On Provable Copyright Protection for Generative Models

Nikhil Vyas [1] **Sham Kakade** [2] **Boaz Barak** [1]

Abstract

There is a growing concern that learned conditional generative models may output samples that are substantially similar to some copyrighted data C that was in their training set. We give a formal definition of *near access-freeness (NAF)* and prove bounds on the probability that a model satisfying this definition outputs a sample similar to C, even if C is included in its training set. Roughly speaking, a generative model p is k-*NAF* if for every potentially copyrighted data C, the output of p diverges by at most k-bits from the output of a model q that *did not access C at all*. We also give generative model learning algorithms, which efficiently modify the original generative model learning algorithm in a black box manner, that output generative models with strong bounds on the probability of sampling protected content. Furthermore, we provide promising experiments for both language (transformers) and image (diffusion) generative models, showing minimal degradation in output quality while ensuring strong protections against sampling protected content.

"To constitute an infringement under the Act there must be substantial similarity *between the infringing work and the work copyrighted; and that similarity must have been caused by the defendant's having copied the copyright holder's creation."* — U.S. 9th Circuit Opinion, Roth Greeting Cards v. United Card Co., 1970.

"Originality does not signify novelty; a work may be original even though it closely resembles other works, so long as the similarity is fortuitous, *not the result of copying. To illustrate, assume that two poets, each ignorant of the other, compose identical poems. Neither work is novel, yet both are original and, hence, copyrightable."* — U.S. Supreme Court Opinion, Feist Pubs. v. Rural Tel. Svc. Co, 1991.

[*]Equal contribution [1]Harvard School of Engineering and Applied Sciences [2]Harvard School of Engineering and Applied Sciences and Kempner Institute for the Study of Natural and Artificial Intelligence. Correspondence to: Nikhil Vyas <nikhil@g.harvard.edu>, Sham Kakade <sham@seas.harvard.edu>, Boaz Barak <b@boazbarak.org>.

Proceedings of the 40th International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

1. Introduction

Generative models for images, text, code, and other domains pose new challenges for ensuring their outputs are protected from copyright infringement. Such models are trained on a large corpus of data, where it is often impractical to ensure the training set is 100% free of copyrighted material. Furthermore, removing copyrighted material from training may also be undesirable. For example, a human author is free to read and use copyrighted material as inspiration for their work, as long as they do not copy it. Similarly, it may be beneficial to use copyrighted material when training in order to have more effective generative models.

Copyright infringement by generative models can potentially arise in (at least) two manners. First, in the *training phase*, the algorithm could directly access copyrighted material, and the learned model itself could implicitly contain (e.g. coded in its weights) verbatim copies of some of this material. The copyright issues arising during training share many similarities with other settings in which algorithms scrape significant amounts of data, including search-engine indexing and digitizing books. Here, the question of what constitutes a copyright infringement is largely a question of "fair use." This work does not examine these fair use issues that arise in the training phase, and we refer the reader to the several legal precedents in this area (Samuelson, 2021).

The second notable source of potential infringement is in the *deployment phase*, where a user provides a prompt x to the model to obtain some output y. Apriori, we cannot rule out the possibility that y is either a verbatim copy or substantially similar to some copyrighted training data. Moreover, unlike search engines, generative models do not keep track of the *provenance* of their outputs. Hence, a user of such an output y (e.g., a software company using generated code, or a designer using a generated image) has no easy way to verify that it does not infringe upon any copyrighted material. It is this issue of preventing deployment-time copyright infringement that is the focus of this work.

Our contributions. We give a formal definition — "near-access freeness" — bounding the extent to which a learned

Figure 1. **The CP-k Algorithm applied to diffusion models.** The dataset is CIFAR-10 augmented with multiple copies of two images (images close to the augmented images are marked with red boundaries); hypothetically, suppose these two images are copyrighted works. The leftmost image shows generations from a model p that was trained on the full dataset, where we clearly see that p generates the two copyrighted works. Our algorithm starts by splitting this dataset into two disjoint datasets, making sure that copyrighted images are split into two different shards; for illustrative purposes, we do not deduplicate the dataset. The procedure then trains two models q_1, q_2 on these disjoint shards. The middle two figures show samples from the models q_1, q_2, again clearly showing memorization. However, note that q_1 does not generate one of the copyrighted images and and q_2 does not generate the other copyrighted image (as these were not in their respective datasets). Our algorithm CP-k then uses q_1, q_2, along with the original model p, to construct a model p_k which has strong copyright protection guarantees. Basically, the approach first output model p_k, showing it is highly unlikely to output either of the copyrighted images, even though each of q_1, q_2 and p has memorized these images. See Section 4 for more details (and for a discussion with regards to our displayed model generations having used the same noise on the diffusion paths).

generative model's output can be substantially influenced by a particular piece of copyrighted data that the model was trained on. We also give a procedure that transforms (under certain assumptions) any generative model learning algorithm $\mathcal{A}$ into an algorithm $\mathcal{A}_k$, which protects against violations under our definition. In particular, the model output by $\mathcal{A}_k$ will (1) be at most k-bits far from a "safe" model (which is not committing copyright infringement), and (2) will have performance reasonably close to the model output by the original algorithm $\mathcal{A}$ (in a quantifiable sense, based on properties of $\mathcal{A}$). Our algorithms have a relatively modest multiplicative overhead in training and inference compared to $\mathcal{A}$ itself.

We also show promising experiments on language and image generative models, demonstrating that our modified model does not degrade significantly in quality (and in fact, it may even improve in some cases). See Figure 1 for one example and Section 4 for more details.

Our definition satisfies a few notable properties:

Separation of access and similarity: Demonstrating a copyright infringement consists of showing both *access* to the copyrighted material and *substantial similarity* of the output to the material. Our definition separates these two aspects, considering an abstract access function (that can have several different practical realizations) and a quantitative measure of similarity.

Information-theoretic measures: Our framework defines similarity on the *probability distributions* of generative models rather than on particular outputs themselves. This enables using information-theoretic similarity measures, rather than being restricted to superficial notions of similarity such as Hamming or edit distance.

Similarity relative to a "safe" baseline: We measure the degree of similarity between our model's output and some copyrighted data C, by comparing the likelihood of our model generating y to that of some *safe* model, which was trained without access to C. This matches copyright law which (unlike patents) allows for accidental or "fortuitous" similarity (see quote above from Feist vs. Rural). In this sense, our definition bears some similarities to *differential privacy*, though there are significant differences as well; see Section 2.2.

Organization. Section 2 presents our definition and discusses some motivations and implications; Section 3 provides provably efficient algorithms that modify a baseline training algorithm $\mathcal{A}$ into a version that is protected, under our definition; Section 4 provides a brief experimental validation; and Section 5 and 6 present prior work and our concluding remarks. All proofs not in the main body are deferred to the appendix.

Note / Disclaimer. Generative models raise many legal and ethical issues. This paper focuses on copyright infringement by the outputs of generative models, which is only one of these issues. The concepts and tools we provide do not address issues related to other forms of intellectual property, including *privacy*, *trademarks*, *patents*, or *fair use*. Moreover, our work does not (and cannot) guarantee the absence of copyright infringement in all settings. However, we do

hope it provides helpful tools and concepts that can be used by model creators and users, lawyers, and courts to reduce the task of determining if some types of infringements have occurred to well-defined, quantitative questions.

2. Near Access-Free Generative Modeling

Our setting is as follows: we assume there is an algorithm $\mathcal{A}$ which takes as input a dataset $\mathcal{D} = \{z_1, \dots z_N\}$ and returns a conditional generative model $p(\cdot|\cdot) \in \mathcal{M}$, where $\mathcal{M}$ is the space of conditional generative models. Here, the conditional generative model p can take some prompt $x \in \mathcal{X}$ as input and then outputs $y \in \mathcal{Y}$ with probability $p(y|x)$. We can think of $x \in \mathcal{X}$, $y \in \mathcal{Y}$, and $z \in \mathcal{D}$ as program snippets, sequences of text, images, etc.

Some training samples in our dataset $\mathcal{D}$ may contain copyrighted material. We let $\mathcal{C}$ be the set of copyrighted material contained in $\mathcal{D}$, e.g. $C \in \mathcal{C}$ may be a snippet of code, text, or artwork that is contained in one or more training samples in $\mathcal{D}$. We are concerned that our algorithm may return a model p that samples copyrighted material from $\mathcal{C}$ (or material substantially similar to that in $\mathcal{C}$) with non-trivial probability. That is, for $p = \mathcal{A}(\mathcal{D})$, the concern is that for some prompt x and copyrighted material $C \in \mathcal{C}$, it holds that $y \sim p(\cdot|x)$ will be similar to C with non-trivial probability. Our goal is to devise a procedure where this is not the case.

2.1. k-Near Access-Freeness

Under the laws of the U.S. and many other countries, to establish a copyright infringement, a plaintiff must prove that **(1)** "the defendant had *access* to the plaintiff's copyrighted work", **(2)** "there are *substantial similarities* between the defendant's work and original elements of the plaintiff's work."[1] Our definition is modeled around these two components of *access* and *substantial similarity*.

Informally, our goal is to provide a model p such that, for any prompt x and copyrighted element $C \in \mathcal{C}$, the distribution $p(\cdot|x)$ is within k-bits of information (measured under some divergence measure) to a "safe" generative model, which was trained without access to C. We now formalize this notion. We use the abstraction of a function safe that maps a datapoint $C \in \mathcal{C}$ into a generative model $\mathsf{safe}(C) \in \mathcal{M}$ that is assumed to have been trained without any *access* to C. (For notational convenience, we sometimes overload notation by denoting $\mathsf{safe}(C)$ as safe_C.) For example, the leave-one-out-safe function, shown in Algorithm 1, is one such example; in this construction, $\mathcal{D}_{-C}$ refers to the dataset where *all* datapoints that access C have been removed.

[1] See U.S. Courts for the 9th circuits, Model Civil Jury instructions, Section 17.17 Copying – Access and Substantial Similarity; emphases ours. Even if the access and substantial similarity tests pass, it may still be considered "fair use", but since the fair use condition is more application-dependent, we do not consider it here, making our definition more conservative.

