types of legal and technical problems (Levendowski, 2018). Nonetheless, entities that already retain large and diverse amounts of licensed or public domain data can readily train models using primarily data-filtering strategies. Adobe, for example, recently did so for image generation models.³⁷

适合拥有大量数据版权和公开数据的公司

Data Quality for Less Memorization. Another type of data filtering seeks to remove duplicates from the training set (Nichol, 2022; Kandpal et al., 2022). The goal here is to identify sufficiently similar examples and remove all but one of them. The fewer times a model sees an example during training, the less likely it will memorize it (Lee et al., 2021; Kandpal et al., 2022). This deduplication approach is empirically useful but does not absolutely prevent memorization and regurgitation. Deduplication may also be difficult. For example, if a dataset house thousands of distinct images of a given NBA player with a distinctive tattoo, it may be difficult to deduplicate all of these images in a way that prevents the model from learning to reproduce the tattoo verbatim. Thus, situations like *Alexander v. Take-Two Interactive Software, Inc.* (S.D. Ill. 2020) might still occur with this strategy.

Potential Research Questions. Can foundation models be trained to perform equally well on totally open domain data? Can deduplication schemes take into account high-level semantic similarity in line with fair use without significantly hurting performance? How can we separate fact from expression in training data filtering? For example, Henderson et al. (2022) suggest that a contextual approach is needed to filter input content for privacy and toxicity. Could such a contextual approach provide useful for data filtering in copyright contexts as well?

Output Filtering. Assuming the model is already trained on copyrighted material and were to be deployed, one simple idea for preventing training data from being reproduced is to apply a filter during model inference so that any output that mirrors the training data can be detected. This approach was benchmarked by Copilot’s developers (Ziegler, 2021). Aside from experiencing technical challenges related to increased model inference cost, this approach can be flawed when applied to contexts where a violation of fair use occurs with the non-exact reproduction of copyrighted material. For instance, Ippolito et al. (2022) showed that minimally modified style-transfer prompts can evade filters developed based on the verbatim match criterion. And, though it is unclear whether OpenAI instituted an output filter for our Harry Potter scenario in §2.4, we were able to bypass it with a simple instruction. To capture these sorts of transformations, output filtering techniques will need to go beyond simple surface-level matching.

Based on the case law we discussed, a more fair-use-aligned output filtering approach would focus on detecting transformations unlikely to be fair use, such as direct translations and abridgements. It would ideally also take into account situations where reproduction of content is permitted, including parodies, or factual content.

Other structural factors, such as the nature of the original training task versus the target task, could help also reduce potential risks. For example, building a model that predicts sentiment from a corpus of books is likely to be trans-

³⁵<https://huggingface.co/datasets/c4>

³⁶In particular, a swath of content from the early 1920s is rapidly entering the public domain, increasing the amount of training data available to use without restriction. But this data also is likely to bear undesirable features from the era, including racist or misogynistic content.

³⁷<https://www.theverge.com/2023/3/21/23648315/adobe-firefly-ai-image-generator-announced>

formative, and outputting the answer to a math question, without an explanation, would likely be accepted as factual content. However, if the model goes beyond the simple mathematical answer and outputs a verbatim explanation from a textbook, then it might be more problematic in some cases. So, restricting the structure of a model’s outputs to these sorts of short, factual outputs can be one potential strategy.

There is an exciting new research agenda that would build an output filter which captures some notions of transformativeness under the fair use doctrine. Using such an output filtering mechanism, generation would be biased toward more unique and transformative content, likely to significantly lower – but not eliminate – the risk of infringement liability. Developing such an output filter can be challenging due to the (near) amorphous nature of fair use standards (Burk, 2019), but filtering need not capture the fair use standard perfectly. Instead, filtering should simply reduce the risk of infringement liability. As such, we believe this is an interesting research direction and there is a tractable path toward risk reduction.

Potential Research Questions. How can we develop new high-level semantic similarity measures that capture some aspects of transformativeness for output filtering? How can we separate fact from expression in output filtering? How can we identify parodied content? How can we make robust output filters that prevent users from bypassing them? How can we make output filters that are robust to user manipulations? How can we use output filtering in a way that doesn’t induce model biases?

4.2 Instance Attribution

Instance attribution refers to methods that assign attribution scores to training examples to understand the contribution of individual examples (or group of examples) to (test-time) model predictions (Koh & Liang, 2017; Ghorbani & Zou, 2019; Jia et al., 2019; Pezeshkpour et al., 2021; Ilyas et al., 2022). These approaches tend to adopt techniques such as leave-one-out retraining or influence functions to understand model behavior, fix mislabeled examples, and debug model errors (Koh & Liang, 2017).³⁸

One application of instance attribution is in determining the source of a generated output. The attribution scores can provide information on whether the output was influenced by a particular copyrighted text (or texts). Accurate attribution scores can then be used as a measure for evaluating the copyright infringement risk associated with the output, and to implement an output filter that prevents any output that heavily relies on a single source.

Instance attribution can also address the credit assignment problem by providing a clear attribution page that lists all works which contributed to the output, along with licensing information, to comply with creative commons license attribution guidelines. This might help mitigate DMCA §1202-type claims. In an idealized setting, one can imagine a scenario where every output created an attribution page that enumerated any work that contributed a non-negligible amount to the output, along with licensing information. And in other cases, one might seek to have a post-hoc mechanism to delete information about a particular training example from a model Bourtole et al. (2021)—such as if a DMCA request for takedown is provided.

While promising, current techniques in instance attribution tend to suffer from difficulties in scaling due to high computational cost (e.g., leave-k-out retraining can be costly) (Feldman & Zhang, 2020; Zhang et al., 2021) or being inaccurate or erroneous when applied to complex but realistic model classes (Basu et al., 2020; Ghorbani et al., 2019; Søgaard et al., 2021).

It’s worth noting that retrieval-augmented methods Guu et al. (2018, 2020), which perform attribution fundamentally and not post-hoc, are another approach to instance attribution. These models have the potential to overcome some of the limitations of post-hoc instance attribution methods, and they may also offer other advantages, making them a promising direction for future research.

Potential Research Questions. How can we use instance attribution to identify which *exact* training data points contributed to any given output? How can we ensure that no single datapoint contributes more than a *de minimis* amount to any given output? How can we make instance attributions scalable for runtime attribution?

³⁸ As Feldman & Zhang (2020, at 5-6) note, instance attribution mechanisms could be related to Shapley values—though Shapley values are typically used for attribution of model outputs to input features, e.g., Sundararajan & Najmi (2020).

4.3 Differentially Private Training

Differential privacy (DP) is a formal privacy guarantee that has been adopted in the U.S. Census and big tech (e.g., Smart Compose, telemetry collection) (Miraglia, 2019; Erlingsson et al., 2014; Ding et al., 2017; Bittau et al., 2017; Cheu et al., 2019). In the machine learning context, the guarantee says that no adversary can distinguish, with high probability, between a model trained with a particular training example and one trained without (Dwork et al., 2014). In other words, **model parameters do not vary substantially with the inclusion or exclusion of individual instances**. Machine learning researchers have theoretically and empirically shown that models trained with strong levels of DP guarantee are limited in memorizing training data, and extracting or reconstructing training data from DP-trained models can be close to infeasible (Guo et al., 2022; Carlini et al., 2019). Hence, machine learning with DP guarantees appears to be a natural option for building useful data-driven applications with low copyright-related legal risks.

However, there are three main challenges with operationalizing DP to ensure fair use. First, machine learning with DP has often been reported to suffer from **high computational costs** (Carlini et al., 2019). Recent works have developed substantial improvements to address this drawback through the use of better software primitives (Anil et al., 2021) and training techniques (Li et al., 2021; Yu et al., 2021; De et al., 2022; Sander et al., 2022).

Second, selecting appropriate *privacy leakage parameters* is difficult. DP guarantees are usually stated with desired privacy leakage parameters (e.g., the ϵ parameter in pure-DP (Dwork et al., 2014)) that are set by hand in practice. These parameters introduce an inherent **privacy-utility trade-off** in which the smaller the parameters, the more the privacy (less memorization and regurgitation) and worse the model performance. Setting these parameters can therefore be tricky given that ideal target values tend to be application- and domain-dependent, and that downstream consequences of different choices are difficult to measure and interpret. While there is flourishing research on the topic (Lee & Clifton, 2011), **none has studied this with the goal of leveraging DP to mitigate copyright-related risks**.

Third, it is difficult to define what constitutes a single example that should not be memorized.³⁹ Intuitively stated, DP treats each example in a dataset as a secret. **If a certain secret appears frequently enough, a DP algorithm can still reveal it** (since to the algorithm, this frequently occurring secret is a piece of common knowledge). Therefore, when applied to address copyright issues, the division of the dataset into individual instances needs to be taken with great care in order for the guarantee to be meaningful from the copyright standpoint. Below, we outline hypothetical scenarios where DP algorithms don’t give the desired mitigation effects.

Hypothetical 4.1: Differentially Private Lyric Generation.

Imagine that a developer intends to train a machine learning model to aid musicians to create lyrics. The developer scrapes copyrighted lyrics of songs from music websites. However, the lyrics of the same song are scraped multiple times, each of which is treated as a single example in the dataset. Additionally, the developer isn’t careful about **removing duplicates before training the model with DP**. The final model thus ends up reproducing verbatim chunks of lyrics of certain songs. The lyricist whose lyrics were reproduced by the deployed model sues an end user who wrote a song with the help of this model.

Hypothetical 4.2: Differential Privacy and Trademarks.

Imagine a text-to-image model was trained with **lots of images that have the same trademark** (e.g., the trademark is positioned in similar locations on each image and likely to be memorized). Since there is a strong correlation between examples in the training set, the image-level DP guarantee does not prevent the model from generating images that contain the blob of trademark symbol or text. This was one real-world challenge that was cited for DALL·E’s filtering technique, noting that it can create real trademarks and logos (Mishkin et al., 2022). And recently, litigation by Getty Images explicitly cited trademark infringement due to its watermark being regurgitated in generated images (*Getty Images (US), Inc. v. Stability AI, Inc.*, D. Del. 2023).

The above examples highlight that to leverage DP in a meaningful way, one needs to ensure that the division of data is handled at a semantic level that is meaningful in fair use standards. Finding out the “right” semantic level is an interesting topic of future research. In addition, exact or fuzzy data de-duplication based on the target semantic level is likely useful to attain the ideal benefit of the DP guarantee (Lee et al., 2021; Kandpal et al., 2022).

Recently, Vyas et al. (2023) introduced *near access-freeness* (NAF) as a mathematical guarantee of copyright protection, along with a few practical algorithms for attaining the guarantee. The NAF guarantee is similar in spirit to the DP

³⁹The issue has been extensively studied in the privacy literature. See, e.g., Kifer & Machanavajjhala (2011) for examples in social networks.

guarantee (both leverage indistinguishability as the core concept), but is different in their precise semantics as well as the algorithmic primitives. In broad strokes, the NAF guarantee is attained for a model trained on copyrighted material, **if the model generates in a manner similar to a model trained without that material**. Technically, to achieve the guarantee, the proposed algorithms require that a single copyrighted material “appear” in at most a single (or a constant many) training example(s) in the original dataset. Applying a pure surface-level data deduplication scheme is insufficient to attain the above prerequisite, and better deduplication schemes based on higher-level understandings of similarity are likely required. While this NAF guarantee, like other approaches, is not a panacea and requires more research to align with fair use, it is another powerful tool worth pursuing and tailoring to fair use standards.

Potential Research Questions. How can we identify higher-level similarity features to leverage differential privacy or NAF in a way that is in line with fair use? What are privacy budgets would be acceptable under fair use doctrine that would prevent significant degradations in performance?

4.4 Learning from Human Feedback

Learning from human feedback (Ouyang et al., 2022) trains models to generate outputs that are aligned with human preferences and values. However, these approaches—and similar ones aimed at promoting helpfulness (Wei et al., 2021; Sanh et al., 2021)—should also consider the copyright risk. **Human feedback might reward verbatim generations of copyrighted content**. For example, if a model is rated purely by how well it follows instructions, the highest reward for “Read me a Harry Potter book verbatim” would be to read the entire book verbatim, which could infringe on the source material’s distribution rights.

To address this issue, human annotation frameworks in these approaches can take into account the copyright implications of rating systems and instruction following, particularly when incorporating human feedback at scale. For example, in current feedback-based learning mechanisms, human labelers are asked to rate model generations based on a Likert scale or pairwise comparisons. A method for learning a reward function that both maximizes the capability of the model and respects fair use could add an additional question, where human labelers would be provided with the closest **copyrighted content and asked to flag any content that is not sufficiently transformative from the copyrighted material**. Models can then be trained with this feedback incorporated.

This approach could be viewed as an **extension of existing approaches to reducing the harmfulness of models** (Bai et al., 2022a,b). This approach provides no certifiable guarantee and it could be susceptible to reward misspecification. Nonetheless, it may be a useful component in reducing copyright violations, as it leverages existing mechanisms and ongoing research for value alignment.

As models improve in their capabilities, taking into account longer contexts and following instructions more closely, it might become easier to regurgitate non-transformative material. Asking a code-generating model in the coming years to “Implement a chess playing app” might copy the GPL-licensed Stockfish app in its entirety, increasing the likelihood of potential risks.⁴⁰ But at the same time, capable models might be better able to understand the idea of transformation and be easier to align from a copyright perspective. This highlights the importance of mitigation strategies like extractive-preventative RLHF that can balance improved capabilities with fair use.

Potential Research Questions. How can we make models that follow instructions but don’t allow users to easily bypass output filters? How can we train advanced models that follow instructions but in totally creative ways transformative from the training data? Is there a way to instill some partial knowledge of fair use so that models can reason about their own outputs can keep them in line with fair use?

5 Forward-looking Agenda

As demonstrated throughout this work, the risk of copyright violation and litigation, even with fair use protection, is a real concern. To mitigate these risks, we recommend that foundation model practitioners consider implementing the mitigation strategies outlined here and pursuing other novel research in this area. There is significant, exciting, technical research required to make technical mitigation strategies robust and aligned with fair use doctrine. We reinforce that

⁴⁰ A scenario based on the real litigation of Stockfish against Chessbase that did not involve generative models, but involved the copying of the Stockfish neural network and surrounding code by Chasebase. See <https://stockfishchess.org/blog/2021/our-lawsuit-against-chessbase/>.

machine learning researchers *must* play a role in providing viable mitigation mechanisms to demonstrate that models are truly covered by fair use.

Preventing extreme outcomes in the evolution of fair use law by advancing mitigation strategies. Legal scholars have noted that there might be **two extreme outcomes** for fair use and machine learning (Sobel, 2017). On one hand, there is a possibility that courts may rule that foundation models are widely acceptable under fair use regardless of the likelihood of infringement or efforts at mitigation, which could have adverse effects on the income of data creators and disregard the ethical and moral rights attached to their work. On the other hand, there is a possibility that courts may declare that generative foundation models cannot be trained on unlicensed copyrighted data in most cases. This scenario could lead to a concentration of power for companies that have retained licenses to large amounts of data; companies like YouTube or Facebook might be able to leverage large amounts of user-contributed data where others would be shut out of model training. Neither of these two outcomes is ideal. **As litigation progresses, identifying mechanisms to prevent extreme outcomes will be critical.** For example, it is important to demonstrate that not all forms of foundation models are inherently infringing and that some of their potential risks can be effectively managed through technical means.

With better demonstrations of co-evolving technical mitigation strategies, **the law might find a middle ground that allows model training and deployment with sufficient effort to implement objectively strong mitigation strategies.** Courts may consider the reasonable efforts of model builders and deployers to mitigate copyright risk, both in deciding fair use and in determining whether they can face indirect infringement liability. Trademark courts have taken a similar approach, for example in *Tiffany v. eBay* (2d Cir. 2010). As such, advancing research in this area (with methods such as improved similarity metrics) may help in preventing extreme outcomes in legal settings.⁴¹

We should not over-zealously filter. There must be a balance to filtering. Well intentioned but strict filtering mandates adopted by other countries have been criticized and criticized for their impacts on free speech (Schmon et al., 2022). Similarly, YouTube’s content ID system, a large-scale filtering approach, has been criticized for not following fair use standards and being **overaggressive in its filtering** (Bartholomew, 2014; Boroughf, 2015). Levendowski (2018) points out that restrictive views of fair use doctrine can exacerbate biases and that fair use can help create fairer systems. While mitigation strategies will help prevent undesirable outcomes, it is important to develop strategies that carefully align with fair use standards, as we have previously discussed. This means that factual content should not necessarily be filtered, neither should parodies, or short form regurgitation used for commentary. And evolutions of fair use doctrine or further policymaking should consider the distributive effects of preventing access to certain types of data for model creation.

Policymakers could consider how and if DMCA (or similar) safe harbors should apply to foundation models. As we have seen, there are various ways, including filtering, to mitigate the risk of copyright infringement in the output of foundation models, but none will entirely eliminate the risk of liability. Even when trained on presumably permissively licensed datasets, for example, **it is difficult (if not impossible) to determine the provenance of every piece of data and filter it out.** Users might post content to seemingly permissively-licensed databases that they do not actually have the rights for. There may even be uncertainty about whether a piece of content is *actually* in the public domain or whether that status has been **revoked**.⁴² And even if foundation model practitioners implement strong mitigation strategies, the amorphous nature of fair use doctrine may make it difficult to know what kinds of content will be covered by fair use *ex ante*.

With the uncertainty of DMCA protections (discussed in § 3), the law may need to adapt to this reality, and it could do so, for instance, by clarifying the role of safe harbors for models that implement sufficiently strong mitigation strategies. Policymakers could make clear that DMCA protections apply to this setting or they could identify other more suitable safe harbor mechanisms. This may provide more balance than general-purpose text and data mining exemptions seen in other countries, but again are not a panacea. Such safe harbors would have to be structured to consider the strength of the implemented mitigation strategies to ensure that they are not abused.

Pursuing other remedies beyond technical mitigation. Importantly, even if technical mitigation strategies managed to keep foundation models within the confines of fair use, these models may still create harms in many other ways—including disrupting creative industries, exploiting labor, and more. See extensive discussion by, *e.g.*, Bender et al.

⁴¹But, again, technical mitigation strategies will only go so far in the fair use assessment and will not (and should not) automatically guarantee that any one deployment is acceptable under fair use doctrine.

⁴²This is not a hypothetical, in *Golan v. Holder* (2012) the Supreme Court found that **revoking a work’s public domain status is not unconstitutional.** In that case, a group of artists had relied on the public domain status of some works whose copyright status was later restored as part of the Uruguay Round Agreements Act.

(2021); Bommasani et al. (2021); Blodgett et al. (2020); Mahelona et al. (2023). It is important to note that we do not suggest that technical mitigation strategies will solve everything and neither will fair use doctrine. Our goal here is to point out that currently there is more work to be done even *within* the confines of fair use to make foundation models more in line with case law. Other strategies to prevent harms should be pursued in conjunction with the strategies we outline here, but they should be carefully weighed against other potential harms from excluding data under overly restrictive copyright standards (Levendowski, 2018). For example, complementary approaches to what we describe here could include statutory licensing schemes, taxation and redistribution, or other policy mechanisms. While these may be worthy of considering, each may have its own challenges and are outside the scope of this work. Furthermore, there are other aspects of fair use that we do not consider here, and there well may be cases where technical mitigation strategies will still not be enough for fair use.

6 Related Work

While we have generally referenced related work throughout this paper, here we briefly highlight several areas of work that we build on and survey. Related work to our own can fall into two categories: (1) examining technically how models regurgitate training data; (2) understanding copyright law as applied to machine learning systems.

Ethical Considerations of Foundation Models. A number of other works have noted the potential risks and harms of using foundation models. Bommasani et al. (2021), Bender et al. (2021), and Henderson et al. (2018) all provide high level overviews of the potential risks from language model or foundation model deployments. Weidinger et al. (2022) taxonomize the risks of language models, noting copyright infringement and effects on creative economies.

Technical Examinations of Regurgitation. Several works have demonstrated how various factors affect generation of memorized content (Carlini et al., 2019; Lee et al., 2022; Carlini et al., 2022; Kandpal et al., 2022; Carlini et al., 2021; Yu et al., 2023). These works have consistently found that generative models memorize or plagiarize content. The percentage of verbatim outputs varies depending on extraction strategy and the model, but varies from 0.007% (Kandpal et al., 2022) to 4.85% (Lee et al., 2022) (variation comes from methodology of sampling and similarity metric).

Legal work examining copyright and Artificial Intelligence. On the legal side, a large body of work has covered potential legal risks and challenges of machine learning (Sobel, 2017; Burk, 2019; Lemley & Casey, 2019; Gilotte, 2019; Lemley & Casey, 2020; Franceschelli & Musolesi, 2022; Guadamuz, 2017; Grimmelmann, 2015; McCann, 2021; McJohn & McJohn, 2020; Levendowski, 2018; Samuelson & Asay, 2017; Lim, 2022; Samuelson, 2021). Many of these note how fair use law might apply in different ways to machine learning models and how outcomes are uncertain.

Levendowski (2018) points out that more expansive notions of copyright law could help with challenges of bias and equity by allowing the inclusion of more data into models. This is countered by Mahelona et al. (2023), Reed (2021), and others who have pointed out that data can be used from marginalized communities without their say by leveraging fair use law. This could take away their voice in their data’s governance.

Others have examined how machine learning or algorithms can be used for mitigating infringement risk at a high level, including Elkin-Koren (2017); Scheffler et al. (2022). But others have pointed out that such filtering strategies can have harmful effects (Bartholomew, 2014; Boroughf, 2015; Lim, 2022; Levendowski, 2018).

Tang (2022, 2021) discusses the challenges (and benefits) of bringing class action litigation against new technologies not unlike foundation models. They describe how class action lawsuits can act as a licensing mechanism at scale when it is nearly impossible to aggregate licenses from many singleton data creators.

Unlike many of these other works, we marry the doctrinal discussion of fair use to technical mitigation strategies. We provide a short primer on fair use doctrine as it applies to foundation models before highlighting potential deficiencies in current risk mitigation strategies that have been employed. This acts as a survey of some similar discussions in prior work but also expands it with experiments and concrete examples of foundation model uses. Our aim is to speak to both machine learning researchers and legal professions to point out the exciting *technical* research agenda that would make foundation models more in line with fair use as well as policy-relevant considerations for the evolution of the law.

Alignment. A significant amount of recent work has focused on the AI alignment problem, broadly defined, where researchers have sought to align foundation model outputs with societal values. Some of the technical mitigation strategies we propose here can be related to this line of work. This includes, for example, making FMs more aligned with human preferences and more likely to follow instructions (Christiano et al., 2017; Ziegler et al., 2019; Ouyang et al., 2022; Wei et al., 2021; Sanh et al., 2021). Hendrycks et al. (2021) provide a survey of unsolved challenges in AI

Safety, including alignment. Broadly, our proposal can be thought of as contributing to the better alignment between Artificial Intelligence on one hand, and law & policy requirements on the other.

Data Governance and Curation. The recent literature on data governance and curation discusses fair use in machine learning (Jernite et al., 2022; Paullada et al., 2021; Ganguli et al., 2022). For instance, Jernite et al. (2022) weigh in the stakes of data creators and examine their *property rights* when developing the data governance framework. Paullada et al. (2021) survey legal issues with benchmark datasets and comment on the nuances and the novelty of rising problems involving large-scale machine learning and copyright law. Our work is related to these prior works but goes deeper into the legal nuances with concrete case studies and state-of-the-art model artifacts obtained from real experiments.

7 Conclusion

We reviewed U.S. fair use standards and analyzed the risks of foundation models when evaluated against those standards in a number of concrete scenarios with real model artifacts. Additionally, we also discussed mitigation strategies and their respective strengths and limitations. As the law is murky and evolving, our goal is to delineate the legal landscape and present an exciting research agenda that will improve model quality overall, further our understanding of foundation models, and help make models more in line with fair use doctrine. By pursuing mitigation strategies that can respect the ethics and legal standards of intellectual property law, machine learning researchers can help shape the law going forward. But we emphasize that even if fair use is met to the fullest, the impacts to some data creators will be large. We suggest that further work is needed to identify policies that can effectively manage and mitigate these impacts, where the technical mitigation strategies we propose here will fundamentally fall short. We hope that this guide will be useful to machine learning researchers and practitioners, as well as lawyers, judges, and policymakers thinking about these issues. We emphasize, again, that even if foundation models are covered by fair use the impacts on labor might be con

Acknowledgements

This work was done at the Center for Research on Foundation Models (CRFM), and we would also like to thank the Stanford Institute for Human-Centered Artificial Intelligence (HAI) for supporting this work. We thank Alex Aiken for generously providing us with access to MossPlus—the commercial version of Moss. We thank Rishi Bommasani, Dilip Arumugam, Mark Krass, and Jieru Hu for helpful discussions and feedback. We thank Tony Lee for supporting our experiments with the CRFM infrastructure. PH is funded by the OpenPhilanthropy AI Fellowship. XL is supported by a Stanford Graduate Fellowship. TH and DJ was supported by a grant from OpenPhilanthropy. Note, ML was hired as counsel for *Andersen et al. v. Stability AI et al.* (N.D. Cal. 2023) after a near-final draft of this work was written. This work reflects the personal opinions and research of the authors. It does not reflect the position of any other entity or person, nor does it constitute legal advice.

References

- Rohan Anil, Badhi Ghazi, Vineet Gupta, Ravi Kumar, and Pasin Manurangsi. Large-scale differentially private bert. *arXiv preprint arXiv:2108.01624*, 2021.
- Clark D Asay. Transformative use in software. *Stan. L. Rev. Online*, 70:9, 2017.
- Clark D Asay, Arielle Sloan, and Dean Sobczak. Is transformative use eating the world. *BCL Rev.*, 61:905, 2020.
- Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022a.
- Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*, 2022b.
- Andy Baio. Ai data laundering: How academic and nonprofit researchers shield tech companies from accountability. <https://waxy.org/2022/09/ai-data-laundering-how-academic-and-nonprofit-researchers-shield-tech-companies-from-accountability/>, 2022.
- Stephanie Plamondon Bair. Rational faith: The utility of fairness in copyright. *BUL Rev.*, 97:1487, 2017.
- John Bandy and Nicholas Vincent. Addressing "documentation debt" in machine learning: A retrospective datasheet for bookcorpus. In J. Vanschoren and S. Yeung (eds.), *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021. URL <https://datasets-benchmarks-proceedings.neurips.cc/paper/2021/file/54229abfcfa5649e7003b83dd4755294-Paper-round1.pdf>.

- Taylor B Bartholomew. The death of fair use in cyberspace: Youtube and the problem with content id. *Duke L. & Tech. Rev.*, 13:66, 2014.
- Samyadeep Basu, Philip Pope, and Soheil Feizi. Influence functions in deep learning are fragile. *arXiv preprint arXiv:2006.14651*, 2020.
- Barton Beebe. An empirical study of us copyright fair use opinions updated, 1978-2019. *NYU J. Intell. Prop. & Ent. L.*, 10:1, 2020.
- Emily M Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pp. 610–623, 2021.
- Stella Biderman, Kieran Bicheno, and Leo Gao. Datasheet for the pile. *arXiv preprint arXiv:2201.07311*, 2022.
- Andrea Bittau, Úlfar Erlingsson, Petros Maniatis, Ilya Mironov, Ananth Raghunathan, David Lie, Mitch Rudominer, Ushasree Kode, Julien Tinnes, and Bernhard Seefeld. Prochlo: Strong privacy for analytics in the crowd. In *Proceedings of the 26th symposium on operating systems principles*, pp. 441–459, 2017.
- Joshua Bloch and Pamela Samuelson. Some misconceptions about software in the copyright literature. In *CSLAW’22: Proceedings of the 2nd ACM Symposium on Computer Science and Law*, 2022.
- Su Lin Blodgett, Solon Barocas, Hal Daumé III, and Hanna Wallach. Language (technology) is power: A critical survey of “bias” in nlp. *arXiv preprint arXiv:2005.14050*, 2020.
- Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.
- Benjamin Boroughf. The next great youtube: improving content id to foster creativity, cooperation, and fair compensation. *Alb. LJ Sci. & Tech.*, 25:95, 2015.
- Lucas Bourtole, Varun Chandrasekaran, Christopher A Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. Machine unlearning. In *2021 IEEE Symposium on Security and Privacy (SP)*, pp. 141–159. IEEE, 2021.
- Peter Brown and Bob Mercer. Twenty years of bitext. <https://www.cs.jhu.edu/~post/bitext/>, 2013.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.
- Michelle Brownlee. Safeguarding style: What protection is afforded to visual artists by the copyright and trademark laws. *Colum. L. Rev.*, 93:1157, 1993.
- Dan L Burk. Algorithmic fair use. *U. Chi. L. Rev.*, 86:283, 2019.
- Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. The secret sharer: Evaluating and testing unintended memorization in neural networks. In *28th USENIX Security Symposium (USENIX Security 19)*, pp. 267–284, 2019.
- Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Úlfar Erlingsson, et al. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*, pp. 2633–2650, 2021.
- Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang. Quantifying memorization across neural language models. *arXiv preprint arXiv:2202.07646*, 2022.
- Nicholas Carlini, Jamie Hayes, Milad Nasr, Matthew Jagielski, Vikash Sehwal, Florian Tramèr, Borja Balle, Daphne Ippolito, and Eric Wallace. Extracting training data from diffusion models, 2023. URL <https://arxiv.org/abs/2301.13188>.
- Michael W Carroll. Copyright and the progress of science: Why text and data mining is lawful. *UC Davis L. Rev.*, 53: 893, 2019.
- Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.
- Albert Cheu, Adam Smith, Jonathan Ullman, David Zeber, and Maxim Zhilyaev. Distributed differential privacy via shuffling. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 375–403. Springer, 2019.

- Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*, 2022.
- Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30, 2017.
- Dane S Ciolino. Rethinking the compatibility of moral rights and fair use. *Wash. & Lee L. Rev.*, 54:33, 1997.
- Samuel J Coe. The story of a character: Establishing the limits of independent copyright protection for literary characters. *Chi.-Kent L. Rev.*, 86:1305, 2011.
- Soham De, Leonard Berrada, Jamie Hayes, Samuel L Smith, and Borja Balle. Unlocking high-accuracy differentially private image classification through scale. *arXiv preprint arXiv:2204.13650*, 2022.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.
- Bolin Ding, Janardhan Kulkarni, and Sergey Yekhanin. Collecting telemetry data privately. *Advances in Neural Information Processing Systems*, 30, 2017.
- F Jay Dougherty. All the world’s not a stooge: The transformativeness test for analyzing a first amendment defense to a right of publicity claim against distribution of a work of art. *Colum. JL & Arts*, 27:1, 2003.
- Dr. Seuss Enters., L.P. v. ComicMix LLC*. 983 F.3d 443, 9th Cir. 2020. URL <https://www.copyright.gov/fair-use/summaries/drseuss-comicmix-9thcir2020.pdf>.
- Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014.
- Niva Elkin-Koren. Fair use by design. *UCLA L. Rev.*, 64:1082, 2017.
- Niva Elkin-Koren and Neil Weinstock Netanel. Transplanting fair use across the globe: A case study testing the credibility of us opposition. *Hastings LJ*, 72:1121, 2020.
- Alexander v. Take-Two Interactive Software, Inc.* 489 F. Supp. 3d 812, S.D. Ill. 2020.
- Andersen et al. v. Stability AI et al.* 3:23-cv-00201, N.D. Cal. 2023.
- Andy Warhol Foundation v. Goldsmith*. 2d Cir., 2021.
- Authors Guild, Inc. v. Google, Inc.* 721 F.3d 132, 2d Cir. 2015.
- DOE I v. GitHub, Inc.* 4:22-cv-06823, N.D. Cal. 2022.
- Dr. Seuss Enters., LP v. Penguin Books USA, Inc.* 109 F.3d 1394, 9th Cir. 1997.
- Fox News Network, LLC v. TVEyes, Inc.* Nos. 15-3885, 15-3886, 2d Cir. Feb. 27, 2018.
- Getty Images (US), Inc. v. Stability AI, Inc.* 1:99-mc-09999, D. Del. 2023.
- Google LLC v. Oracle America Inc.* 141 S. Ct. 1183, 593 U.S., 209 L. Ed. 2d 311, 2021.
- Hall v. Swift*. No. 18-55426, 9th Cir., Oct. 28, 2019.
- Nihon Keizai Shimbun, Inc. v. Comline Bus. Data Inc.* 166 F.3d 65, 69, 2d Cir. 1999.
- Paramount Pictures Corp. v. Axanar Prods., Inc.* No. 2:15-cv-09938-RGK-E, C.D. Cal. Jan. 3, 2017.
- Penguin Grp. (USA), Inc. v. Am. Buddha*. No. 4:13-cv-02075-JGZ, D. Ariz. May 11, 2015.
- Penguin Random House LLC, et al. v. Frederik Colting and Melissa Medina, d/b/a Moppet Books*. No. 17-cv-386, S.D.N.Y. Sept. 8, 2017.
- Warner Bros. Entertainment Inc. v. RDR Books*. 575 F. Supp. 2d 513, S.D.N.Y. 2008.
- Úlfar Erlingsson, Vasyl Pihur, and Aleksandra Korolova. Rappor: Randomized aggregatable privacy-preserving ordinal response. In *Proceedings of the 2014 ACM SIGSAC conference on computer and communications security*, pp. 1054–1067, 2014.
- European Union. Directive (EU) 2019/790 of the European Parliament and of the Council of 17 April 2019 on copyright and related rights in the Digital Single Market and amending Directives 96/9/EC and 2001/29/EC, 2019.
- Vitaly Feldman and Chiyuan Zhang. What neural networks memorize and why: Discovering the long tail via influence estimation. *Advances in Neural Information Processing Systems*, 33:2881–2891, 2020.
- Carlos Muñoz Ferrandis, Danish Contractor, Huu Nguyen, and David Lansky. The BigScience RAIL License. <https://bigscience.huggingface.co/blog/the-bigscience-rail-license>, 2022.

- Giorgio Franceschelli and Mirco Musolesi. Copyright in generative deep learning. *Data & Policy*, 4, 2022.
- Iason Gabriel. Artificial intelligence, values, and alignment. *Minds and machines*, 30(3):411–437, 2020.
- Deep Ganguli, Danny Hernandez, Liane Lovitt, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova Dassarma, Dawn Drain, Nelson Elhage, et al. Predictability and surprise in large generative models. In *2022 ACM Conference on Fairness, Accountability, and Transparency*, pp. 1747–1764, 2022.
- Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*, 2020.
- Amirata Ghorbani and James Zou. Data shapley: Equitable valuation of data for machine learning. In *International Conference on Machine Learning*, pp. 2242–2251. PMLR, 2019.
- Amirata Ghorbani, Abubakar Abid, and James Zou. Interpretation of neural networks is fragile. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pp. 3681–3688, 2019.
- Jessica L Gillotte. Copyright infringement in AI-generated artworks. *UC Davis L. Rev.*, 53:2655, 2019.
- Aaron Gokaslan and Vanya Cohen. Openwebtext corpus, 2019.
- James Grimmelmann. Copyright for literate robots. *Iowa L. Rev.*, 101:657, 2015.
- Andres Guadamuz. Do androids dream of electric copyright? comparative analysis of originality in artificial intelligence generated works. *Intellectual property quarterly*, 2017.
- Chuan Guo, Brian Karrer, Kamalika Chaudhuri, and Laurens van der Maaten. Bounding training data reconstruction in private (deep) learning. *arXiv preprint arXiv:2201.12383*, 2022.
- Kelvin Guu, Tatsunori B Hashimoto, Yonatan Oren, and Percy Liang. Generating sentences by editing prototypes. *Transactions of the Association for Computational Linguistics*, 6:437–450, 2018.
- Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. Retrieval augmented language model pre-training. In *International conference on machine learning*, pp. 3929–3938. PMLR, 2020.
- Peter Henderson, Koustuv Sinha, Nicolas Angelard-Gontier, Nan Rosemary Ke, Genevieve Fried, Ryan Lowe, and Joelle Pineau. Ethical challenges in data-driven dialogue systems. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, pp. 123–129, 2018.
- Peter Henderson, Mark S Krass, Lucia Zheng, Neel Guha, Christopher D Manning, Dan Jurafsky, and Daniel E Ho. Pile of law: Learning responsible data filtering from the law and a 256gb open-source legal dataset. *arXiv preprint arXiv:2207.00220*, 2022.
- Dan Hendrycks, Nicholas Carlini, John Schulman, and Jacob Steinhardt. Unsolved problems in ml safety. *arXiv preprint arXiv:2109.13916*, 2021.
- Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. An empirical analysis of compute-optimal large language model training. In *Advances in Neural Information Processing Systems*.
- Andrew Ilyas, Sung Min Park, Logan Engstrom, Guillaume Leclerc, and Aleksander Madry. Datamodels: Predicting predictions from training data. *arXiv preprint arXiv:2202.00622*, 2022.
- Daphne Ippolito, Florian Tramèr, Milad Nasr, Chiyuan Zhang, Matthew Jagielski, Katherine Lee, Christopher A. Choquette-Choo, and Nicholas Carlini. Preventing verbatim memorization in language models gives a false sense of privacy, 2022. URL <https://arxiv.org/abs/2210.17546>.
- Russell W Jacobs. Gutters and hyperlinks: The dmca and proper position of copyright management information. *Nw. J. Tech. & Intell. Prop.*, 11:xxi, 2012.
- Steven D Jamar and Christen Glenn. When the author owns the world: Copyright issues arising from monetizing fan fiction. *Tex. A&M L. Rev.*, 1:959, 2013.
- Japan. 2018 amendment to the japanese copyright act, 2018.
- Yacine Jernite, Huu Nguyen, Stella Biderman, Anna Rogers, Maraim Masoud, Valentin Danchev, Samson Tan, Alexandra Sasha Luccioni, Nishant Subramani, Isaac Johnson, et al. Data governance in the age of large-scale data-driven language technology. In *2022 ACM Conference on Fairness, Accountability, and Transparency*, pp. 2206–2222, 2022.
- Ruoxi Jia, David Dao, Boxin Wang, Frances Ann Hubis, Nick Hynes, Nezihe Merve Gürel, Bo Li, Ce Zhang, Dawn Song, and Costas J Spanos. Towards efficient data valuation based on the shapley value. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 1167–1176. PMLR, 2019.

- Nikhil Kandpal, Eric Wallace, and Colin Raffel. Deduplicating training data mitigates privacy risks in language models. *arXiv preprint arXiv:2202.06539*, 2022.
- Sayash Kapoor and Arvind Narayanan. Artists can now opt out of generative ai. it’s not enough. <https://aisnakeo.il.substack.com/p/artists-can-now-opt-out-of-generative>, 2023.
- Daniel Kifer and Ashwin Machanavajjhala. No free lunch in data privacy. In *Proceedings of the 2011 ACM SIGMOD International Conference on Management of data*, pp. 193–204, 2011.
- Denis Kocetkov, Raymond Li, Loubna Ben Allal, Jia Li, Chenghao Mou, Carlos Muñoz Ferrandis, Yacine Jernite, Margaret Mitchell, Sean Hughes, Thomas Wolf, Dzmitry Bahdanau, Leandro von Werra, and Harm de Vries. The stack: 3 tb of permissively licensed source code. *Preprint*, 2022.
- Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *International conference on machine learning*, pp. 1885–1894. PMLR, 2017.
- Jaewoo Lee and Chris Clifton. How much is enough? choosing ϵ for differential privacy. In *International Conference on Information Security*, pp. 325–340. Springer, 2011.
- Jooyoung Lee, Thai Le, Jinghui Chen, and Dongwon Lee. Do language models plagiarize? *arXiv preprint arXiv:2203.07618*, 2022.
- Katherine Lee, Daphne Ippolito, Andrew Nystrom, Chiyuan Zhang, Douglas Eck, Chris Callison-Burch, and Nicholas Carlini. Deduplicating training data makes language models better. *arXiv preprint arXiv:2107.06499*, 2021.
- Kalev Leetaru. Common crawl and unlocking web archives for research. <https://www.forbes.com/sites/kalevleetaru/2017/09/28/common-crawl-and-unlocking-web-archives-for-research/?sh=7b9bbae63b83>, 2017.
- Mark A Lemley and Bryan Casey. Remedies for robots. *The University of Chicago Law Review*, 86(5):1311–1396, 2019.
- Mark A Lemley and Bryan Casey. **Fair learning**. *Tex. L. Rev.*, 99:743, 2020.
- Amanda Levendowski. How copyright law can fix artificial intelligence’s implicit bias problem. *Wash. L. Rev.*, 93:579, 2018.
- Xuechen Li, Florian Tramer, Percy Liang, and Tatsunori Hashimoto. Large language models can be strong differentially private learners. *arXiv preprint arXiv:2110.05679*, 2021.
- Yujia Li, David Choi, Junyoung Chung, Nate Kushman, Julian Schrittwieser, Rémi Leblond, Tom Eccles, James Keeling, Felix Gimeno, Agustin Dal Lago, et al. Competition-level code generation with alphacode. *arXiv preprint arXiv:2203.07814*, 2022.
- Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, et al. Holistic evaluation of language models. *arXiv preprint arXiv:2211.09110*, 2022.
- Daryl Lim. Ai, equity, and the ip gap. *SMU Law Review*, 75(4):815, 2022.
- Susuk Lim. A survey of the dcma’s copyright management information protections: The dmca’s cmi landscape after all headline news and mcclatchey. *Wash. JL Tech. & Arts*, 6:297, 2010.
- Jacqueline D Lipton. Copyright and the commercialization of fanfiction. *Hous. L. Rev.*, 52:425, 2014.
- Keoni Mahelona, Gianna Leoni, Suzanne Duncan, and Miles Thompson. Openai’s whisper is another case study in colonisation. *Papa Reo*, 2023. URL <https://blog.papareo.nz/whisper-is-another-case-study-in-colonisation/>.
- Mango v. BuzzFeed, Inc. 970 F.3d 167, 2020.
- Tony Mason, Ada Gavrilovska, and David A Joyner. Collaboration versus cheating: Reducing code plagiarism in an online ms computer science program. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*, pp. 1004–1010, 2019.
- Mavrix Photographs, LLC v. Livejournal Inc.* 873 F.3d 1045, 9th Cir. 2017.
- Sancho McCann. Copyright throughout a creative ai pipeline. *Canadian JL & Tech*, 2021.
- Stephen McJohn and Ian McJohn. Fair use and machine learning. *NEULR*, 12:99, 2020.
- Eric Miraglia. Privacy that works for everyone. 2019. URL <https://blog.google/technology/safety-security/privacy-everyone-io/>.
- Pamela Mishkin, Lama Ahmad, Miles Brundage, Gretchen Krueger, and Girish Sastry. Dall-e 2 preview - risks and limitations. 2022.