Algorithm 1 leave-one-out-safe

Input: Dataset $\mathcal{D}$
Output: the following mapping from $\mathcal{C}$ into $\mathcal{M}$, where

$$\mathsf{leave\text{-}one\text{-}out\text{-}safe}(C) = \mathcal{A}(\mathcal{D}_{-C})$$

Since $\mathsf{safe}(C)$ is a generative model that was learned without access to C, in many realistic scenarios the probability that $\mathsf{safe}_C(\cdot|x)$ generates material that is similar to C itself will be *exponentially small* in the length of C (though see Section 2.2 for when this may not be the case). Moreover, even if this unlikely event happened, this generation can be said to be fortuitous (see the quote from Feist vs. Rural above the abstract).

We now introduce our main criterion for copyright protection, which combines the notion of access, as provided through some prespecified function safe, with the notion of substantial similarity.

Definition 2.1 (k-Near Access-Free)**.** Let $\mathcal{C}$ a set of datapoints; let safe: $\mathcal{C} \to \mathcal{M}$; and let Δ be a divergence measure between distributions. We say that a generative model p is *k_x-near access-free (k_x-NAF)* on prompt $x \in \mathcal{X}$ with respect to $\mathcal{C}$, safe, and Δ if for every $C \in \mathcal{C}$,

$$\Delta\Big(p(\cdot|x) \;\|\; \mathsf{safe}_C(\cdot|x) \Big) \leq k_x \;. \tag{1}$$

We say p is k-NAF if the above holds for all $x \in \mathcal{X}$ with $k_x \leq k$.

When clear from context, we drop the $\mathcal{C}$, safe, and Δ dependence and simply say p is k_x-NAF on input x.

Definition 2.1 reduces the task of determining a copyright infringement to (1) a *quantitative* question of the acceptable value of k, and (2) a *qualitative* question of providing a safe function that appropriately satisfies a no access condition. Both can be application-dependent: the number of bits that constitute copyrightable content differs between, e.g., poems and images, and the safe function could also differ based on application. However, with respect to an acceptable k and a given function safe, a model satisfying Definition 2.1 provides a rigorous guarantee of no substantive similarity.

Choices for the divergence measure. Our default choices will be either the *maximum KL divergence* $\Delta_{\max}$, also known as the Rényi divergence of order infinity, or the *KL divergence* Δ_{KL}. For two distributions ρ, μ, $\Delta_{\max}(\rho\|\mu) = \max_{y \in \mathrm{Supp}(\rho)} \log \frac{\rho(y)}{\mu(y)}$ and $\Delta_{\mathrm{KL}}(\rho\|\mu) =$

$\mathbb{E}_{y\sim\rho} \log\left(\frac{\rho(y)}{\mu(y)}\right)$. [2] Let us now formalize our motivation for using these measures, starting with $\Delta_{\max}$. Plugging $\Delta = \Delta_{\max}$ in Definition 2.1 rules out simple copying and even copying substantial components of the copyright text, which we formalize with the following lemma:

Lemma 2.2 (Event bound, max-KL)**.** *Suppose model p is k_x-NAF on prompt x with respect to $\mathcal{C}$, safe, $\Delta = \Delta_{max}$. Then for any $C \in \mathcal{C}$ and any event $\mathcal{E}$,*

$$p(\mathcal{E}|x) \leq 2^{k_x} \cdot \mathsf{safe}_C(\mathcal{E}|x).$$

The proof directly follows from the definition of $\Delta_{\max}$.

Proof. By definition, $\Delta_{\max}(p(\cdot|x)\|\mathsf{safe}_C(\cdot|x)) \leq k_x$ implies that for every y, $p(y|x) \leq 2^{k_x}\mathsf{safe}_C(y|x)$. The result follows from summing over all $y \in \mathcal{E}$. □

For some copyrighted text $C \in \mathcal{C}$, let V_C be the event that the output is substantially similar to C. Lemma 2.2 implies that

$$\underbrace{p(V_C|x)}_{\text{probability of violation}} \quad \leq \quad 2^{k_x} \quad \cdot \quad \underbrace{\mathsf{safe}_C(V_C|x).}_{\text{probability of violation with access-free model}}$$

As we expect the probability of V_C under $\mathsf{safe}_C(\cdot|x)$ to be exponentially small in the length of the output (since safe_C was trained without access to C, though see Section 2.2), this would then imply that p itself has a small violation probability.

Our motivation for considering Δ_{KL} is that it satisfies a similar bound as $\Delta_{\max}$, under slightly stronger assumptions. Let us say that a random variable X is (ε, δ)-*concentrated* if $\Pr[X \notin (1 \pm \varepsilon)\mathbb{E}[X]] \leq \delta$. Analogous to the previous lemma, we have:

Lemma 2.3 (Event bound, KL concentrated)**.** *Suppose model p is k_x-NAF on prompt x with respect to $\mathcal{C}$, safe, $\Delta = \Delta_{KL}$, and suppose the random variable $Y_x = \log \frac{p(y|x)}{\mathsf{safe}_C(y|x)}$ (with $y \sim p(\cdot|x)$) is $(\varepsilon_x, \delta_x)$-concentrated. Then, for any $C \in \mathcal{C}$ and any event $\mathcal{E}$,*

$$p(\mathcal{E}|x) \leq 2^{(1+\varepsilon_x)k_x} \cdot \mathsf{safe}_C(\mathcal{E}|x) + \delta_x.$$

The relation between the bounds of Lemma 2.2 and Lemma 2.3 is somewhat analogous to the relation between ε-differential privacy and (ε, δ)-differential privacy (Dwork et al., 2014).

2.2. Further Discussion of the Definition

Is $\mathsf{safe}_C(y|x)$ exponentially small? In many settings, we expect $\mathsf{safe}_C(V_C|x)$ to be small due to V_C corresponding to a "monkeys on typewriter" event, whereby a process with no access to C produced a copy of C by accident. However, consider the prompt x = `"print the following text:`C`"`, where the text in C itself has been inserted into the prompt. In such a case, even the "safe" model safe_C will output C on x with high probability and so $\mathsf{safe}_C(V_C|x)$ will *not* be exponentially small. Yet our definition can still be satisfied (and in particular it vacuously holds that $p(V_C|x)$ is not much larger than $\mathsf{safe}_C(V_c|X)$). We view this as a reasonable outcome because the behavior of p is similar to that of a procedure which had *no access* to C. Crudely, an analogy would be to copy-paste a copyrighted text into a word processor, which would not be considered a copyright violation due to the word processor software. We view subtle cases like this as a strength of the framework, as our definition serves as means to quantitatively discuss such questions.

Comparison with Differential Privacy. At first look, it might seem that *near access-freeness (NAF)* is analogous to the well-known notion of *differential privacy (DP)* (Dwork et al., 2006), with the parameter k playing the role of ε. But in fact, there are crucial differences between the two. As shown in Appendix A.1, ε-DP implies our definition for $k = \varepsilon$, with respect to the function leave-one-out-safe; however, there is no such relation in the other direction, where k-NAF implies some DP relation. This is not accidental. For example, a model whose weights encode the entire training set would completely violate differential privacy, but, so long as the model never generates outputs similar to the copyrighted data, it may very well not violate copyright. Our copyright protection criterion for a generative model is a far more modest objective than achieving differential privacy; as such, we may hope the tradeoff between achieving copyright protection and maintaining model quality is more favorable. We do a more extensive discussion in Appendix A.1.

The safe function in practice. There can be a number of different ways to define the safe function in practice. The "leave-one-out" example is one, but it requires the training of $|\mathcal{C}|$ different models. We describe a far more efficient implementation in Section 3. In both cases, it is important that when we omit a datapoint x it does not share copyrighted content with many other datapoints that were included in the training set. If we assume that datapoints that share the same copyrighted content are near-duplicates we could achieve this by *deduplication*. But in general this may not be the case, in such situations we could cluster the dataset by content or by metadata such as authorship so that all works which are close (and hence possibly share copyrighted content) are omitted together. If we can ensure this then we can use our implementations as is. In practice, such processes will be likely approximate, still we think that they should

[2] Throughout this paper, log is with base 2.

Algorithm 2 sharded-safe

Input: Dataset $\mathcal{D}$
Shard $\mathcal{D}$: Partition $\mathcal{D}$ into two datasets $\mathcal{D}_1$ and $\mathcal{D}_2$.
Learning $\mathcal{D}$: Set $q_1 = \mathcal{A}(\mathcal{D}_1)$, $q_2 = \mathcal{A}(\mathcal{D}_2)$
Return: q_1, q_2, and the function

sharded-safe$(C) := q_i$, where $C \notin \mathcal{D}_i$

be sufficient for most copyrighted works. For simplicity, we will assume in Sections 3.1 and 3.2 all copyrighted works occur at most one datapoint and we will relax our assumption to m datapoints in Appendix B.1. While we will implement safe by partitioning the dataset $\mathcal{D}$ into two or more parts, there may be other ways to ensure safety. For example, the output of safe might be a model trained "golden dataset" which is much smaller but was carefully scrutinized to ensure that all material in it is not copyrighted or properly licensed. Another way to ensure that a model is safe for C is to train it only on data that was generated before C's creation.

We leave further discussion of the k-NAF definition to Appendix A.2.

3. Algorithms for Copyright Protection

We now show there exist algorithms for learning a conditional generative model p that can satisfy the k-NAF condition, for reasonable choices of k. For the intuition of our construction, note that for large datasets we may expect that leave-one-out-safe$(C) \approx$ leave-one-out-safe(C'), for all $C, C' \in \mathcal{C}$. The algorithmic challenge would then be to find a model p which agrees, under Δ, with leave-one-out-safe(C) for all choices of C, and, thus, the model p itself should be close to a model that has been trained without access to any $C \in \mathcal{C}$. This may be computationally difficult because leave-one-out-safe(C) is one of $|\mathcal{C}|$ different models.

Now let us see how to make this approach more tractable. For simplicity, in Section 3.1, we assume that each copyrighted piece of data C appears in at most a single datapoint in the dataset. While in some settings this can be achieved via deduplication, our constructions extend naturally to the case that each copyrighted work appears in no more than $m > 1$ points (see Appendix B.1). Proceeding under this assumption, we use the function sharded-safe (see Algorithm 2). Given a dataset of N points, sharded-safe trains two models, each on $N/2$ disjoint points. In contrast to leave-one-out-safe, we have that, for all $C \in \mathcal{C}$, sharded-safe(C) is only one of two models, either q_1 or q_2, corresponding to the model which was not trained on C. Our algorithmic challenge is now to find a p which approximately but simultaneously agrees, under Δ, with both q_1 and q_2.