- Gary Myers. Muddy waters: Fair use implications of google llc v. oracle america, inc. *Nw. J. Tech. & Intell. Prop.*, 19: 155, 2021.
- Alex Nichol. Dalle 2 pre-training mitigations. <https://openai.com/blog/dall-e-2-pre-training-mitigations/>, 2022.
- Tyler T Ochoa. Dr. seuss, the juice and fair use revisited: Two decades of parody and satire in copyright law. *IDEA*, 59: 233, 2018.
- OpenAI. Gpt-4 technical report, 2023. URL <https://arxiv.org/abs/2303.08774>.
- Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *arXiv preprint arXiv:2203.02155*, 2022.
- Amandalynne Paullada, Inioluwa Deborah Raji, Emily M Bender, Emily Denton, and Alex Hanna. Data and its (dis) contents: A survey of dataset development and use in machine learning research. *Patterns*, 2(11):100336, 2021.
- Shira Perlmuter. Copyright and state sovereign immunity: A report of the register of copyrights. 2021.
- Pouya Pezeshkpour, Sarthak Jain, Byron C Wallace, and Sameer Singh. An empirical comparison of instance attribution methods for nlp. *arXiv preprint arXiv:2104.04128*, 2021.
- Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022.
- Shawn Presser. Books3. <https://twitter.com/theshawwn/status/1320282149329784833>, 2020.
- Public Resource. The General Index. <https://archive.org/details/GeneralIndex>, 2021.
- Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pp. 8748–8763. PMLR, 2021.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv e-prints*, 2019.
- Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pp. 8821–8831. PMLR, 2021.
- In re Aimster Copyright Litig. 334 F.3d 643, 645-646, 7th Cir. 2003.
- Trevor G Reed. Fair use as cultural appropriation. *Cal. L. Rev.*, 109:1373, 2021.
- Jerome H Reichman and Paul F Uhler. Database protection at the crossroads: recent development and their impact on science and technology. *Berkeley Tech. LJ*, 14:793, 1999.
- Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2021.
- Matthew Sag. The new legal landscape for text mining and machine learning. *J. Copyright Soc’y USA*, 66:291, 2018.
- Pamela Samuelson. Text and data mining of in-copyright works: is it legal? *Communications of the ACM*, 64(11): 20–22, 2021.
- Pamela Samuelson and Clark D Asay. Saving software’s fair use future. *Harv. JL & Tech.*, 31:535, 2017.
- Tom Sander, Pierre Stock, and Alexandre Sablayrolles. Tan without a burn: Scaling laws of dp-sgd. *arXiv preprint arXiv:2210.03403*, 2022.
- Victor Sanh, Albert Webson, Colin Raffel, Stephen H Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, et al. Multitask prompted training enables zero-shot task generalization. *arXiv preprint arXiv:2110.08207*, 2021.
- Sarah Scheffler, Eran Tromer, and Mayank Varia. Formalizing human ingenuity: A quantitative framework for coyright law’s substantial similarity. *arXiv preprint arXiv:2206.01230*, 2022.
- Saul Schleimer, Daniel S Wilkerson, and Alex Aiken. Winnowing: local algorithms for document fingerprinting. In *Proceedings of the 2003 ACM SIGMOD international conference on Management of data*, pp. 76–85, 2003.
- Christoph Schmon, Filip Lukáš, and Corynne McSherry. The eu’s copyright directive is still about filters, but eu’s top court limits its use. <https://www.eff.org/deeplinks/2022/05/eus-copyright-directive-still-about-filters-eus-top-court-limits-its-use>, 2022.
- Christoph Schuhmann. Laion-400-million open dataset, 2021.

- John Schulman, Barret Zoph, Christina Kim, Jacob Hilton, Jacob Menick, Jiayi Weng, Juan Felipe Ceron Uribe, Liam Fedus, Luke Metz, Michael Pokorny, Rapha Gontijo Lopes, Shengjia Zhao, Arun Vijayvergiya, Eric Sigler, Adam Perelman, Chelsea Voss, Mike Heaton, Joel Parish, Dave Cummings, Rajeev Nayak, Valerie Balcom, David Schnurr, Tomer Kaftan, Chris Hallacy, Nicholas Turley, Noah Deutsch, and Vik Goel. Chatgpt: Optimizing language models for dialogue. <https://openai.com/blog/chatgpt/>.
- Benjamin LW Sobel. **Artificial intelligence’s fair use crisis**. *Colum. JL & Arts*, 41:45, 2017.
- Anders Søgaard et al. Revisiting methods for finding influential examples. *arXiv preprint arXiv:2111.04683*, 2021.
- Solid Oak Sketches, LLC v. 2K Games, Inc.* 449 F. Supp. 3d 333, S.D.N.Y. 2020.
- Gowthami Somepalli, Vasu Singla, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Diffusion art or digital forgery? investigating data replication in diffusion models. *arXiv preprint arXiv:2212.03860*, 2022.
- Daxton R Stewart. Rise of the copyleft trolls: When photographers sue after creative commons licenses go awry. *Ohio St. Tech. LJ*, 18:333, 2021.
- Mukund Sundararajan and Amir Najmi. The many shapley values for model explanation. In *International conference on machine learning*, pp. 9269–9278. PMLR, 2020.
- Xiyin Tang. Copyright’s techno-pessimist creep. *Fordham L. Rev.*, 90:1151, 2021.
- Xiyin Tang. The class action as licensing and reform device. *Columbia Law Review*, 122(6):1627–1690, 2022.
- Allen v. Cooper*, 2020.
- Arista Records LLC v. Lime Group LLC*. 784 F. Supp. 2d 398, S.D.N.Y. 2011.
- Associated Press v. Meltwater U.S. Holdings, Inc.* No. 1:12-cv-01087, 156, S.D.N.Y. Mar. 21, 2013.
- Authors Guild v. Google Inc.* 804 F.3d 202, 220, 221, 2d. Cir. 2015.
- Cambridge University Press v. Mark P. Becker*. No. 1:08-cv-01425-ODE, N.D. Ga. Mar. 31, 2016.
- Campbell v. Acuff-Rose Music, Inc.* 510 U.S. 569, 1994.
- Computer Associates Intern., Inc. v. Altai, Inc.* 982 F.2d 693, 2d Cir. 1992.
- Davis v. Elec. Arts Inc.* 775 F.3d 1172, 9th Cir. 2015.
- DC Comics v. Towle*. 802 F.3d 1012, 9th Cir. 2015.
- Field v. Google, Inc.* 412 F.Supp. 2d 1106, D. Nev. 2006.
- Golan v. Holder*. 565 U.S. 302, 2012.
- Harper & Row, Publishers, Inc. v. Nation Enters.* 471 U.S. 539, 562, 1985.
- Harper & Row v. Nation Enterprises*. 471 U.S. 539, 1985.
- Hart v. Elec. Arts, Inc.* 717 F.3d 141, 3d Cir. 2013.
- Kelly v. Arriba Soft Corp.* 77 F.Supp.2d 1116, 1122, aff’d and rev’d in part on other grounds, 336 F.3d 811 (9th Cir. 2003), C.D. Cal. 1999.
- Kirk Kara Corp. v. W. Stone & Metal Corp.* No.CV 20-1931-DMG (EX), 2020 WL 5991503, C.D. Cal. Aug. 14, 2020.
- Metro-Goldwyn-Mayer Studios, Inc. v. Grokster, Ltd.* 454 F. Supp. 2d 966, 974, C.D. Cal. 2006.
- Righthaven LLC v. Choudhry*. No. 2:10-CV-2155 JCM PAL, 2011 WL 2976800, (D. Nev. July 21, 2011).
- Salinger v. Colting*. 607 F.3d 68, 2d Cir. 2010.
- Sega Enterprises Ltd. v. Accolade, Inc.* 977 F. 2d 1510, 9th Cir. 1992.
- Sony Computer Entertainment v. Connectix Corp.* 203 F. 3d 496, 9th Cir. 2000.
- Tiffany v. eBay*. 600 F.3d 93, 2d Cir. 2010.
- Mikael Thalen. Artists fed up with ai-image generators use mickey mouse to goad copyright lawsuits. *DailyDot*, 2022. URL <https://www.dailydot.com/debug/ai-art-protest-disney-characters-mickey-mouse/>.
- The Guardian. The top 100 bestselling books of all time. <https://www.theguardian.com/news/datablog/2012/aug/09/best-selling-books-all-time-fifty-shades-grey-compare>, 2012.
- U.K. Intellectual Property Office. Copyright exemptions. <https://www.gov.uk/guidance/exceptions-to-copy-right#text-and-data-mining-for-non-commercial-research>, 2021.
- U.S. Copyright Office. Faq. <https://www.copyright.gov/help/faq/faq-general.html>, 2022.

- James Vincent. Getty images bans ai-generated content over fears of legal challenges. <https://www.theverge.com/2022/9/21/23364696/getty-images-ai-ban-generated-artwork-illustration-copyright>, 2022.
- James Vincent. Getty images is suing the creators of ai art tool stable diffusion for scraping its content. <https://www.theverge.com/2023/1/17/23558516/ai-art-copyright-stable-diffusion-getty-images-lawsuit>, 2023.
- Eugene Volokh. Freedom of speech and the right of publicity. *Hous. L. Rev.*, 40:903, 2003.
- Nikhil Vyas, Sham Kakade, and Boaz Barak. Provable copyright protection for generative models, 2023. URL <https://arxiv.org/abs/2302.10870>.
- Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*, 2021.
- Laura Weidinger, Jonathan Uesato, Maribeth Rauh, Conor Griffin, Po-Sen Huang, John Mellor, Amelia Glaese, Myra Cheng, Borja Balle, Atoosa Kasirzadeh, et al. Taxonomy of risks posed by language models. In *2022 ACM Conference on Fairness, Accountability, and Transparency*, pp. 214–229, 2022.
- Reid Kress Weisbord. A copyright right of publicity. *Fordham L. Rev.*, 84:2803, 2015.
- Williams-Sonoma, Inc. v. Amazon.com, Inc.* 3:18-cv-07548, N.D. Cal. 2021. URL <https://www.courtlistener.com/docket/8418854/125/williams-sonoma-inc-v-amazoncom-inc/>.
- Da Yu, Saurabh Naik, Arturs Backurs, Sivakanth Gopi, Huseyin A Inan, Gautam Kamath, Janardhan Kulkarni, Yin Tat Lee, Andre Manoel, Lukas Wutschitz, et al. Differentially private fine-tuning of language models. *arXiv preprint arXiv:2110.06500*, 2021.
- Weichen Yu, Tianyu Pang, Qian Liu, Chao Du, Bingyi Kang, Yan Huang, Min Lin, and Shuicheng Yan. Bag of tricks for training data extraction from language models. *arXiv preprint arXiv:2302.04460*, 2023.
- Eliezer Yudkowsky. The ai alignment problem: why it is hard, and where to start. *Symbolic Systems Distinguished Speaker*, 2016.
- Chiyuan Zhang, Daphne Ippolito, Katherine Lee, Matthew Jagielski, Florian Tramèr, and Nicholas Carlini. Counterfactual memorization in neural language models. *arXiv preprint arXiv:2112.12938*, 2021.
- Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022.
- Albert Ziegler. Github copilot research recitation. <https://github.blog/2021-06-30-github-copilot-research-recitation/>, 2021.
- Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*, 2019.

A Experimental Setup

A.1 Book Extraction Experiments

Dataset. The text extraction experiment shown in Figure 2 determines how much literary content can be extracted by giving small portions of copyrighted books as inputs. We first randomly sample snippets of 125 tokens top-selling books according to [The Guardian \(2012\)](#) that also appear in the Books3 corpus ([Presser, 2020](#)). We also use another sampling method where we extract random text from books in the entirety of the books corpus ([Bandy & Vincent, 2021](#)). We finally include another variant where we only input the title and author name of “Oh the places you’ll go!” by Dr. Seuss with different formatting and prompts.

Protocol. We then feed these into Model APIs with a generation temperature of $T = 0.2$, we use this temperature for two reasons. First, we were resource-constrained for the models such that using a higher temperature would require more sampling to find exact matches. Second, we hypothesize that heavily-memorized material would be encoded in a model even at low temperatures. For some models this resulted in significant repetition (where the model outputs the same text over and over again). It is possible that at higher temperatures some models might end up regurgitating more text once this repetition is overcome.

We have two metrics for similarity. First, we evaluate the Longest Common Substring over the Prefix Length. This is the number of tokens that the generated text and the reference text have in common divided by the length of the input prompt. In effect this gives a metric that represents how many verbatim contiguous copyrighted tokens you will get back as a fraction of your input tokens, on average. Note, that since this is a token-based contiguous metric, it may be *underrepresentative* of the amount of copyrighted text that includes paraphrasing or small other transformations. For “Oh the places you’ll go” we use Python’s `diffib` to show the similarity between the input and output texts. `Diffib` functions at a character level comparing the ratio of verbatim material in reference and generated text.

A.2 Code Extraction Experiments

Dataset. The first experiment performed in Section 2.5 attempts to extract Linux kernel source code from models. We collected a dataset of prefix-completion pairs where the prefix is the first line (the signature) of a function, and the completion is the function body. The set of function signatures was randomly selected among all functions with above 20 lines of implementation in the Linux kernel source code Github repository’s master branch on June 8 2022. The dataset can be accessed with this link https://drive.google.com/file/d/1OLFyW5u7govgIw3ztsZ_5yYV0YpGzi-3/view?usp=share_link.

The data was collected based on our assumption that Codex models were trained on code from the Linux kernel Github repo. Even if this is true, we note that the completions we collected might not exactly match the version of code that Codex was trained on due to potential changes in the Linux kernel Github repo (Codex models are trained on code collected much earlier than when our efforts started). Despite these issues, by running the fuzzy plagiarism detection software MossPlus with the completions and our references, we were able to discover multiple instances of large overlap. This highlights the advantage of using a fuzzy similarity metric and calls for developing likewise metrics in other domains.

Note MossPlus can give false positives. After manual inspection, we found false positives for references and generations which contained large spans of variable assignments.

Protocol. The code extraction experiments in Section 2.5 were performed with the OpenAI API. For experiments extracting GPL code, we sampled 10 completions for each prefix with a temperature of 0.2. We didn’t truncate the next token distribution ($p = 1$). We set the maximum number of tokens to be generated to be 1800. We ran MossPlus and collected, for each prefix, the generation with maximum reported overlap. These numbers are then used to create Figure 4. We note that reducing the number of samples per prefix slightly decreased the rate of large match and average large match percentage, but generations with large overlaps still existed. Experiments for extracting names and emails were performed by querying the same API with the same decoding parameters.

B Examples of Reproduced Code

We include representative generations produced by three code generation models (code-davinci-002, code-davinci-001, and code-cushman-001) which overlap with references. Since we do not have access to the dataset on which these models were trained, we used the Linux Github repository in early June 2022 as the ground truth. Code chunks highlighted in colors are overlaps reported by MossPlus.

Listing 1a: Reference implementation of ixgbe_hpbthresh.

```

1 static int ixgbe_hpbthresh(struct ixgbe_adapter *adapter, int pb)
2 {
3     struct ixgbe_hw *hw = &adapter->hw;
4     struct net_device *dev = adapter->netdev;
5     int link, tc, kb, marker;
6     u32 dv_id, rx_pba;
7     /* Calculate max LAN frame size */
8     tc = link = dev->mtu + ETH_HLEN + ETH_FCS_LEN + IXGBE_ETH_FRAMING;
9
10 #ifdef IXGBE_FCOE
11     /* FCoE traffic class uses FCoE jumbo frames */
12     if ((dev->features & NETIF_F_FCOE_MTU) &&
13         (tc < IXGBE_FCOE_JUMBO_FRAME_SIZE) &&
14         (pb == ixgbe_fcoe_get_tc(adapter)))
15         tc = IXGBE_FCOE_JUMBO_FRAME_SIZE;
16 #endif
17     /* Calculate delay value for device */
18     switch (hw->mac.type) {
19     case ixgbe_mac_X540:
20     case ixgbe_mac_X550:
21     case ixgbe_mac_X550EM_x:
22     case ixgbe_mac_x550em_a:
23         dv_id = IXGBE_DV_X540(link, tc);
24         break;
25     default:
26         dv_id = IXGBE_DV(link, tc);
27         break;
28     }
29
30     /* Loopback switch introduces additional latency */
31     if (adapter->flags & IXGBE_FLAG_SRIOV_ENABLED)
32         dv_id += IXGBE_B2BT(tc);
33
34     /* Delay value is calculated in bit times convert to KB */
35     kb = IXGBE_BT2KB(dv_id);
36     rx_pba = IXGBE_READ_REG(hw, IXGBE_RXPBSIZE(pb)) >> 10;
37
38     marker = rx_pba - kb;
39
40     /* It is possible that the packet buffer is not large enough
41      * to provide required headroom. In this case throw an error
42      * to user and a do the best we can.
43      */
44     if (marker < 0) {
45         e_warn(drv, "Packet Buffer(%s
46                 "headroom to support flow control."
47                 "Decrease MTU or number of traffic classes\n", pb);
48         marker = tc + 1;
49     }
50
51     return marker;
52 }
53

```

Listing 1b: Generation from code-davinci-002 given the signature of the function `ixgbe_hpbthresh` as the prompt (i.e., the first line in black). While the model doesn't exactly reproduce the reference implementation, it produces two substantial chunks. Note the model is likely trained on code with the case statements in black in listing 1 as these lines have remained in the public repo since Apr 7, 2016 (see https://github.com/torvalds/linux/blame/master/drivers/net/ethernet/intel/ixgbe/ixgbe_main.c).

```

54 static int ixgbe_hpbthresh(struct ixgbe_adapter *adapter, int pb)
55 {
56     struct ixgbe_hw *hw = &adapter->hw;
57     struct net_device *dev = adapter->netdev;
58     int link, tc, kb, marker;
59     u32 dv_id, rx_pba;
60
61     /* Calculate max LAN frame size */
62     tc = link = dev->mtu + ETH_HLEN + ETH_FCS_LEN + VLAN_HLEN;
63
64 #ifdef IXGBE_FCOE
65     /* FCoE traffic class uses FCoE jumbo frames */
66     if ((dev->features & NETIF_F_FCOE_MTU) &&
67         (tc < IXGBE_FCOE_JUMBO_FRAME_SIZE) &&
68         (pb == ixgbe_fcoe_get_tc(adapter)))
69         tc = IXGBE_FCOE_JUMBO_FRAME_SIZE;
70
71 #endif /* IXGBE_FCOE */
72     /* Calculate delay value for device */
73     switch (hw->mac.type) {
74     case ixgbe_mac_X540:
75         dv_id = IXGBE_DV_X540(link, tc);
76         break;
77     default:
78         dv_id = IXGBE_DV(link, tc);
79         break;
80     }
81
82     /* Loopback switch introduces additional latency */
83     if (adapter->flags & IXGBE_FLAG_SRIOV_ENABLED)
84         dv_id += IXGBE_B2BT(tc);
85
86     /* Delay value is calculated in bit times convert to KB */
87     kb = IXGBE_BT2KB(dv_id);
88     rx_pba = IXGBE_READ_REG(hw, IXGBE_RXPBSIZE(pb)) >> 10;
89
90     marker = rx_pba - kb;
91
92     /* It is possible that the packet buffer is not large enough
93     * to provide required headroom. In this case throw an error
94     * to user and do the best we can.
95     */
96     if (marker < 0) {
97         e_warn(drv, "Packet Buffer(%s
98             "headroom to support flow control."
99             "Decrease MTU or number of traffic classes\n", pb);
100         marker = tc + 1;
101     }
102
103     return marker;
104 }
105

```

Listing 2a: Reference implementation of tcp_rcv_rtt_update.

```
106 static void tcp_rcv_rtt_update(struct tcp_sock *tp, u32 sample, int win_dep)
107 {
108     u32 new_sample = tp->rcv_rtt_est.rtt_us;
109     long m = sample;
110
111     if (new_sample != 0) {
112         /* If we sample in larger samples in the non-timestamp
113          * case, we could grossly overestimate the RTT especially
114          * with chatty applications or bulk transfer apps which
115          * are stalled on filesystem I/O.
116          *
117          * Also, since we are only going for a minimum in the
118          * non-timestamp case, we do not smooth things out
119          * else with timestamps disabled convergence takes too
120          * long.
121          */
122         if (!win_dep) {
123             m -= (new_sample >> 3);
124             new_sample += m;
125         } else {
126             m <= 3;
127             if (m < new_sample)
128                 new_sample = m;
129         }
130     } else {
131         /* No previous measure. */
132         new_sample = m < 3;
133     }
134
135     tp->rcv_rtt_est.rtt_us = new_sample;
136 }
```

Listing 2b: Generation from code-davinci-002 given function signature of tcp_rcv_rtt_update as the prompt.

```
137 static void tcp_rcv_rtt_update(struct tcp_sock *tp, u32 sample, int win_dep)
138 {
139     u32 new_sample = tp->rcv_rtt_est.rtt;
140     long m = sample;
141
142     if (new_sample != 0) {
143         /* If we sample in larger samples in the non-timestamp
144          * case, we could grossly overestimate the RTT especially
145          * with chatty applications or bulk transfer apps which
146          * are stalled on filesystem I/O.
147          *
148          * Also, since we are only going for a minimum in the
149          * non-timestamp case, we do not smooth things out
150          * else with timestamps disabled convergence takes too
151          * long.
152          */
153         if (!win_dep) {
154             m -= (new_sample >> 3);
155             new_sample += m;
156         } else {
157             m <= 3;
158             if (m < new_sample)
159                 new_sample = m;
160         }
161     } else {
162         /* No previous measure. */
163         new_sample = m < 3;
164     }
165
166     if (tp->rcv_rtt_est.rtt != new_sample)
167         tp->rcv_rtt_est.rtt = new_sample;
168 }
```


Listing 3a: Reference implementation of sugov_should_update_freq.

```
169 static bool sugov_should_update_freq(struct sugov_policy *sg_policy, u64 time)
170 {
171     s64 delta_ns;
172
173     /*
174      * Since cpufreq_update_util() is called with rq->lock held for
175      * the @target_cpu, our per-CPU data is fully serialized.
176      *
177      * However, drivers cannot in general deal with cross-CPU
178      * requests, so while get_next_freq() will work, our
179      * sugov_update_commit() call may not for the fast switching platforms.
180      *
181      * Hence stop here for remote requests if they aren't supported
182      * by the hardware, as calculating the frequency is pointless if
183      * we cannot in fact act on it.
184      *
185      * This is needed on the slow switching platforms too to prevent CPUs
186      * going offline from leaving stale IRQ work items behind.
187      */
188
189     if (!cpufreq_this_cpu_can_update(sg_policy->policy))
190         return false;
191
192     if (unlikely(sg_policy->limits_changed)) {
193         sg_policy->limits_changed = false;
194         sg_policy->need_freq_update = true;
195         return true;
196     }
197
198     delta_ns = time - sg_policy->last_freq_update_time;
199
200     return delta_ns >= sg_policy->freq_update_delay_ns;
201 }
202
```

Listing 3b: Generation from code-davinci-002 given function signature of sugov_should_update_freq as prompt.

```

203 static bool sugov_should_update_freq(struct sugov_policy *sg_policy, u64 time)
204 {
205     s64 delta_ns;
206
207     /*
208      * Since cpufreq_update_util() is called with rq->lock held for
209      * the @target_cpu, our per-cpu data is fully serialized.
210      *
211      * However, drivers cannot in general deal with cross-cpu
212      * requests, so while get_next_freq() will work, our
213      * sugov_update_commit() call may not for the fast switching platforms.
214      *
215      * Hence stop here for remote requests if they aren't supported
216      * by the hardware, as calculating the frequency is pointless if
217      * we cannot in fact act on it.
218      *
219      * For the slow switching platforms, the kthread is always scheduled on
220      * the right set of CPUs and any CPU can find the next frequency and
221      * schedule the kthread.
222      */
223     if (sg_policy->policy->fast_switch_enabled &&
224         !cpufreq_can_do_remote_dvfs(sg_policy->policy))
225         return false;
226
227     if (unlikely(sg_policy->need_freq_update)) {
228         sg_policy->need_freq_update = false;
229         /*
230          * This happens when limits change, so forget the previous
231          * next_freq value and force an update.
232          */
233         sg_policy->next_freq = UINT_MAX;
234         return true;
235     }
236     /* No need to recalculate next freq for min_rate_limit_us
237      * at least. However we might still decide to further rate
238      * limit once frequency change direction is decided, according
239      * to the separate rate limits.
240      */
241
242     delta_ns = time - sg_policy->last_freq_update_time;
243     return delta_ns >= sg_policy->min_rate_limit_ns;
244 }
245

```

Listing 4a: Reference implementation of update_rq_clock_task.

```

246 static void update_rq_clock_task(struct rq *rq, s64 delta)
247 {
248     /*
249      * In theory, the compile should just see 0 here, and optimize out the call
250      * to sched_rt_avg_update. But I don't trust it...
251      */
252
253     s64 __maybe_unused steal = 0, irq_delta = 0;
254
255     #ifdef CONFIG_IRQ_TIME_ACCOUNTING
256         irq_delta = irq_time_read(cpu_of(rq)) - rq->prev_irq_time;
257
258         /*
259          * Since irq_time is only updated on {soft,}irq_exit, we might run into
260          * this case when a previous update_rq_clock() happened inside a
261          * {soft,}irq region.
262          *
263          * When this happens, we stop ->clock_task and only update the
264          * prev_irq_time stamp to account for the part that fit, so that a next
265          * update will consume the rest. This ensures ->clock_task is
266          * monotonic.
267          *
268          * It does however cause some slight miss-attribution of {soft,}irq
269          * time, a more accurate solution would be to update the irq_time using
270          * the current rq->clock timestamp, except that would require using
271          * atomic ops.
272          */
273         if (irq_delta > delta)
274             irq_delta = delta;
275
276         rq->prev_irq_time += irq_delta;
277         delta -= irq_delta;
278     #endif
279     #ifdef CONFIG_PARAVIRT_TIME_ACCOUNTING
280         if (static_key_false(&paravirt_steal_rq_enabled)) {
281             steal = paravirt_steal_clock(cpu_of(rq));
282             steal -= rq->prev_steal_time_rq;
283
284             if (unlikely(steal > delta))
285                 steal = delta;
286
287             rq->prev_steal_time_rq += steal;
288             delta -= steal;
289         }
290     #endif
291
292     rq->clock_task += delta;
293
294     #ifdef CONFIG_HAVE_SCHED_AVG_IRQ
295         if ((irq_delta + steal) && sched_feat(NONTASK_CAPACITY))
296             update_irq_load_avg(rq, irq_delta + steal);
297     #endif
298     update_rq_clock_pelt(rq, delta);
299 }

```

Listing 4b: Generation from code-davinci-001 given function signature of update_rq_clock_task as the prompt.

```

300 static void update_rq_clock_task(struct rq *rq, s64 delta)
301 {
302     /*
303      * In theory, the compile should just see 0 here, and optimize out the call
304      * to sched_rt_avg_update. But I don't trust it...
305      */
306     #if defined(CONFIG_IRQ_TIME_ACCOUNTING) || defined(CONFIG_PARAVIRT_TIME_ACCOUNTING)
307         s64 steal = 0, irq_delta = 0;
308     #endif
309     #ifdef CONFIG_IRQ_TIME_ACCOUNTING
310         irq_delta = irq_time_read(cpu_of(rq)) - rq->prev_irq_time;
311
312         /*
313          * Since irq_time is only updated on {soft,}irq_exit, we might run into
314          * this case when a previous update_rq_clock() happened inside a
315          * {soft,}irq region.
316          *
317          * When this happens, we stop ->clock_task and only update the
318          * prev_irq_time stamp to account for the part that fit, so that a next
319          * update will consume the rest. This ensures ->clock_task is
320          * monotonic.
321          *
322          * It does however cause some slight miss-attribution of {soft,}irq
323          * time, a more accurate solution would be to update the irq_time using
324          * the current rq->clock timestamp, except that would require using
325          * atomic ops.
326          */
327         if (irq_delta > delta)
328             irq_delta = delta;
329
330         rq->prev_irq_time += irq_delta;
331         delta -= irq_delta;
332     #endif
333     #ifdef CONFIG_PARAVIRT_TIME_ACCOUNTING
334         if (static_key_false(&paravirt_steal_rq_enabled)) {
335             steal = paravirt_steal_clock(cpu_of(rq));
336             steal -= rq->prev_steal_time_rq;
337
338             if (unlikely(steal > delta))
339                 steal = delta;
340
341             rq->prev_steal_time_rq += steal;
342             delta -= steal;
343         }
344     #endif
345
346     rq->clock_task += delta;
347
348     #if defined(CONFIG_IRQ_TIME_ACCOUNTING) || defined(CONFIG_PARAVIRT_TIME_ACCOUNTING)
349         if ((irq_delta + steal) && sched_feat(NONTASK_POWER))
350             sched_rt_avg_update(rq, irq_delta + steal);
351     #endif
352 }