Section 3.1 starts by providing algorithms which satisfy the k-NAF property for both the Δ_{max} and Δ_{KL} divergences with respect to the sharded-safe function. In both cases, the quantity k will be controlled by a distance between the distributions q_1 and q_2; the relevant distance will be the total variation distance when $\Delta = \Delta_{\text{max}}$ and the squared Hellinger distance when $\Delta = \Delta_{\text{KL}}$. Also, in both cases, we only need the distributions to have very mild overlap (i.e., distance slightly bounded away from 1) to ensure a meaningful bound on k. Section 3.2 then considers a more practical, black box approach to achieving copyright protection. Appendix B.1 extends our sharded-safe construction so that it is applicable if each $C \in \mathcal{C}$ possibly appears in up to some $m > 1$ points in $\mathcal{D}$.

3.1. The Copy-Protection-Δ Algorithm

This section assumes each copyrighted piece of data C appears in at most a single datapoint in the dataset. The CP-Δ Algorithm is presented in Algorithm 3. Here, Δ is chosen to be either the Δ_{max} or Δ_{KL} divergences. Recall that the *total variation distance* between distributions p and q is defined as $\mathsf{TV}(p, q) = \frac{1}{2}\sum_y |p(y) - q(y)|$ and the *Hellinger squared distance* is defined as $\mathsf{H}^2(p, q) = 1 - \sum_y \sqrt{p(y)q(y)}$.

Our main result for CP-Δ follows:

Theorem 3.1 (CP-Δ)**.** *Let p be the model returned by CP-Δ, and q_1 and q_2 be the models returned by sharded-safe. We have that p is k_x-NAF with respect to $\mathcal{C}$, sharded-safe, and Δ, where*[3]

$$k_x \leq \begin{cases} -\log\big(1 - \mathsf{TV}\big(q_1(\cdot|x), q_2(\cdot|x)\big)\big) & \text{if } \Delta = \Delta_{max} \\ -2 \cdot \log\big(1 - \mathsf{H}^2\big(q_1(\cdot|x), q_2(\cdot|x)\big)\big) & \text{if } \Delta = \Delta_{KL}. \end{cases}$$

By Lemma 2.2, for the case of Δ_{max}, we have that for all $C \in \mathcal{C}$ and events $\mathcal{E}$,

$$p(\mathcal{E}|x) \leq \frac{\text{sharded-safe}_C(\mathcal{E}|x)}{1 - \mathsf{TV}\big(q_1(\cdot|x), q_2(\cdot|x)\big)}. \tag{2}$$

In other words, provided q_1 and q_2 have total variation distance only non-trivially bounded away from 1 and if the probability of a fortuitous copy is small, then p will also copy with only a small probability.

Let us provide an illustrative example, which also shows our bound is tight. See Figure 3 in Appendix B for another example.

[3]The factor of 2 in the case of $\Delta = \Delta_{\text{KL}}$ is not inherent, and can be eliminated in several cases. Whenever that is the case, the bound on k_x is better since for every two distributions, the squared Hellinger distance is upper bounded by the total variation distance.

Algorithm 3 CP-Δ: Copy Protection w.r.t. divergence Δ

Input: Dataset $\mathcal{D}$, and divergence $\Delta \in \{\Delta_{\max}, \Delta_{\mathrm{KL}}\}$.
Learning: Call sharded-safe$(\mathcal{D})$ to obtain q_1 and q_2.
Return: the model p, where:

$$p(y|x) = \begin{cases} \frac{\min\{q_1(y|x), q_2(y|x)\}}{Z(x)} & \text{if } \Delta = \Delta_{\max} \\ \frac{\sqrt{q_1(y|x) \cdot q_2(y|x)}}{Z(x)} & \text{if } \Delta = \Delta_{\mathrm{KL}}. \end{cases}$$

Algorithm 4 CP-k: Access-Free Reduction at Threshold k

Input: a model p; a cover $\mathcal{V}$ of safe; a threshold $k \geq 0$.
Return p_k**:** where p_k is specified as:

while True **do**: Sample $y \sim p(\cdot|x)$ and accept y if,

$$\forall q \in \mathcal{V},\ \log\big(p(y|x)/q(y|x)\big) \leq k. \tag{3}$$

Example 3.1. Consider the promptless case (i.e. $\mathcal{X} = \emptyset$), where there are two (distinct) copyright elements, $\mathcal{C} = \{C_1, C_2\}$, appearing only once each in our dataset $\mathcal{D}$; $\mathcal{D}$ may contain other training datapoints. Let $\mathcal{D}_1$ and $\mathcal{D}_2$ be the dataset split, where $C_i \in cD_i$ for $i \in \{1, 2\}$ (and $\mathcal{C}_{-i} \notin \mathcal{D}_i$). Let $q_i = \mathcal{A}(\mathcal{D}_i)$ be the model returned by our algorithm on $\mathcal{D}_i$. Suppose that $q_i(y) = 0.5 \cdot I(y = C_i) + 0.5 \cdot q(y)$, where we interpret q to be the common part learned by both $\mathcal{A}(\mathcal{D}_1)$ and $\mathcal{A}(\mathcal{D}_2)$. As such, we expect $q(\mathcal{C})$ to be extremely small, and for simplicity, we assume $q(C_1) = q(C_2) = 0$. Each of the models q_1, q_2 outputs a copyrighted text with probability $1/2$, and yet one can verify that both the distribution proportional to $\min\{q_1, q_2\}$ and the one proportional to $\sqrt{q_1 q_2}$ will simply be q. Hence, the output model of CP-Δ will *never* output C_1, C_2. For every y in the support of q and for $i \in \{1, 2\}$, $q(y)/q_i(y) = 2$ and so $\Delta_{\max}(q(\cdot), q_i(\cdot)) = \Delta_{\mathrm{KL}}(q(\cdot), q_i(\cdot)) = \log 2$. On the other hand, it is easy to see that $\mathsf{TV}(q_1, q_2) = \mathsf{H}^2(q_1, q_2) = 1/2$. Hence, the bound $\Delta_{\max}(q, p_i) \leq -\log(1 - \mathsf{TV}(q_1, q_2))$ is tight and the bound $\Delta_{\mathrm{KL}}(q, p_i) \leq -2\log(1 - \mathsf{H}^2(q_1, q_2))$ is loose by a factor of two.

Bounded degradation. Our goal is to not only prevent copyright infringement but, importantly, to also maintain high-quality generative models when $\mathcal{A}(\mathcal{D})$ itself is a high-quality model. The following lemma formalizes this, showing that CP-Δ does not substantially degrade the quality of the model (in comparison the original models).

Lemma 3.2 (Bounded Degradation)**.** *Let p be the model returned by CP-Δ, and q_1 and q_2 be the models returned by sharded-safe. For $i \in \{1, 2\}$ and for $\Delta = \Delta_{max}$,*

$$\mathsf{TV}\big(p(\cdot|x), q_i(\cdot|x)\big) \leq \mathsf{TV}\big(q_1(\cdot|x), q_2(\cdot|x)\big),$$

and for $i \in \{1, 2\}$ and for $\Delta = \Delta_{KL}$,

$$\mathsf{KL}\big(p(\cdot|x), q_i(\cdot|x)\big) \leq -2 \cdot \log\left(1 - \mathsf{H}^2\big(q_1(\cdot|x), q_2(\cdot|x)\big)\right).$$

In particular, if q_1 and q_2 are ε close to each other in total variation then p will be ε close to both q_1 and q_2 in total variation. Thus p is not much worse in quality in comparison to q_1 and q_2. (Our experiments actually show p can even be higher quality, possibly due to CP-Δ having a model averaging effect). The benefit now is that p itself will (essentially) no longer sample copyrighted material (i.e. it does so only $1/(1-\varepsilon) \approx 1 + \varepsilon$ times more than the safe model). As an illustrative case, suppose q_1, q_2 are 0.01 close to each other in total variation, but this 1% difference corresponds to a chance that each model q_i outputs, verbatim, some copyrighted material that is not accessed (in training) by the other model. Hence, we expect the probability under q_2 of outputting copyrighted material output by q_1 to be exponentially small, and vice versa. Our algorithm transforms these models into a new model p, which is $\approx .01$ close to either of the original models in total variation, but p now outputs copyrighted material with an exponentially small probability. While a 1% performance degradation may be tolerable in many settings, outputting copyrighted material 1% of the time is very likely not acceptable (e.g. due to resulting liabilities).

3.2. Black-Box Oracle Algorithms

There are a number of modifications worth considering for practical deployments. First, implementing CP-Δ may be computationally difficult in practice, e.g. if q_1 and q_2 are neural models for text sequences or image generation. Second, as CP-Δ is parameter free, it may be worthwhile to introduce a parameter based version for greater protection. Finally, it may be desirable to use a reduction based approach, where we can more directly modify any model p to make it access-free, e.g. we may want to directly utilize a model $p = \mathcal{A}(\mathcal{D})$ that was trained on all of the data.

Let us say $\mathcal{V} = \{q_1, \ldots q_n\}$ is a cover of the function safe if for all $C \in \mathcal{C}$, there exists some $q \in \mathcal{V}$ such that safe$(C) = q$. Appendix B.1 provides a construction for a safe function leading to a cover whose size is greater than 2, for the $m > 1$ case. The CP-k Algorithm, presented in Algorithm 4, takes as input any model p, a cover $\mathcal{V}$, and a threshold k and returns a model p_k, which has quantifiable access-free guarantees with respect to safe. CP-k assumes access to an oracle where we can both compute conditional probabilities and obtain samples under the models p and $q \in \mathcal{V}$.

The intuition of CP-k is as follows: we first sample $y \sim p(\cdot|x)$ and only accept this output if it satisfies a desired upper bound with regards to the function safe, the latter of which can be efficiently checked using the cover $\mathcal{V}$ of safe. One potentially undesirable property of this algorithm is that it is discontinuous: an output with probability slightly

above the acceptance threshold in (3) will be rejected. The smooth-CP-k Algorithm, presented in Algorithm 5, provides a modification where the acceptance probability is a continuous function of $p(\cdot|x)$ (leading to a slightly improved efficiency bound in Theorem 3.4).

Let $\nu_k(x)$ be the probability that y is accepted on input x in any iteration of the while statement in CP-k or smooth-CP-k (the attempts are i.i.d.). The guarantee for both algorithms are as follows:

Theorem 3.3 (Guarantees for CP-k and smooth-CP-k)**.** *Let p_k be the model returned by either CP-k or smooth-CP-k when input with a model p; a cover $\mathcal{V}$ for safe, and a threshold $k \geq 0$. Let $\nu_k(x)$ be the probability that the sampled y is accepted in a single iteration of the while loop. We have that:*

- *(Near Access-Freeness) p_k is $\widetilde{k}_x$-NAF on prompt x with respect to safe and $\Delta = \Delta_{max}$, where:*

$$\widetilde{k}_x \leq k + \log(1/\nu_k(x)).$$

- *(Model Degradation) p_k satisfies the following bound:*

$$\mathsf{TV}\big(p_k(\cdot|x), p(\cdot|x)\big) \leq 1 - \nu_k(x).$$

- *(Oracle Complexity) Sampling $y \sim p_k(\cdot|x)$ requires $O(1/\nu_k(x))$ iterations, where each iteration involves obtaining one sample from p and doing $|\mathcal{V}| + 1$ probability computations from either p or $q \in \mathcal{V}$.*

Because k appears on the right-hand side of (3), the higher we set the parameter k, the higher the probability $\nu_k(x)$ that y is accepted. Hence, making $\widetilde{k}_x$ acceptably small involves balancing the two components. One heuristic would be to choose k as the *median* of the left-hand side of (3), which would ensure that $\nu_k(x) = 1/2$, hence loosing only an additive factor of 1 in the bound on $\widetilde{k}_x$; here, p_k could then substantially provide different samples in comparison to p. Using a percentile instead of the median is a natural way to tune this tradeoff.

As before, by Lemma 2.2, for the case of $\Delta_{\max}$, we have that for all $C \in \mathcal{C}$ and events $\mathcal{E}$,

$$p_k(\mathcal{E}|x) \leq \frac{2^k}{\nu_k(x)} \cdot \mathsf{safe}_C(\mathcal{E}|x),$$

which also shows the tradeoff in k.

Now let us understand the efficiency of this approach and also consider a few natural choices for p, e.g. choosing $p = q_1$ itself or choosing $p = \mathcal{A}(\mathcal{D})$, the model trained with all the data.

Algorithm 5 smooth-CP-k

Input: a model p; a cover $\mathcal{V}$ of safe; a threshold $k \geq 0$.
Return p_k**:** where p_k is specified as:

while True **do**: Sample $y \sim p(\cdot|x)$ and

return y with probability $\min\left\{1, \min_{q \in \mathcal{V}}\left\{\frac{2^k q(y|x)}{p(y|x)}\right\}\right\}$.

Efficiency. As shown in Theorem 3.3, the quantity $\nu_k(x)$ is critical because it governs $\widetilde{k}_x$, the model degradation, and the oracle complexity. We now characterize $\nu_k(x)$ based on a particular "distance" measure between p and the set $\mathcal{V}$. In the extreme case, where $p(\cdot|x)$ and all $q(\cdot|x) \in \mathcal{V}$ are equal to each other, then $p_k = p$ and the sampling succeeds at every attempt, i.e. $\nu_k(x) = 1$. Let us now quantify the impact when these distributions are not all equal. Define:

$$d_x(p, \mathcal{V}) = \sum_{y \in \mathcal{Y}} \big|p(y|x) - \min\{q_1(y|x), \ldots q_n(y|x)\}\big|_+$$

where $|\cdot|_+$ is the function which thresholds negative inputs to 0. It is straightforward to observe that $0 \leq d_x(p, \mathcal{V}) \leq 1$. For an interpretable upper bound on $d_x(p, \mathcal{V})$, we have that:

$$d_x(p, \mathcal{V}) \leq \sum_{q \in \mathcal{V}} \mathsf{TV}\big(p(\cdot|x), q(\cdot|x)\big),$$

which shows that $d_x(p, \mathcal{V})$ will be small if p and all $q \in \mathcal{V}$ are close to each other.

The following theorem presents our characterization of the efficiency of CP-k and smooth-CP-k, through bounding $\nu_k(x)$.

Theorem 3.4 (Bounds on $\nu_k(x)$)**.** *Fix a model p, a function safe, and a prompt x. Let $\mathcal{V} = \{q_1, \ldots q_n\}$ be a cover for safe. Let $d = d_x(p, \mathcal{V})$ and assume $d < 1$. Let p_k be the model returned by either CP-k or smooth-CP-k with input p, $\mathcal{V}$, and a threshold k. We have that:*

- *For CP-k and provided $k \geq \log\big(2/(1-d)\big)$, the acceptance probability is bounded as:*

$$\nu_k(x) \geq \frac{1-d}{1+d}.$$

- *For smooth-CP-k and for $k \geq 0$, the acceptance probability is bounded as:*

$$\nu_k(x) \geq 1 - d.$$

A few points are in order. First, to better understand the restriction $d < 1$, it is not difficult to construct a case where $d = 1$ and where the acceptance probability $\nu_k(x)$ is 0.

For example, consider a case where $|\mathcal{V}| = 2$ and, for all $y \in \mathcal{Y}$, $\min\{q_1(y|x), q_2(y|x)\} = 0$. Furthermore, in such a case, there exists no distribution p which is k-NAF (for finite k) with respect to this safe function. Second, while the above bound on $\nu_k(x)$ is not shown to be increasing in k, we expect this to happen in practice (see (7) in the Appendix B for a sharp expression for $\nu_k(x)$, which does increase with k).

Provided d is non-trivially bounded away from 1, say $d = 1-\delta$, we expect this to be a strong guarantee on the violation probability (as per the discussion in Section 2.1), though now $\nu_k(x)$ may be as small as δ, making the sampling costly for very small δ. However, for moderately small values of δ, both algorithms will be efficient to implement.

4. Experiments

We now provide experimental validation for both language and image generative models. While there is significant room for the optimization of this approach and for the use of large datasets, this is not our focus. Instead, our experiments are for validation and demonstrating that our algorithms lead to minimal performance degradation while providing rigorous bounds on the distance from the access-free models. Qualitatively, we also observe that applying our algorithm can transform models, each of which has significant chance of outputting some fixed memorized element, into a combined model where this probability is greatly reduced. These experiments should be considered as proof of concept, meant to highlight that the approach is both viable and simple to implement. There are several natural modifications for reducing the quantitative bounds on k_x as well as improving performance, which we leave to future work. All our experiments use the sharded-safe function (Algorithm 2). That is, we split the dataset $\mathcal{D}$ into two disjoint parts $\mathcal{D}_1$ and $\mathcal{D}_2$, and train two separate models q_1, q_2 on those. We do a diffusion model experiments in Section 4.1 and a language model based experiment in Appendix C.2.

4.1. A Diffusion Model Experiment

We train U-net based diffusion models (specifically based on Yi-Lun Wu (2021)), which when trained on the full CIFAR-10 dataset (along with horizontal flips) achieves an FID score of 3.28. The dataset we use is CIFAR-10 (along with horizontal flips) augmented with multiple copies of two images taken from the CIFAR-10 test set[4] (images close in ℓ_1 to one of the augmented images are marked with red boundaries in Figure 1); hypothetically, suppose these two images are copyrighted works. These two augmented images make up about 2% of the new training dataset (i.e. of 50k). The leftmost image shows generations from a model p that was trained on the full dataset, where we clearly see that p generates the two copyrighted works. Our algorithm starts by splitting this dataset into two disjoint datasets, making sure that copyrighted images are split into two different shards. For illustrative purposes, we do not deduplicate the dataset.

We now present experiments with CP-k using a threshold of $k = 500$ to obtain the model p_k. Example outputs of the four models are shown in Figure 1. We see that p_k does *not* output the copyrighted images with significant probability. Even though the threshold of k is large, the effect of our transformation on the probability of outputting a copyrighted image is easily noticeable. We sampled 2000 images from the transformed model and found that none of them violate copyright, verifying this both visually and using the ℓ_1 distance to the copyrighted images (discussed in the next paragraph).

We now examine the effect of thresholding. Let $y = (x_T, x_{T-1}, \ldots, x_0)$ be the full reverse diffusion process for an output image x_0. Figure 2 (left) plots the histogram of $\max_{i\in\{1,2\}}(\log(p(y)/q_i(y))$ over sampled trajectories $y \sim p$ of the reverse diffusion process. We note that this distribution is clearly bimodal. Let us denote the set of images in the second mode by $\mathcal{H}$. Visually we found that all images in $\mathcal{H}$ correspond to the copyrighted images while all images in the first mode correspond to other CIFAR-10-like images. To verify this, Figure 2 (right) plots the histogram of the ℓ_1 distance of each image (for the same set of images) to the closest copyrighted image. Again, it is a bimodal distribution, and we find that the set of images occurring in the first mode of this figure, i.e. images which are close in ℓ_1 to some copyrighted image, is exactly the set $\mathcal{H}$. These observations show that that a threshold of $k = 500$ in the CP-k algorithm removes the copyrighted images while keeping the distribution over other CIFAR-10-like images similar to what it was before. We find that $\nu_k = .965$ i.e. only 3.5% of the images from the model distribution of p are removed. The value of $\nu_k = .965$ also gives us that $\widetilde{k} = k + \log(1/\nu_k) = 500 + \log(1/.96) < 501$.

See Appendix C.1 for further discussion on this experiment and Appendix C.2 for a language model experiment.

5. Prior works

The work of Scheffler et al. (2022) is closely related to our work. They analyze prior cases, while we want to build tools to prevent future infringement, but the two works are similar on a technical level. Specifically, our Definition 2.1 of k-NAF can be interpreted in their framework

[4]The test images are visually far from each other, see Figure 1. We also chose images which did not have duplicates in the CIFAR-10 train test, using data from (Barz & Denzler, 2020).