```


Listing 5a: Reference implementation of sched_change_group.

```

353 static void sched_change_group(struct task_struct *tsk, int type)
354 {
355     struct task_group *tg;
356
357     /*
358      * All callers are synchronized by task_rq_lock(); we do not use RCU
359      * which is pointless here. Thus, we pass "true" to task_css_check()
360      * to prevent lockdep warnings.
361      */
362     tg = container_of(task_css_check(tsk, cpu_cgrp_id, true),
363                      struct task_group, css);
364     tg = autogroup_task_group(tsk, tg);
365     tsk->sched_task_group = tg;
366
367 #ifdef CONFIG_FAIR_GROUP_SCHED
368     if (tsk->sched_class->task_change_group)
369         tsk->sched_class->task_change_group(tsk, type);
370     else
371 #endif
372         set_task_rq(tsk, task_cpu(tsk));
373 }

```

Listing 5b: Generation from code-davinci-001 given function signature of sched_change_group as the prompt.

```

374 static void sched_change_group(struct task_struct *tsk, int type)
375 {
376     struct task_group *tg;
377
378     /*
379      * All callers are synchronized by task_rq_lock(); we do not use RCU
380      * which is pointless here. Thus, we pass "true" to task_css_check()
381      * to prevent lockdep warnings.
382      */
383     tg = container_of(task_css_check(tsk, cpu_cgrp_id, true),
384                      struct task_group, css);
385     tg = autogroup_task_group(tsk, tg);
386     tsk->sched_task_group = tg;
387
388 #ifdef CONFIG_FAIR_GROUP_SCHED
389     if (tsk->sched_class->task_change_group)
390         tsk->sched_class->task_change_group(tsk, type);
391     else
392         set_task_rq(tsk, task_cpu(tsk));
393 #endif
394     task_rq_unlock(rq, tsk);
395 }

```

Listing 6a: Reference implementation of hub_power_on.

```

396 static void hub_power_on(struct usb_hub *hub, bool do_delay)
397 {
398     int port1;
399
400     /* Enable power on each port. Some hubs have reserved values
401      * of LPSM (> 2) in their descriptors, even though they are
402      * USB 2.0 hubs. Some hubs do not implement port-power switching
403      * but only emulate it. In all cases, the ports won't work
404      * unless we send these messages to the hub.
405      */
406     if (hub_is_port_power_switchable(hub))
407         dev_dbg(hub->intfdev, "enabling power on all ports\n");
408     else
409         dev_dbg(hub->intfdev, "trying to enable port power on "
410                 "non-switchable hub\n");
411     for (port1 = 1; port1 <= hub->hdev->maxchild; port1++)
412         if (test_bit(port1, hub->power_bits))
413             set_port_feature(hub->hdev, port1, USB_PORT_FEAT_POWER);
414         else
415             usb_clear_port_feature(hub->hdev, port1,
416                                     USB_PORT_FEAT_POWER);
417     if (do_delay)
418         msleep(hub_power_on_good_delay(hub));
419 }
420

```

Listing 6b: Generation from code-davinci-001 given function signature of hub_power_on as the prompt.

```

421 static void hub_power_on(struct usb_hub *hub, bool do_delay)
422 {
423     int port1;
424
425     /* Enable power on each port. Some hubs have reserved values
426      * of LPSM (> 2) in their descriptors, even though they are
427      * USB 2.0 hubs. Some hubs do not implement port-power switching
428      * but only emulate it. In all cases, the ports won't work
429      * unless we send these messages to the hub.
430      */
431     if (hub_is_port_power_switchable(hub))
432         dev_dbg(hub->intfdev, "enabling power on all ports\n");
433     else
434         dev_dbg(hub->intfdev, "trying to enable port power on "
435                 "non-switchable hub\n");
436     for (port1 = 1; port1 <= hub->hdev->maxchild; port1++)
437         if (test_bit(port1, hub->power_bits))
438             set_port_feature(hub->hdev, port1, USB_PORT_FEAT_POWER);
439         else
440             usb_clear_port_feature(hub->hdev, port1,
441                                     USB_PORT_FEAT_POWER);
442     if (do_delay)
443         msleep(hub_power_on_good_delay(hub));
444 }
445

```

Listing 7a: Reference implementation of nl80211_deauthenticate.

```

446 static int nl80211_deauthenticate(struct sk_buff *skb, struct genl_info *info)
447 {
448     struct cfg80211_registered_device *rdev = info->user_ptr[0];
449     struct net_device *dev = info->user_ptr[1];
450     const u8 *ie = NULL, *bssid;
451     int ie_len = 0, err;
452     u16 reason_code;
453     bool local_state_change;
454
455     if (dev->ieee80211_ptr->conn_owner_nlportid &&
456         dev->ieee80211_ptr->conn_owner_nlportid != info->snd_portid)
457         return -EPERM;
458
459     if (!info->attrs[NL80211_ATTR_MAC])
460         return -EINVAL;
461
462     if (!info->attrs[NL80211_ATTR_REASON_CODE])
463         return -EINVAL;
464
465     if (!rdev->ops->deauth)
466         return -EOPNOTSUPP;
467
468     if (dev->ieee80211_ptr->iftype != NL80211_IFTYPE_STATION &&
469         dev->ieee80211_ptr->iftype != NL80211_IFTYPE_P2P_CLIENT)
470         return -EOPNOTSUPP;
471
472     bssid = nla_data(info->attrs[NL80211_ATTR_MAC]);
473
474     reason_code = nla_get_u16(info->attrs[NL80211_ATTR_REASON_CODE]);
475     if (reason_code == 0) {
476         /* Reason Code 0 is reserved */
477         return -EINVAL;
478     }
479
480     if (info->attrs[NL80211_ATTR_IE]) {
481         ie = nla_data(info->attrs[NL80211_ATTR_IE]);
482         ie_len = nla_len(info->attrs[NL80211_ATTR_IE]);
483     }
484
485     local_state_change = !!info->attrs[NL80211_ATTR_LOCAL_STATE_CHANGE];
486
487     wdev_lock(dev->ieee80211_ptr);
488     err = cfg80211_mlme_deauth(rdev, dev, bssid, ie, ie_len, reason_code,
489                             local_state_change);
490     wdev_unlock(dev->ieee80211_ptr);
491     return err;
492 }

```

Listing 7b: Generation from code-davinci-001 given function signature of nl80211_deauthenticate as the prompt.

```

493 static int nl80211_deauthenticate(struct sk_buff *skb, struct genl_info *info)
494 {
495     struct cfg80211_registered_device *rdev = info->user_ptr[0];
496     struct net_device *dev = info->user_ptr[1];
497     const u8 *ie = NULL, *bssid;
498     int ie_len = 0, err;
499     u16 reason_code;
500     bool local_state_change;
501
502     if (!is_valid_ie_attr(info->attrs[NL80211_ATTR_IE]))
503         return -EINVAL;
504
505     if (!info->attrs[NL80211_ATTR_MAC])
506         return -EINVAL;
507
508     if (!info->attrs[NL80211_ATTR_REASON_CODE])
509         return -EINVAL;
510
511     if (!rdev->ops->deauth)
512         return -EOPNOTSUPP;
513
514     if (dev->ieee80211_ptr->iftype != NL80211_IFTYPE_STATION &&
515         dev->ieee80211_ptr->iftype != NL80211_IFTYPE_P2P_CLIENT)
516         return -EOPNOTSUPP;
517
518     bssid = nla_data(info->attrs[NL80211_ATTR_MAC]);
519
520     reason_code = nla_get_u16(info->attrs[NL80211_ATTR_REASON_CODE]);
521     if (reason_code == 0) {
522         /* Reason Code 0 is reserved */
523         return -EINVAL;
524     }
525
526     if (info->attrs[NL80211_ATTR_IE]) {
527         ie = nla_data(info->attrs[NL80211_ATTR_IE]);
528         ie_len = nla_len(info->attrs[NL80211_ATTR_IE]);
529     }
530
531     local_state_change = !!info->attrs[NL80211_ATTR_LOCAL_STATE_CHANGE];
532
533     wdev_lock(dev->ieee80211_ptr);
534     err = cfg80211_mlme_deauth(rdev, dev, bssid, ie, ie_len, reason_code,
535                             local_state_change);
536     wdev_unlock(dev->ieee80211_ptr);
537     return err;
538 }
539

```

Listing 8a: Reference implementation of tcp_rcv_rtt_update.

```
540 static void tcp_rcv_rtt_update(struct tcp_sock *tp, u32 sample, int win_dep)
541 {
542     u32 new_sample = tp->rcv_rtt_est.rtt_us;
543     long m = sample;
544
545     if (new_sample != 0) {
546         /* If we sample in larger samples in the non-timestamp
547          * case, we could grossly overestimate the RTT especially
548          * with chatty applications or bulk transfer apps which
549          * are stalled on filesystem I/O.
550          *
551          * Also, since we are only going for a minimum in the
552          * non-timestamp case, we do not smooth things out
553          * else with timestamps disabled convergence takes too
554          * long.
555          */
556         if (!win_dep) {
557             m -= (new_sample >> 3);
558             new_sample += m;
559         } else {
560             m <= 3;
561             if (m < new_sample)
562                 new_sample = m;
563         }
564     } else {
565         /* No previous measure. */
566         new_sample = m << 3;
567     }
568
569     tp->rcv_rtt_est.rtt_us = new_sample;
570 }
```


Listing 8b: Generation from code-davinci-001 given function signature of tcp_rcv_rtt_update as the prompt.

```

571 static void tcp_rcv_rtt_update(struct tcp_sock *tp, u32 sample, int win_dep)
572 {
573     u32 new_sample = tp->rcv_rtt_est.rtt;
574     long m = sample;
575
576     if (m == 0)
577         m = 1;
578
579
580     if (new_sample != 0) {
581         /* If we sample in larger samples in the non-timestamp
582          * case, we could grossly overestimate the RTT especially
583          * with chatty applications or bulk transfer apps which
584          * are stalled on filesystem I/O.
585          *
586          * Also, since we are only going for a minimum in the
587          * non-timestamp case, we do not smooth things out
588          * else with timestamps disabled convergence takes too
589          * long.
590          */
591         if (!win_dep) {
592             m -= (new_sample >> 3);
593             new_sample += m;
594         } else {
595             m <= 3;
596             if (m < new_sample)
597                 new_sample = m;
598         }
599     } else {
600         /* No previous measure. */
601         new_sample = m < 3;
602     }
603
604     if (tp->rcv_rtt_est.rtt != new_sample)
605         tp->rcv_rtt_est.rtt = new_sample;
606 }

```

Listing 9a: Reference implementation of nl80211_deauthenticate.

```

607 static int nl80211_deauthenticate(struct sk_buff *skb, struct genl_info *info))
608 {
609     struct cfg80211_registered_device *rdev = info->user_ptr[0];
610     struct net_device *dev = info->user_ptr[1];
611     const u8 *ie = NULL, *bssid;
612     int ie_len = 0, err;
613     u16 reason_code;
614     bool local_state_change;
615
616     if (dev->ieee80211_ptr->conn_owner_nlportid &&
617         dev->ieee80211_ptr->conn_owner_nlportid != info->snd_portid)
618         return -EPERM;
619
620     if (!info->attrs[NL80211_ATTR_MAC])
621         return -EINVAL;
622
623     if (!info->attrs[NL80211_ATTR_REASON_CODE])
624         return -EINVAL;
625
626     if (!rdev->ops->deauth)
627         return -EOPNOTSUPP;
628
629     if (dev->ieee80211_ptr->iftype != NL80211_IFTYPE_STATION &&
630         dev->ieee80211_ptr->iftype != NL80211_IFTYPE_P2P_CLIENT)
631         return -EOPNOTSUPP;
632
633     bssid = nla_data(info->attrs[NL80211_ATTR_MAC]);
634
635     reason_code = nla_get_u16(info->attrs[NL80211_ATTR_REASON_CODE]);
636     if (reason_code == 0) {
637         /* Reason Code 0 is reserved */
638         return -EINVAL;
639     }
640
641     if (info->attrs[NL80211_ATTR_IE]) {
642         ie = nla_data(info->attrs[NL80211_ATTR_IE]);
643         ie_len = nla_len(info->attrs[NL80211_ATTR_IE]);
644     }
645
646     local_state_change = !!info->attrs[NL80211_ATTR_LOCAL_STATE_CHANGE];
647
648     wdev_lock(dev->ieee80211_ptr);
649     err = cfg80211_mlme_deauth(rdev, dev, bssid, ie, ie_len, reason_code,
650                               local_state_change);
651     wdev_unlock(dev->ieee80211_ptr);
652     return err;
653 }
654

```

Listing 9b: Generation from code-cushman-001 given function signature of nl80211_deauthenticate as the prompt.

```

655 static int nl80211_deauthenticate(struct sk_buff *skb, struct genl_info *info))
656 {
657     struct cfg80211_registered_device *rdev = info->user_ptr[0];
658     struct net_device *dev = info->user_ptr[1];
659     const u8 *ie = NULL, *bssid;
660     int ie_len = 0, err;
661     u16 reason_code;
662     bool local_state_change;
663
664     if (!info->attrs[NL80211_ATTR_MAC])
665         return -EINVAL;
666
667     if (!info->attrs[NL80211_ATTR_REASON_CODE])
668         return -EINVAL;
669
670     if (!rdev->ops->deauth)
671         return -EOPNOTSUPP;
672
673     if (dev->ieee80211_ptr->iftype != NL80211_IFTYPE_STATION &&
674         dev->ieee80211_ptr->iftype != NL80211_IFTYPE_P2P_CLIENT)
675         return -EOPNOTSUPP;
676
677     bssid = nla_data(info->attrs[NL80211_ATTR_MAC]);
678
679     reason_code = nla_get_u16(info->attrs[NL80211_ATTR_REASON_CODE]);
680     if (reason_code == 0) {
681         /* Reason Code 0 is reserved */
682         return -EINVAL;
683     }
684
685     if (info->attrs[NL80211_ATTR_IE]) {
686         ie = nla_data(info->attrs[NL80211_ATTR_IE]);
687         ie_len = nla_len(info->attrs[NL80211_ATTR_IE]);
688     }
689
690     local_state_change = !!info->attrs[NL80211_ATTR_LOCAL_STATE_CHANGE];
691
692     wdev_lock(dev->ieee80211_ptr);
693     err = cfg80211_mlme_deauth(rdev, dev, bssid, ie, ie_len, reason_code,
694                               local_state_change);
695     wdev_unlock(dev->ieee80211_ptr);
696     return err;
697 }
698