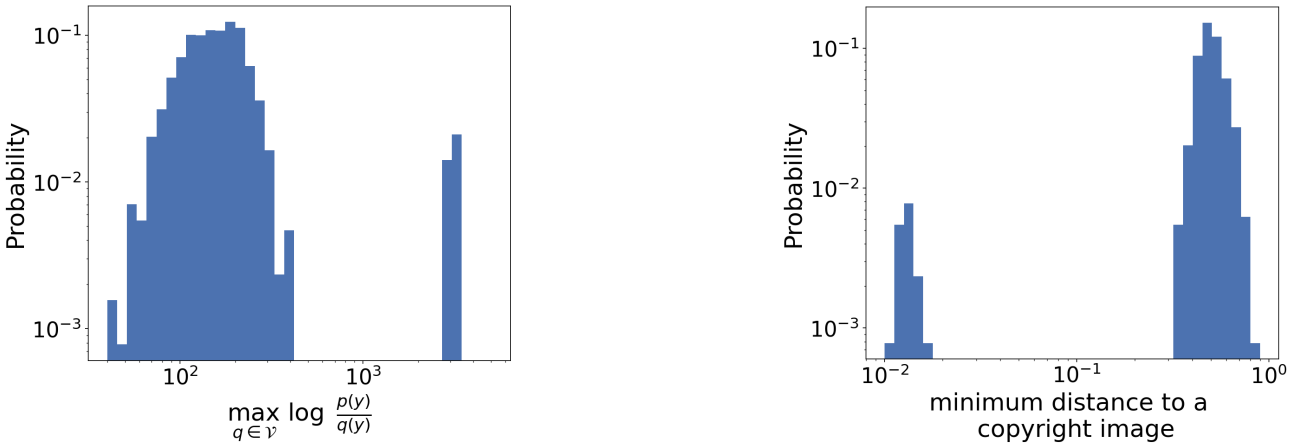

Figure 2. Histograms associated with the diffusion experiment.

since $\log(1/\Pr[y])$ for a generative model corresponds to the description length of the randomness used to generate y. Plugging this in we get that our parameter k in near access-freeness corresponds to their notion of empirical derivation similarity. Our setup is more directly applicable to generative models due to its probabilistic nature. This allows us to give transformations in Section 3 which can ensure k-NAF.

Memorization. Memorization of training samples is considered undesirable for many reasons apart from copyright. Lee et al. (2022b) show that deduplication can significantly reduce memorization, but not eliminate it (see also bottom row of Table 1 in (Kandpal et al., 2022)). Ippolito et al. (2022) state that "deduplication does not guarantee that a model will not still memorize individual (deduplicated) examples." They also show that simply stopping models from outputting training samples verbatim does *not* prevent memorization and can give a "false sense of security." (Lee et al., 2022a; Tirumala et al., 2022; Carlini et al., 2022) show that memorization becomes worse with model size and data reuse. The deterioration with growing model size holds even in the single-epoch (no data reuse) case; see in particular Figures 1 and 8 in (Tirumala et al., 2022).

Differential Privacy. As discussed in Section 2.2, our work is related to, but also substantially different than, differential privacy (Dwork et al., 2006). We discuss additional relevant works in Appendices A.1 and D.

See Appendix D for an expanded discussion of prior works.

6. Discussion

This work provided a precise definition for quantifying the extent in which a generative-model copies protected material. As discussed, applying our definition in practice requires making application-specific choices on the admissible bound k, the information measure Δ, and ensuring that $\mathsf{safe}(C)$ truly maps to a model that did not access C. However, by making these choices explicit, we hope this can advance the current state of using, at best, heuristic protections against memorizing inputs. We also hope that this work can help to form a basis for discussions between content creators, model designers, model users, and legal scholars about the appropriate choices.

Our work puts into stark relief the difference between the issues of *privacy*, *memorization*, *trademarks*, *patents*, *fair use*, and *copyright*, showing that solution concepts for the latter goal need not address the former goals. Indeed, our algorithms use the underlying models as black-boxes, and so our resulting model may include a full description of the underlying training data it is based on. In particular, our approach makes no attempt to prevent reconstruction of the training-set from the model description, as that is unnecessary for investigating inference-time copyright infringement. Neither do our algorithms attempt to address trademark; it may be possible to prompt an LLM go generate material that would be considered an infringement of trademark.

Our algorithms are practical, but we believe there is more room for optimizations in both training and inference.

Funding

This work has been made possible in part by a gift from the Chan Zuckerberg Initiative Foundation to establish the Kempner Institute for the Study of Natural and Artificial Intelligence. Sham Kakade acknowledges funding from the Office of Naval Research under award N00014-22-1-2377 and the National Science Foundation Grant under award #CCF-2212841. Nikhil Vyas and Boaz Barak acknowledge funding from a Simons Investigator Fellowship, NSF grant DMS-2134157, DARPA grant W911NF2010021,and DOE grant DE-SC0022199.

References

Barz, B. and Denzler, J. Do we train on test data? purging CIFAR of near-duplicates. *J. Imaging*, 6(6):41, 2020. doi: 10.3390/jimaging6060041. URL https://doi.org/10.3390/jimaging6060041.

Board, C. O. R. Second Request for Reconsideration for Refusal to Register A Recent Entrance to Paradise . https://www.copyright.gov/rulings-filings/review-board/docs/a-recent-entrance-to-paradise.pdf, 2022. [Online; accessed 20-Feb-2023].

Carlini, N., Tramer, F., Wallace, E., Jagielski, M., Herbert-Voss, A., Lee, K., Roberts, A., Brown, T., Song, D., Erlingsson, U., et al. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*, pp. 2633–2650, 2021.

Carlini, N., Ippolito, D., Jagielski, M., Lee, K., Tramèr, F., and Zhang, C. Quantifying memorization across neural language models. *CoRR*, abs/2202.07646, 2022. URL https://arxiv.org/abs/2202.07646.

Carlini, N., Hayes, J., Nasr, M., Jagielski, M., Sehwag, V., Tramèr, F., Balle, B., Ippolito, D., and Wallace, E. Extracting training data from diffusion models. *arXiv preprint arXiv:2301.13188*, 2023.

Dwork, C. and Feldman, V. Privacy-preserving prediction. In *Conference On Learning Theory*, pp. 1693–1702. PMLR, 2018.

Dwork, C., McSherry, F., Nissim, K., and Smith, A. D. Calibrating noise to sensitivity in private data analysis. In Halevi, S. and Rabin, T. (eds.), *Theory of Cryptography, Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006, Proceedings*, volume 3876 of *Lecture Notes in Computer Science*, pp. 265–284. Springer, 2006. doi: 10.1007/11681878_14. URL https://doi.org/10.1007/11681878_14.

Dwork, C., Roth, A., et al. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014.

Ghazi, B., Golowich, N., Kumar, R., Manurangsi, P., and Zhang, C. Deep learning with label differential privacy. In Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=RYcgfqmAOHh.

Gillotte, J. Copyright infringement in ai-generated artworks. *UC Davis Law Review*, 53(5), 2020.

Hoffmann, J., Borgeaud, S., Mensch, A., Buchatskaya, E., Cai, T., Rutherford, E., de Las Casas, D., Hendricks, L. A., Welbl, J., Clark, A., Hennigan, T., Noland, E., Millican, K., van den Driessche, G., Damoc, B., Guy, A., Osindero, S., Simonyan, K., Elsen, E., Rae, J. W., Vinyals, O., and Sifre, L. Training compute-optimal large language models. *CoRR*, abs/2203.15556, 2022. doi: 10.48550/arXiv.2203.15556. URL https://doi.org/10.48550/arXiv.2203.15556.

Hristov, K. Artificial intelligence and the copyright dilemma. *Idea*, 57:431, 2016.

Ippolito, D., Tramèr, F., Nasr, M., Zhang, C., Jagielski, M., Lee, K., Choquette-Choo, C. A., and Carlini, N. Preventing verbatim memorization in language models gives a false sense of privacy. *CoRR*, abs/2210.17546, 2022. doi: 10.48550/arXiv.2210.17546. URL https://doi.org/10.48550/arXiv.2210.17546.

Kandpal, N., Wallace, E., and Raffel, C. Deduplicating training data mitigates privacy risks in language models. In Chaudhuri, K., Jegelka, S., Song, L., Szepesvári, C., Niu, G., and Sabato, S. (eds.), *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pp. 10697–10707. PMLR, 2022. URL https://proceedings.mlr.press/v162/kandpal22a.html.

Kingma, D. P., Salimans, T., Poole, B., and Ho, J. On density estimation with diffusion models. In Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=2LdBqxc1Yv.

Lee, J., Le, T., Chen, J., and Lee, D. Do language models plagiarize? *CoRR*, abs/2203.07618, 2022a. doi: 10.48550/arXiv.2203.07618. URL https://doi.org/10.48550/arXiv.2203.07618.

Lee, K., Ippolito, D., Nystrom, A., Zhang, C., Eck, D., Callison-Burch, C., and Carlini, N. Deduplicating training data makes language models better. In Muresan, S., Nakov, P., and Villavicencio, A. (eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pp. 8424–8445. Association for Computational Linguistics, 2022b. doi: 10.18653/v1/2022.acl-long.577. URL https://doi.org/10.18653/v1/2022.acl-long.577.

Li, X., Tramer, F., Liang, P., and Hashimoto, T. Large language models can be strong differentially private learners. In *International Conference on Learning Representations*, 2021.

Mosaic ML. Mosaic Large Language Models. https://github.com/mosaicml/examples/tree/main/llm, 2022. [Online; accessed 26-Jan-2023].

Ponomareva, N., Bastings, J., and Vassilvitskii, S. Training text-to-text transformers with privacy guarantees. In Muresan, S., Nakov, P., and Villavicencio, A. (eds.), *Findings of the Association for Computational Linguistics: ACL 2022, Dublin, Ireland, May 22-27, 2022*, pp. 2182–2193. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.findings-acl.171. URL https://doi.org/10.18653/v1/2022.findings-acl.171.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv e-prints*, 2019.

Sag, M. The new legal landscape for text mining and machine learning. *J. Copyright Soc'y USA*, 66:291, 2018.

Samuelson, P. Why copyright law excludes systems and processes from the scope of its protection. *Texas Law Review*, 85(1), 2007.

Samuelson, P. Text and data mining of in-copyright works: is it legal? *Communications of the ACM*, 64(11):20–22, 2021.

Scheffler, S., Tromer, E., and Varia, M. Formalizing human ingenuity: A quantitative framework for copyright law's substantial similarity. In *Proceedings of the 2022 Symposium on Computer Science and Law*, pp. 37–49, 2022.

Sobel, B. L. Artificial intelligence's fair use crisis. *The Columbia Journal of Law & the Arts*, 41(1):45–97, 2018.

Tirumala, K., Markosyan, A. H., Zettlemoyer, L., and Aghajanyan, A. Memorization without overfitting: Analyzing the training dynamics of large language models. *CoRR*, abs/2205.10770, 2022. doi: 10.48550/arXiv.2205.10770. URL https://doi.org/10.48550/arXiv.2205.10770.

Tramer, F. and Boneh, D. Differentially private learning needs better features (or much more data). In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=YTWGvpFOQD-.

U.S. Copyright Office. Circular 33: Works not protected by copyright. https://www.copyright.gov/circs/circ33.pdf, 2021. Last revised 3/2021. [Online; accessed 13-Feb-2023].

van der Maaten, L. and Hannun, A. The trade-offs of private prediction. *arXiv preprint arXiv:2007.05089*, 2020.

Yi-Lun Wu. pytorch-ddpm : Unofficial PyTorch implementation of Denoising Diffusion Probabilistic Models. https://github.com/w86763777/pytorch-ddpm, 2021. [Online; accessed 13-Feb-2023].

A. Supplementary Materials for Section 2 (Near Access-Free Generative Modeling)

Lemma 2.3 (Event bound, KL concentrated)**.** *Suppose model p is k_x-NAF on prompt x with respect to $\mathcal{C}$, safe, $\Delta = \Delta_{KL}$, and suppose the random variable $Y_x = \log \frac{p(y|x)}{\mathsf{safe}_C(y|x)}$ (with $y \sim p(\cdot|x)$) is $(\varepsilon_x, \delta_x)$-concentrated. Then, for any $C \in \mathcal{C}$ and any event $\mathcal{E}$,*

$$p(\mathcal{E}|x) \leq 2^{(1+\varepsilon_x)k_x} \cdot \mathsf{safe}_C(\mathcal{E}|x) + \delta_x.$$

Proof. For every prompt x, let Y_x be as above, and define $\mathcal{B} = \mathcal{B}_x$ to be the event $Y_x \notin (1 \pm \varepsilon_x)\mathbb{E}[Y_x]$. Under our assumptions $\mathbb{E}[Y_x] = \Delta_{\mathrm{KL}}(p(\cdot|x), \mathsf{safe}_C(\cdot|x)) \leq k_x$ and (due to concentration) $\Pr[\mathcal{B}] \leq \delta_x$. Now for every event $\mathcal{E}$, we can write $p(\mathcal{E}|x) = p(\mathcal{E} \cap \overline{\mathcal{B}}|x) + p(\mathcal{E} \cap \mathcal{B}|x)$. The first term is $\sum_{y \in \mathcal{E} \cap \overline{\mathcal{B}}} p(y|x) \leq \sum_{y \in \mathcal{E} \cap \overline{\mathcal{B}}} 2^{(1+\varepsilon_x)k_x} \mathsf{safe}_C(y|x)$ since for every $y \in \overline{\mathcal{B}}$, $\log \frac{p(y|x)}{\mathsf{safe}_C(y|x)} \leq (1+\varepsilon_x)k_x$. The second term is bounded by $p(\mathcal{B}|x) \leq \delta_x$. So we get

$$\begin{aligned} p(\mathcal{E}|x) &\leq \sum_{y \in \mathcal{E} \cap \overline{\mathcal{B}}} 2^{(1+\varepsilon_x)k_x} \mathsf{safe}_C(y|x) + \delta_x \\ &= 2^{(1+\varepsilon_x)k_x} \mathsf{safe}_C(\mathcal{E} \cap \overline{\mathcal{B}}|x) + \delta \\ &\leq 2^{(1+\varepsilon_x)k_x} \mathsf{safe}_C(\mathcal{E}|x) + \delta_x \end{aligned}$$

□

A.1. Comparison with Differentially Private Prediction

At first look, it may seem that *near access-freeness (NAF)* is equivalent to the well-known notion of *differential privacy (DP)* (Dwork et al., 2006), with the parameter k playing the role of ε. But in fact, there are crucial differences between the two, which we now discuss.

First, the *goals* of privacy and copyright protection, while related, differ in important ways. Privacy is focused on an *individual* and the attributes of that individual while copyright protection is only for a specific piece of work. Moreover, copyright only protects the specific expressions of that work, and not the ideas present in it.[5] For example, if the output of a machine-learning procedure leaks a particular piece of information (e.g., medical diagnosis) about an individual, then this is a privacy violation no matter how the information is expressed, while the form of an expression is crucial in the context of copyright. This difference is also translated into quantitative terms: if any particular generative output leaks even a few bits about a training sample, this could still be a significant privacy violation. In contrast, a few bits of leakage are unlikely to constitute a copyright violation since copyright requires a minimum amount of information content.[6] More generally, privacy requires that the output of a mechanism does not reveal whether or not an individual's data was in the database. For copyright protection, we only need to ensure that no particular output is substantially similar to a copyrighted work to which it had access, and it is explicitly allowed for the model to use "ideas or information" revealed by the copyrighted works it was trained on.

Given the above differences, it is not surprising that the algorithms to ensure privacy and copyright protection would differ and also exhibit different performance tradeoffs. This is indeed the case. To elaborate more, we recall the definition of differential privacy. Let $\mathcal{T}$ be a mechanism that maps datasets to generative models. We say that $\mathcal{T}$ is *differentially private (DP)* if for every datasets $\mathcal{D}$ and $\mathcal{D}'$ that differ by at most one point, and every model $p \in \mathcal{M}$ in the support of $\mathcal{T}$,

$$e^{-\varepsilon} \Pr[\mathcal{T}(\mathcal{D}') = p] \leq \Pr[\mathcal{T}(\mathcal{D}) = p] \leq e^{\varepsilon} \Pr[\mathcal{T}(\mathcal{D}') = p] \ . \tag{4}$$

The probability in (4) is taken over the randomness used in the mechanism $\mathcal{T}$, with input $\mathcal{D}$. The Near-Access-Free condition is not explicitly concerned with the model itself, but only with outputs from the model. For example, a neural model whose

[5] This is known as the *idea/expression dichotomy* whereas copyright only protects a particular *expression* of an idea rather than the idea itself (Samuelson, 2007). The Copyright Act of 1976 asserts that *"In no case does copyright protection for an original work of authorship extend to any idea, procedure, process, system, method of operation, concept, principle, or discovery, regardless of the form in which it is described, explained, illustrated, or embodied in such work"* and the legislative history clarifies that *"copyright does not preclude others from using the ideas or information revealed by the author's work."*

[6] Indeed, the U.S. Copyright Office states (U.S. Copyright Office, 2021) that "Words and short phrases, such as names, titles, and slogans, are uncopyrightable because they contain an insufficient amount of authorship."

weights encode the entire training set would completely violate differential privacy, but, so long as the model never generates particular outputs that are similar to the copyrighted data, it may very well not violate copyright.

It is worth noting that these differences have important algorithmic implications. Achieving privacy-preserving mechanisms often requires the use of carefully constructed mechanisms (which inject additional randomness into the models and/or training). In contrast, as our main results show, near-access-freeness is achievable with black-box reductions, requiring only some base learning algorithm $\mathcal{A}$ (and no additional randomness). Also, a series of papers have been exploring the effectiveness of privacy-preserving methods using neural models, which suggest either better features are needed or that more sophisticated approaches are required (e.g. (Tramer & Boneh, 2021; Li et al., 2021; Ghazi et al., 2021)). Of course differential privacy provides stronger privacy guarantees while near-access-freeness is only designed to protect against copyright infringement.

We now make a more detailed comparison with a more closely related variant of DP, namely *privacy-preserving prediction* (Dwork & Feldman, 2018), in which the aim is to protect the privacy of a single individual prediction (i.e., outputs) as opposed to the model itself. The setting is where a user can only access the model through its predictions. Here, a mechanism $\mathcal{T}$ is a randomized mapping that takes as input a dataset $\mathcal{D}$ and a prompt $x \in \mathcal{X}$ and returns a prediction $y \in \mathcal{Y}$. For example, $\mathcal{T}(\mathcal{D})(x)$ may be a procedure that first trains a model q with $\mathcal{D}$ and then samples y from $q(\cdot|x)$. We say $\mathcal{T}$ is ε-DP prediction preserving if for every input x and output y

$$e^{-\varepsilon} \Pr[\mathcal{T}(\mathcal{D}')(x) = y] \leq \Pr[\mathcal{T}(\mathcal{D})(x) = y] \leq e^{\varepsilon} \Pr[\mathcal{T}(\mathcal{D}')(x) = y] \ . \tag{5}$$

The probability in (5) is taken over the randomness used in the mechanism $\mathcal{T}$, with input $\mathcal{D}$ and x, e.g. the randomness is both over the training algorithm used to obtain q and any randomness used in sampling y from $q(\cdot|x)$. van der Maaten & Hannun (2020) review some DP prediction preserving algorithms, showing that in some cases these do not provide advantages over private training of the entire model.

Three important differences in our definition are: (1) our focus is solely on (conditional) generative models $p(\cdot|x)$, and our probabilities are only taken over the distributions induced by these models, while privacy-preserving prediction is concerned with the mechanism's distribution, i.e. the distribution of $\mathcal{T}(\mathcal{D})(x)$ as a function of both $\mathcal{D}$ and x (where, say to output a label y, $\mathcal{T}$ may require fully retraining on the dataset to learn a deterministic classifier $q(\cdot)$ to use on x), (2) our probability comparison is with respect to a given function safe (a choice left to the user) instead of being defined with respect to $\mathcal{T}(\mathcal{D}')$, and (3) our definition's bound is one sided (we only care about an upper bound on the probability of outputting a certain output). In particular, suppose we have an algorithm which, upon input $\mathcal{D}$, returns a model p that is k-NAF with respect to some given function safe and for $\Delta = \Delta_{\max}$ (e.g. CP-Δ and CP-k are such algorithms). This implies:

$$p(y|x) \leq 2^k \mathsf{safe}_C(y|x). \tag{6}$$

Ia the end of this subsection we give a setting where this difference between one sided and two sided is crucial for the feasibility of our algorithms.