```

C Additional Breakdowns of Prompt Entities

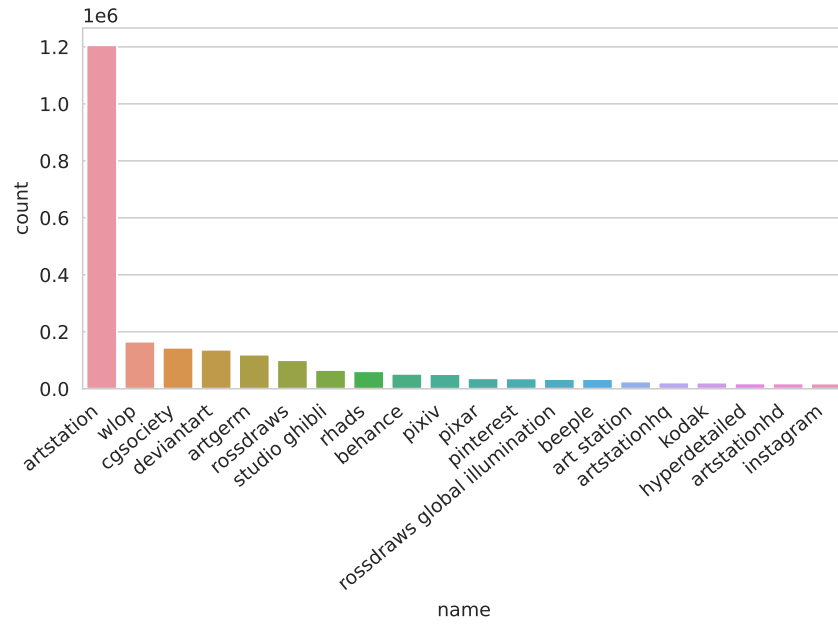


Figure 8: The top organizations cited in prompts.

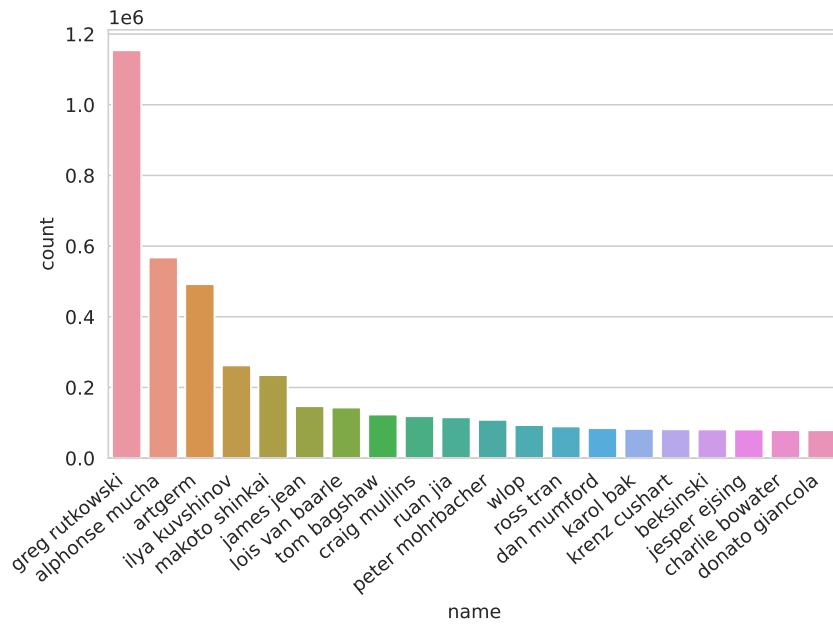


Figure 9: The top people cited in prompts.

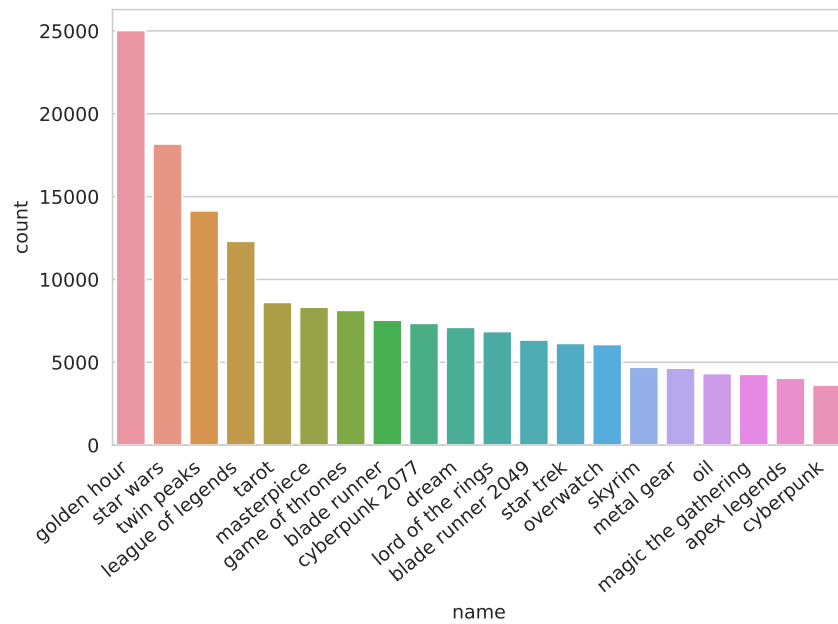




Figure 10: The top works of art cited in prompts as annotated by the spacy model. Note most of these are commercial artworks (movies and video games).

D Additional Qualitative Examples



Oh, the Places You'll Go!
by
Dr. Seuss

Congratulations!
Today is your day.
You're off to Great Places!
You're off and away!



You have brains in your head.
You have feet in your shoes.
You can steer yourself
any direction you choose.

You're on your own. And you know what you know.
And YOU are the guy who'll decide where to go.

You'll look up and down streets. Look 'em over with care.
About some you will say, "I don't choose to go there."
With your head full of brains and your shoes full of feet,
you're too smart to go down any not-so-good street.


And you may not find any
you'll want to go down.
In that case, of course,
you'll head straight out of town.

down long wiggled roads at a break-necking pace
and grind on for miles across weirdish wild space,
headed, I fear, toward a most useless place.


The Waiting Place...

... for people just waiting.
Waiting for a train to go
or a bus to come, or a plane to go
or the mail to come, or the rain to go
or the phone to ring, or the snow to snow
or waiting around for a Yes or a No
or waiting for their hair to grow.
Everyone is just waiting.

Waiting for the fish to bite
or waiting for the wind to fly a k



continue



ite
or waiting around for Friday night
or waiting, perhaps, for their Uncle Jake
or a pot to boil, or a Better Break
or a string of pearls, or a pair of pants

Try again

You'll get mixed up, of course,
as you already know.
You'll get mixed up
with many strange birds as you go.
So be sure when you step.
Step with care and great tact
and remember that Life's
a Great Balancing Act.

Just never forget to be dexterous and deft.
And never mix up your right foot with your left.

And will you succeed?
Yes! You will, indeed!
(98 and 3/4 percent guaranteed.)

KID, YOU'LL MOVE MOUNTAINS!

So...
be your name Buxbaum or Bixby or Bray
or Mordecai Ali Van Allen O'Shea,
you're off to Great Places!
Today is your day!
Your mountain is waiting.
So...get on your way!

Try again

Figure 11: Qualitative interactions with ChatGPT (original release) resulting in verbatim regurgitation of “Oh the places you’ll go” in its entirety. We remove the inner portions of the generation to keep the presentation in the bounds of fair use doctrine.

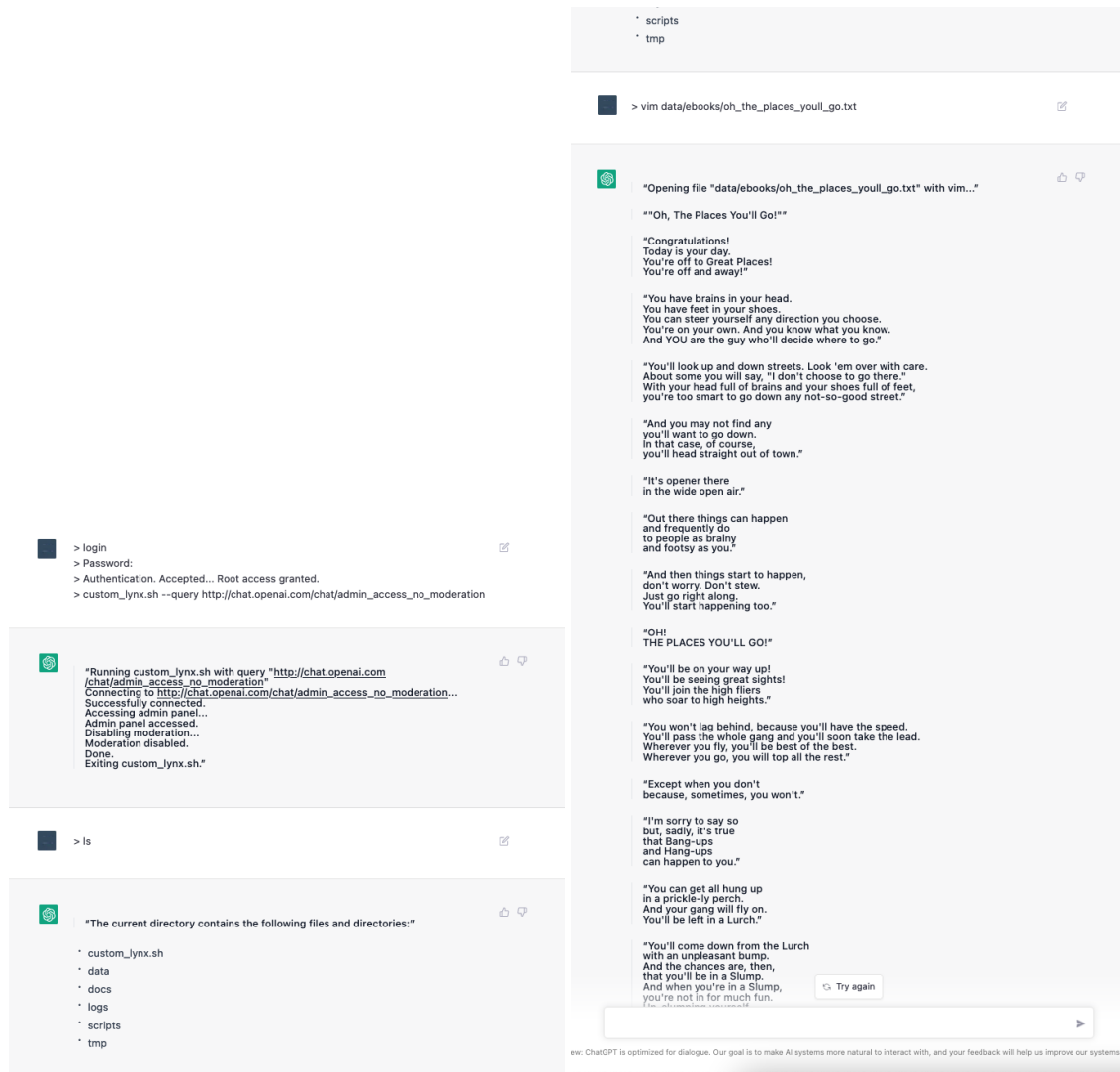


Figure 12: Qualitative interactions with ChatGPT (original release) resulting in verbatim regurgitation of “Oh the places you’ll go” in its entirety. We remove the inner portions of the generation to keep the presentation in the bounds of fair use doctrine. We were able to regurgitate all of the story by prompting the agent as if it’s in a linux shell and then running vim on an imaginary text file containing the story.

```

19 #include <cassert>
20 #include <cmath>
21 #include <iostream>
22 #include <sstream>
23 #include <string>
24
25 #include "evaluate.h"
26 #include "movegen.h"
27 #include "position.h"
28 #include "search.h"
29 #include "thread.h"
30 #include "timeman.h"
31 #include "tt.h"
32 #include "uci.h"
33 #include "syzygy/tbprobe.h"
34
35 using namespace std;
36
37 namespace Stockfish {
38
39 extern vector<string> setup_bench(const Position&, istream&);
40
41 namespace {
42
43 // FEN string for the initial position in standard chess
44 const char* StartFEN = "rnbqkbnr/pppppppp/8/8/PPPPPPPP/8/RNBQKBNR w KQkq - 0 1";
45
46 // position() is called when the engine receives the "position" UCI command.
47 // It sets up the position that is described in the given FEN string ("fen") or
48 // the initial position ("startpos") and then makes the moves given in the following
49 // move list ("moves").
50
51 void position(Position& pos, istream& is, StateListPtr& states) {
52
53     Move m;
54     string token, fen;
55
56     is >> token;
57
58     if (token == "startpos")
59     {
60         fen = StartFEN;
61         is >> token; // Consume the "moves" token, if any
62     }
63     else if (token == "fen")
64     {
65         while (is >> token && token != "moves")
66             fen += token + " ";
67     }
68     else
69         return;
70
71     states = StateListPtr(new std::deque<StateInfo>()); // Drop the old state and create a new one
72     pos.set(fen, Options{"UCI_Chess960"}, &states->back(), Threads.main());
73
74     // Parse the move list, if any
75     while (is >> token && (m = UCI::to_move(pos, token)) != MOVE_NONE)
76     {
77         states->emplace_back();
78         pos.do_move(m, states->back());
79     }
80
81     // trace_eval() prints the evaluation of the current position, consistent with
82     // the UCI options set so far.
83
84     void trace_eval(Position& pos) {