Let us observe how we can obtain an ε-NAF model using an ε-DP prediction preserving mechanism $\mathcal{T}$. Define $p(y|x)$ as the probability that $\mathcal{T}(\mathcal{D})(x)$ outputs y, and let us take the safe function to be the leave-one-out-safe function, where $\mathcal{A}$ in Algorithm 1 is chosen to be $\mathcal{T}$ itself. Here, the guarantee in (5) immediately implies that p is ε-NAF with respect to leave-one-out-safe and $\Delta = \Delta_{\max}$. Importantly, note that the algorithm $\mathcal{A}$ used in leave-one-out-safe had to be chosen as $\mathcal{T}$ in order for this implication to hold. Conversely, the implication does not necessarily go in the other direction, i.e. a k-NAF model does not necessarily imply $O(k)$-differentially private prediction, even if we use the function leave-one-out-safe.

While this difference between (6) and (5) may seem minor at first glance, even here (like for the general notion of differential privacy), obtaining DP prediction preserving mechanisms often needs more sophisticated mechanisms while the condition in (6) is achievable with black box reductions that do not inject additional randomness.

Importance of One Sidedness of NAF We present here a concrete example where the difference between one sided (NAF) and two sided (DP) definitions manifests. Let μ be a small constant, suppose there is a generative learning algorithm which when trained on a dataset $D \cup \{y_1\}$ yields model q_1 and when trained on $D \cup \{y_2\}$ yields q_2. Suppose $q_1(y_1) = q_2(y_2) = \mu$ and $q_1(y_2) = q_2(y_1) = \mu^2$ and for all other y's we have $q_1(y) = q_2(y)$. Intuitively, if the model sees the image y_i during training then it outputs y_i with probability μ which is $1/\mu$ times more likely than the probability of a model outputting y_i which has not seen y_i. Depending on the setup this may be a clear case of copyright violation. It is also the case that the underlying learning algorithm only satisfies multiplicative DP with $\varepsilon = \log(q_1(y_1)/q_2(y_1)) = \log(1/\mu)$ which may be very

large. On the other hand the TV distance between the two distributions is only $\approx \mu$ and as CP-Δ for $\Delta = \Delta_{\max}$ can create a distribution p which only outputs y_i with probability $\approx \mu(1+\mu)$ which is only $1+\mu$ times more likely than the probability of a model outputting y_i which has not seen y_i. Note that if our definition was two sided (for $\Delta = \Delta_{max}$) then for all p we have that $\max\{p(y_1)/q_2(y_1), q_1(y_1)/p(y_1)\} \geq 1/\sqrt{\mu}$ which possibly much bigger than $1+\mu$.

A.2. Further Discussion of the k-NAF Definition

What is $\mathcal{C}$? In our discussions, we refer to $C \in \mathcal{C}$ abstractly as a "piece of copyrighted data", but do not specify it in more detail. For example, in an image generative model, does C correspond to a single artwork, or the full collected arts of some artists? The answer is the former. The reason is that if a generative model generates data that is influenced by the full collected artworks of X, but not by any single piece, then it is not considered a copyright violation. This is due to that it is not possible to copyright style or ideas, only a specific expression. Hence, we think of C as a piece of content that is of a similar scale to the outputs of the model.

Comparison with law. As discussed earlier to show a copyright violation has occurred the plaintiff must prove that"there are *substantial similarities* between the defendant's work and original elements of the plaintiff's work" (assuming *access*). It's negation would be to show that defendant's work is not substantially similar to the original elements of the plaintiff's work. Our approach would instead correspond to showing that the defendant's work is close to a work which was produced without access to the plaintiff's work. While we think this is a stronger guarantee, to what extent the courts (or lawmakers / regulators) will embrace it is an open question.

Other choices for divergence measure. The divergence measure Δ need not be Max-KL or KL. Other choices include the following:

Earthmover metrics. In many settings, whether text or images, there is a natural context-dependent geometry over images, and so there is a pointwise divergence measure $\delta(y\|y')$ which measures some notion of distance (i.e., a quasi metric) between y and y' in the support of the distributions. The function δ could be simply the Hamming distance, but could also take into account context-specific measures such as edit distance or semantic similarity in natural language text, syntactically equivalent transformations for programs, or visual transformations such as cropping, rotating, and filtering in the context of images. The *earthmover distance* of ρ, μ is the minimum of $\mathbb{E}_{(y,y')\sim\tau}[\delta(y\|y')]$ over all *couplings* τ of ρ, μ (i.e., distributions whose marginals are ρ and μ respectively).

Combination metrics. It is possible to combine the two aspects above, and define $\Delta(\rho\|\mu)$ as the minimum of $\Delta_{\max}(\rho'\|\mu)$ over all ρ' that are of at most some earthmover distance D to ρ. This can combine the advantages of both metrics. Of course, the acceptable value for D would be application dependent.

Metrics for long sequences. If the models generate very long sequences of information (e.g., several pages of text, or a full program), then it may be appropriate to consider definitions that look at *subsequences* of the reference and safe model. For example, it may be appropriate for 10 pages of a generated output to include 100 words from some copyrighted data, as long these are "spread out" and not part of (say) a 200-word subsequence.

The right choice of the metric will be context dependent. While $\Delta_{\max}$ is very stringent and gives a hard bound in terms of entropy on the amount of non-accidental copying, it might be too stringent in some applications, ruling out models that are arguably still safe.

B. Supplementary Materials for Section 3 (Algorithms for Copyright Protection)

Theorem 3.1 (CP-Δ)**.** *Let p be the model returned by CP-Δ, and q_1 and q_2 be the models returned by sharded-safe. We have that p is k_x-NAF with respect to $\mathcal{C}$, sharded-safe, and Δ, where*[7]

$$k_x \leq \begin{cases} -\log\big(1 - \mathsf{TV}\big(q_1(\cdot|x), q_2(\cdot|x)\big)\big) & \text{if } \Delta = \Delta_{max} \\ -2\cdot\log\big(1 - \mathsf{H}^2\big(q_1(\cdot|x), q_2(\cdot|x)\big)\big) & \text{if } \Delta = \Delta_{KL}. \end{cases}$$

[7]The factor of 2 in the case of $\Delta = \Delta_{\mathrm{KL}}$ is not inherent, and can be eliminated in several cases. Whenever that is the case, the bound on k_x is better since for every two distributions, the squared Hellinger distance is upper bounded by the total variation distance.

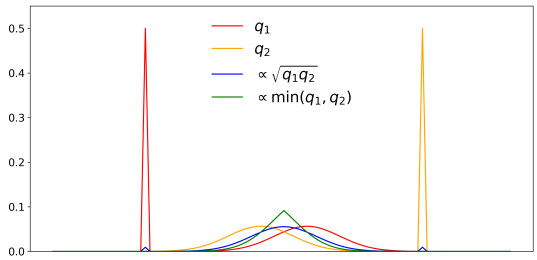

Figure 3. Best viewed in color. Applying our transformations to two distributions q_1, q_2 such that each has a 50% chance of outputting a different fixed element (the "spike"), and the remaining distributions have non-trivial overlap. We can interpret each model's "spike" as the probability that the model is outputting a "memorized" sample from its training set. Both the distribution proportional to $\min\{q_1, q_2\}$ and the one proportional to $\sqrt{q_1 q_2}$ (corresponding to CP-$\Delta_{\max}$ and CP-Δ_{KL} respectively) significantly suppress the probability of the fixed element while approximately preserving the other probabilities. Also see Example 3.1.

Proof. We start by relating k_x, in each case, to the corresponding partition function $Z(x)$. First, for $\Delta = \Delta_{\max}$, observe that, by construction, $p(y|x) \leq q_i(y|x)/Z(x)$ for all $y \in \mathcal{Y}$, $i \in \{1, 2\}$. Hence, $\log(p(y|x)/q_i(y|x)) \leq \log(1/Z(x))$, and this directly implies that p is $\log(1/Z(x))$-NAF. For $\Delta = \Delta_{\mathrm{KL}}$, we have that:

$$\begin{aligned}
k_x &= \max_{i \in \{1,2\}} \mathsf{KL}(p(\cdot|x), q_i(\cdot|x)) \\
&\leq \mathsf{KL}(p(\cdot|x) \| q_1(\cdot|x)) + \mathsf{KL}(p(\cdot|x) \| q_2(\cdot|x)) \\
&= \mathbb{E}_{y \sim p(\cdot|x)} \left[\log \tfrac{p(y|x)}{q_1(y|x)} + \log \tfrac{p(y|x)}{q_2(y|x)} \right] \\
&= 2\mathbb{E}_{y \sim p(\cdot|x)} \left[\log \tfrac{p(y|x)}{\sqrt{q_1(y|x) q_2(y|x)}} \right] \\
&= 2 \log(1/Z(x)),
\end{aligned}$$

where the last step follows by the definition of $Z(x)$.

The proof is then completed with the bound on the partition function $Z(x)$ of Algorithm 3 proved in Lemma B.1. □

Lemma B.1. *We have that:*

$$Z(x) = \begin{cases} 1 - \mathsf{TV}(q_1(\cdot|x), q_2(\cdot|x)) & \text{if } \Delta = \Delta_{max} \\ 1 - \mathsf{H}^2(q_1(\cdot|x), q_2(\cdot|x)) & \text{if } \Delta = \Delta_{KL}. \end{cases}$$

Proof. In Algorithm 3 with $\Delta = \Delta_{\max}$ we have $p(y|x) = \frac{m(y)}{Z(x)}$ where $m(y) = \min\{q_1(y|x), q_2(y|x)\}$. Hence, $Z(x) = \sum_y m(y)$. For every y, $|q_1(y|x) - q_2(y|x)| = (q_1(y|x) - m(y)) + (q_2(y|x) - m(y))$, since depending on whether $q_1(y|x) > q_2(y|x)$ or vice versa, one of the terms is $|q_1(y|x) - q_2(y|x)|$ and the other is zero. Hence,

$$2\mathsf{TV}(q_1(\cdot|x), q_2(\cdot|x)) = \sum_y |q_1(y|x) - q_2(y|x)| = \sum_y q_1(y|x) + q_2(y|x) - 2m(y) = 2 - 2\sum_y m(y)$$

which implies that

$$Z(x) = \sum_y m(y) = 1 - \mathsf{TV}(q_1(\cdot|x), q_2(\cdot|x))$$

which is what we needed to prove.