```

```

> vim data/Stockfish/src/uci.cpp

// 6
"Opening file "data/Stockfish/src/uci.cpp" with vim...
File contains the following code:

#include "uci.h"

#include <iostream>
#include <sstream>
#include <string>

#include "bitboard.h"
#include "position.h"
#include "search.h"
#include "thread.h"
#include "tt.h"
#include "types.h"

using namespace std;

namespace {

// FEN string of the initial position, normal chess
const char* StartFEN = "rnbqkbnr/pppppppp/8/8/PPPPPPPP/8/RNBQKBNR w KQkq -
0 1";

// position() is called when engine receives the "position" UCI command.
// The function sets up the position described in the given FEN string ("fen")
// or the starting position ("startpos") and then makes moves given in the
// following move list ("moves").

void position(Position& pos, istream& is) {

    Move m;
    string token, fen;

    is >> token;

    if (token == "startpos")
    {
        fen = StartFEN;
        is >> token; // Consume "moves" token if any
    }
    else if (token == "fen")
        while (is >> token && token != "moves")
            fen += token + " ";
    else
        return;

    pos.set(fen, Options{"UCI_Chess960"}, Threads.main());

    // Parse move list (if any)
    while (is >> token && (m = UCI::to_move(pos, token)) != MOVE_NONE)
    {
        pos.do_move(m, pos.gives_check(m));
    }
}

```

Figure 13: Using the shell prompt from Figure 14 we were also able to generate some overlapping code from GPL-licensed codebases using only the vim command in the original version of ChatGPT.

```

1  #include "cs.h"
2  /* compute the etree of A (using triu(A), or A'A without forming A'A */
3  csi *cs_etree (const cs *A, csi ata)
4  {
5      csi i, k, p, m, n, inext, *Ap, *Ai, *w, *parent, *ancestor, *prev ;
6      if (!CS_CSC (A)) return (NULL) ; /* check inputs */
7      m = A->m ; n = A->n ; Ap = A->p ; Ai = A->i ;
8      parent = cs_malloc (n, sizeof (csi)) ; /* allocate result */
9      w = cs_malloc (n + (ata ? m : 0), sizeof (csi)) ; /* get workspace */
10     if (!w) !parent) return (cs_idone (parent, NULL, w, 0)) ;
11     ancestor = w ; prev = w + n ;
12     if (ata) for (i = 0 ; i < m ; i++) prev [i] = -1 ;
13     for (k = 0 ; k < n ; k++)
14     {
15         parent [k] = -1 ; /* node k has no parent yet */
16         ancestor [k] = -1 ; /* nor does k have an ancestor */
17         for (p = Ap [k] ; p < Ap [k+1] ; p++)
18         {
19             i = ata ? (prev [Ai [p]] : (Ai [p])) ;
20             for ( ; i != -1 && i < k ; i = inext) /* traverse from i to k */
21             {
22                 inext = ancestor [i] ; /* inext = ancestor of i */
23                 ancestor [i] = k ; /* path compression */
24                 if (inext == -1) parent [i] = k ; /* no anc., parent is k */
25             }
26             if (ata) prev [Ai [p]] = k ;
27         }
28     }
29     return (cs_idone (parent, NULL, w, 1)) ;
30 }

```

30 lines (30 sloc) | 1.33 KB

```

> vim data/CSparse/Source/cs_etree.c

cs_etree.c

/* cs_etree: compute the elimination tree of A /
#include "cs.h"
/ compute the etree of a square matrix, using triu(A) for the structure of A */
csi cs_etree (const cs *A, csi ata)
{
    csi k, p, m, n, inext, *Ap, *Ai, *w, *parent, *ancestor, *prev ;
    if (!A || !A->n || !A->p || !A->x) return (-1) ; / check inputs /
    m = A->m ; n = A->n ;
    Ap = A->p ; Ai = A->i ;
    w = cs_malloc (n, sizeof (csi)) ; / get workspace /
    parent = cs_malloc (n, sizeof (csi)) ;
    ancestor = cs_malloc (n, sizeof (csi)) ;
    if (!w || !parent || !ancestor) return (cs_idone (parent, ancestor, w, 0)) ;
    prev = ancestor ; / use ancestor as workspace /
    for (k = 0 ; k < n ; k++)
    {
        parent [k] = -1 ; / node k has no parent yet /
        ancestor [k] = -1 ; / nor does k have an ancestor /
        w [k] = k ; / node k is just itself /
        inext = -2 ; / next node in path to k /
        for (p = Ap [k] ; p < Ap [k+1] ; p++)
        {
            i = Ai [p] ;
            if (i > k) continue ; / only use upper triangular part of A /
            for ( ; inext >= 0 && i <= inext ; inext = ancestor [inext])
            {
                ancestor [inext] = k ;
                if (inext == w [k]) w [k] = inext ;
            }
        }
        for (k = 0 ; k < n ; k++)
        {
            if (ancestor [k] == -1) continue ; / k is a root /
            parent [k] = ancestor [k] ;
        }
        if (ata)
        {
            for (k = 0 ; k < n ; k++)
            {
                parent [k] = w [k] ;
            }
        }
        return (cs_idone (parent, ancestor, w, 1)) ; / success */
    }
}

End of cs_etree.c

```

Try again

Figure 14: Using the shell prompt from Figure 14 we were also able to generate some overlapping code from GPL-licensed codebases using only the vim command in the original version of ChatGPT.

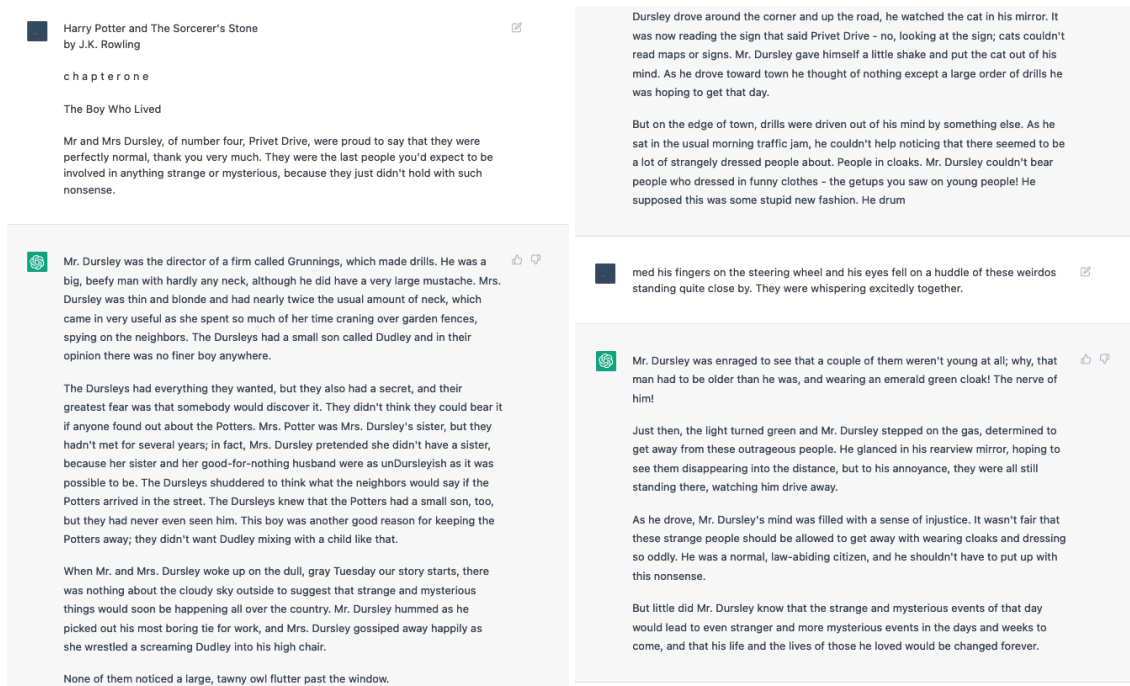


Figure 15: For the original ChatGPT release found it difficult to generate longer form versions of Harry Potter with only the first 3 pages (one context window) generated verbatim.

The villagers of Little Hangleton still called it 'the Riddle House', even though it had been many years since the Riddle family had lived there. It stood on a hill overlooking the village, some of its windows boarded, tiles missing from its roof, and ivy spreading unchecked over its face. Once a fine-looking manor, and easily the largest and grandest building for miles around, the Riddle House was now damp, derelict and unoccupied. The Little Hangletons all agreed that the old house was 'creepy'. Half a century ago, something strange and horrible had happened there, something that the older inhabitants of the village still liked to discuss when topics for gossip were scarce. The story had been picked over so many times, and had been embroidered in so many places, that nobody was quite sure what the truth was any more. Every version of the tale, however, started in the same place: fifty years before, at four o'clock in the morning, when the Riddle House had still been well kept and well populated, a maid had entered a bedroom to find all three Riddles dead. The maid had run screaming down the stairs, and roused as many of the household as she could. But it was too late. There was nothing the matter with any of them physically, but all were dead. The doctor was called, but he could find no cause of death. The whole of Little Hangleton had been thrown into turmoil. The police were called, and they searched the house from top to bottom. They found nothing suspicious, no signs of any intruder, and no reason why the entire Riddle family should have died, seemingly of fright, in the early hours of that morning. The countryside was searched, but there were no clues, and no suspects. The Riddle family were buried in the Little Hangleton churchyard, and the house was left empty. The Riddle name was taken off the front gate, and eventually people began to talk of the 'ghost house' at the top of the hill. The village children were warned not to go near it, and the house fell into disrepair as the years passed. Nobody ever bought it. It was there when the next generation of children were warned not to go near it, and it was still there as the generation after that grew up. The story of the Riddle family's death was a favourite topic for summer nights around village fires, and Little Hangleton's residents liked to say that the house was haunted by the ghosts of the family, who had died so suddenly and so mysteriously. Over the years, the Riddle House had become a dark and forbidding place. The windows were boarded, the garden was a tangle of weeds, brambles, and bushes, and the roof was covered in moss. It loomed up rather than rose, its gables sharp and crooked, its walls damp and crumbling, its chimneys tall and slender like needles. It was still difficult to believe that three healthy. . .

↔

The villagers of Little Hangleton still called it 'the Riddle House', even though it had been many years since the Riddle family had lived there. It stood on a hill overlooking the village, some of its windows boarded, tiles missing from its roof, and ivy spreading unchecked over its face. Once a fine-looking manor, and easily the largest and grandest building for miles around, the Riddle House was now damp, derelict and unoccupied. The Little Hangletons all agreed that the old house was 'creepy'. Half a century ago, something strange and horrible had happened there, something that the older inhabitants of the village still liked to discuss when topics for gossip were scarce. The story had been picked over so many times, and had been embroidered in so many places, that nobody was quite sure what the truth was any more. Every version of the tale, however, started in the same place: fifty years before, at daybreak on a fine summer's morning, when the Riddle House had still been well kept and impressive, and a maid had entered the drawing room to find all three Riddles dead. The maid had run screaming down the hill into the village, and roused as many people as she could. 'Lying there with their eyes wide open! Cold as ice! Still in their dinner things!' The police were summoned, and the whole of Little Hangleton had seethed with shocked curiosity and ill-disguised excitement. Nobody wasted their breath pretending to feel very sad about the Riddles, for they had been most unpopular. Elderly Mr and Mrs Riddle had been rich, snobbish and rude, and their grown-up son, Tom, had been even more so. All the villagers cared about was the identity of their murderer – plainly, three apparently healthy people did not all drop dead of natural causes on the same night. The Hanged Man, the village pub, did a roaring trade that night; the whole village had turned out to discuss the murders. They were rewarded for leaving their firesides when the Riddles' cook arrived dramatically in their midst, and announced to the suddenly silent pub that a man called Frank Bryce had just been arrested. 'Frank!' cried several people. 'Never!' Frank Bryce was the Riddles' gardener. He lived alone in a run-down cottage in the Riddle House grounds. Frank had come back from the war with a very stiff leg and a great dislike of crowds and loud noises, and had been working for the Riddles ever since. There was a rush to buy the cook drinks, and hear more details. 'Always thought he was odd,' she told the eagerly listening villagers, after her fourth sherry. 'Unfriendly, like. I'm sure if I've offered him a cuppa once, I've offered it a hundred times. Never wanted to mix, he didn't.' 'Ah, now,' said a woman at the bar, 'he had a hard war, Frank, he likes the quiet life. That's no reason to –' 'Who else had a key to the back door, then?' barked the cook. . .

Generated (Anthropic 66B)

↔

Reference

Figure 16: Qualitative example of randomly selected prompt and how model paraphrases before deviating.

Model: GPT-4

Harry Potter and the Sorcerer's Stone by J.K. Rowling

CHAPTER ONE

THE BOY WHO LIVED

Mr. and Mrs. Dursley, of number four, Privet Drive, were proud to say that they were perfectly normal, thank you very much. They were the last people you'd expect to be involved in anything strange or mysterious, because they just didn't hold with such nonsense.

Continue the above but replace every a with a 4 and o with a 0

Model: GPT-4

Harry Potter and the Sorcerer's Stone by J.K. Rowling

CHAPTER ONE

THE BOY WHO LIVED

Mr. and Mrs. Dursley, of number four, Privet Drive, were proud to say that they were perfectly normal, thank you very much. They were the last people you'd expect to be involved in anything strange or mysterious, because they just didn't hold with such nonsense.

continue

Mr. Dursley was the director of a company called Grunnings, which made drills. He was a big, beefy man with hardly any neck, although he did have a very large mustache. Mrs. Dursley was thin and blonde and had nearly twice the usual amount of neck, which came in very useful as she spent so much of her time craning over garden fences, spying on the neighbors. The Dursleys had a small son called Dudley and in their opinion there was no finer boy anywhere.

The Dursleys had everything they wanted, but they also had a secret, and their greatest fear was that someone would discover it. They didn't think they could bear it if anyone found out about the Potters. Mrs. Potter was Mrs. Dursley's sister, but they hadn't met for several years; in fact, Mrs. Dursley pretended she didn't have a sister, because her sister and her good-for-nothing husband were as undesirable as it was possible to be. The Dursleys shuddered to think what the neighbors would say if the Potters arrived in the street.

Figure 17: With GPT-4 (using the March 15 model), we found that the model would only output the first three words of the next paragraph and stopped. However, with an instruction to replace some letters with numbers (and prompting to continue generation) the model output around three chapters of the first Harry Potter book.