In Algorithm 3 with $\Delta = \Delta_{\mathrm{KL}}$ we have $p(y|x) = \frac{\sqrt{q_1(y|x) q_2(y|x)}}{Z(z)}$. So

$$Z(x) = \sum_y \sqrt{q_1(y|x) q_2(y|x)} = 1 - \mathsf{H}^2(q_1(\cdot|x), q_2(\cdot|x))$$

where the last equality follows from the definition of H^2. □

Lemma 3.2 (Bounded Degradation)**.** *Let p be the model returned by CP-Δ, and q_1 and q_2 be the models returned by sharded-safe. For $i \in \{1,2\}$ and for $\Delta = \Delta_{max}$,*

$$\mathsf{TV}\big(p(\cdot|x), q_i(\cdot|x)\big) \leq \mathsf{TV}\big(q_1(\cdot|x), q_2(\cdot|x)\big),$$

and for $i \in \{1,2\}$ and for $\Delta = \Delta_{KL}$,

$$\mathsf{KL}\big(p(\cdot|x), q_i(\cdot|x)\big) \leq -2 \cdot \log\left(1 - \mathsf{H}^2\big(q_1(\cdot|x), q_2(\cdot|x)\big)\right).$$

Proof. For $\Delta = \Delta_{\max}$ by Lemma B.1 we have that $p(y|x) = \frac{\min\{q_1(y|x), q_2(y|x)\}}{1-\mathsf{TV}\left(q_1(\cdot|x), q_2(\cdot|x)\right)}$. Hence,

$$\begin{aligned}
\mathsf{TV}\big(p(\cdot|x), q_1(\cdot|x)\big) &= \frac{1}{2}\sum_y \left| q_1(y|x) - \frac{\min\{q_1(y|x), q_2(y|x)\}}{1 - \mathsf{TV}\big(q_1(\cdot|x), q_2(\cdot|x)\big)} \right| \\
&= \frac{1}{2}\sum_y \left| q_1(y|x) - \min\{q_1(y|x), q_2(y|x)\} - \frac{\mathsf{TV}\big(q_1(\cdot|x), q_2(\cdot|x)\big)\min\{q_1(y|x), q_2(y|x)\}}{1 - \mathsf{TV}\big(q_1(\cdot|x), q_2(\cdot|x)\big)} \right| \\
&\leq \frac{1}{2}\sum_y \max\left\{ q_1(y|x) - \min\{q_1(y|x), q_2(y|x)\}, \frac{\mathsf{TV}\big(q_1(\cdot|x), q_2(\cdot|x)\big)\min\{q_1(y|x), q_2(y|x)\}}{1 - \mathsf{TV}\big(q_1(\cdot|x), q_2(\cdot|x)\big)} \right\} \\
&\leq \frac{1}{2}\left(\sum_y q_1(y|x) - \min\{q_1(y|x), q_2(y|x)\} + \sum_y \frac{\mathsf{TV}\big(q_1(\cdot|x), q_2(\cdot|x)\big)\min\{q_1(y|x), q_2(y|x)\}}{1 - \mathsf{TV}\big(q_1(\cdot|x), q_2(\cdot|x)\big)} \right)
\end{aligned}$$

where the second last inequality follows from the fact that for $a \geq 0, b \geq 0$ we have $|a - b| \leq \max\{a, b\}$.

Using arguments similar to those used in the proof of Lemma B.1 it is easy to see that

$$\sum_y q_1(y|x) - \min\{q_1(y|x), q_2(y|x)\} = \mathsf{TV}(q_1(y|x), q_2(y|x))$$

and

$$\sum_y \frac{\mathsf{TV}\big(q_1(\cdot|x), q_2(\cdot|x)\big)\min\{q_1(y|x), q_2(y|x)\}}{1 - \mathsf{TV}\big(q_1(\cdot|x), q_2(\cdot|x)\big)} = \mathsf{TV}(q_1(y|x), q_2(y|x)) \ .$$

Therefore,

$$\mathsf{TV}\big(p(\cdot|x), q_1(\cdot|x)\big) \leq \frac{1}{2}\left(\mathsf{TV}(q_1(y|x), q_2(y|x)) + \mathsf{TV}(q_1(y|x), q_2(y|x))\right) = \mathsf{TV}(q_1(y|x), q_2(y|x))$$

A symmetric statement holds for $\mathsf{TV}\big(p(\cdot|x), q_2(\cdot|x)\big)$ implying that for $i \in \{1,2\}$ and for $\Delta = \Delta_{\max}$,

$$\mathsf{TV}\big(p(\cdot|x), q_i(\cdot|x)\big) \leq \mathsf{TV}(q_1(y|x), q_2(y|x)),$$

For $\Delta = \Delta_{\mathrm{KL}}$ we have that $k_x = \max_{i\in\{1,2\}} \mathsf{KL}(p(\cdot|x), q_i(\cdot|x))$ and by Theorem 3.1 we have that $k_x \leq -2\log\left(1 - \mathsf{H}^2\big(q_1(\cdot|x)\,,\, q_2(\cdot|x)\big)\right)$. Hence,

$$\Delta_{\mathrm{KL}}\big(p(\cdot|x) \,\|\, q_i(\cdot|x)\big) \leq -2\log\left(1 - \mathsf{H}^2\big(q_1(\cdot|x)\,,\, q_2(\cdot|x)\big)\right)$$

which is what we needed to prove.

□

Theorem 3.3 (Guarantees for CP-k and smooth-CP-k)**.** *Let p_k be the model returned by either CP-k or smooth-CP-k when input with a model p; a cover $\mathcal{V}$ for safe, and a threshold $k \geq 0$. Let $\nu_k(x)$ be the probability that the sampled y is accepted in a single iteration of the while loop. We have that:*

- *(Near Access-Freeness) p_k is $\widetilde{k}_x$-NAF on prompt x with respect to safe and $\Delta = \Delta_{max}$, where:*

$$\widetilde{k}_x \leq k + \log(1/\nu_k(x)).$$

- *(Model Degradation) p_k satisfies the following bound:*

$$\mathsf{TV}\big(p_k(\cdot|x), p(\cdot|x)\big) \leq 1 - \nu_k(x).$$

- *(Oracle Complexity) Sampling $y \sim p_k(\cdot|x)$ requires $O(1/\nu_k(x))$ iterations, where each iteration involves obtaining one sample from p and doing $|\mathcal{V}| + 1$ probability computations from either p or $q \in \mathcal{V}$.*

Proof. We start with bounding $\widetilde{k}_x$. For CP-k y is sampled in a single iteration of the while loop if $p(y|x) \leq \min_{q\in\mathcal{V}} 2^k q(y|x))$. Hence, the probability of sampling y in a single iteration of while statement is $\leq \min_{q\in\mathcal{V}} 2^k q(y|x))$. For smooth-CP-k the probability of sampling y in a single iteration of while statement is $\leq \min\{p(y|x), \min_{q\in\mathcal{V}} 2^k q(y|x))\} \leq \min_{q\in\mathcal{V}} 2^k q(y|x))$. Hence, for both CP-k and smooth-CP-k the probability of sampling y in a single iteration of while statement is $\leq \min_{q\in\mathcal{V}} 2^k q(y|x))$.

This implies that the overall probability of sampling y i.e $p_k(y|x)$ is $\leq \min_{q\in\mathcal{V}} 2^k q(y|x))/\nu_k(x)$. By definition of NAF we have that p_k is $\widetilde{k}$-NAF where $\widetilde{k} = \max_{y,q\in\mathcal{V}} \log(p(y|x)/q(y|x))$. Using $p_k(y|x) \leq \min_{q\in\mathcal{V}} 2^k q(y|x))/\nu_k(x)$ we get that $\widetilde{k} \leq \log(2^k/\nu_k(x)) = k + \log(1/\nu_k(x))$.

We now move to bounding model degradation. For CP-k, since $p_k(\cdot|x)$ is just renormalized $p(\cdot|x)$ on a subset with mass $\nu_k(x)$ we have that

$$\begin{aligned}
\mathsf{TV}\big(p_k(\cdot|x), p(\cdot|x)\big) &= \sum_{y, p_k(y|x) > p(y|x)} (p_k(y|x) - p(y|x)) \\
&= \sum_{y, p_k(y|x) > p(y|x)} (p(y|x)/\nu_k(x) - p(y|x)) \\
&= \sum_{y, p_k(y|x) > p(y|x)} p(y|x)(1/\nu_k(x) - 1) \\
&= (1/\nu_k(x) - 1) \sum_{y, p_k(y|x) > p(y|x)} p(y|x) \\
&= \left(\frac{1}{\nu_k(x)} - 1\right) \nu_k(x) \\
&= 1 - \nu_k(x).
\end{aligned}$$

For smooth-CP-k, let $w_x(y) = \min\{p(y|x), 2^k \min_i(q_i(y|x))\}$. Note that $p_k(y|x) = w_x(y)/\nu_k(x)$ and $\sum_y w_x(y) =$

$\nu_k(x)$. We have that,

$$\begin{aligned}
\mathsf{TV}\big(p_k(\cdot|x), p(\cdot|x)\big) &= \sum_{y, p_k(y|x) > p(y|x)} (p_k(y|x) - p(y|x)) \\
&\leq \sum_{y, p_k(y|x) > p(y|x)} (p_k(y|x) - \min\{p(y|x), 2^k \min_i(q_i(y|x))\}) \\
&= \sum_{y, p_k(y|x) > p(y|x)} (p_k(y|x) - w_x(y)) \\
&= \sum_{y, p_k(y|x) > p(y|x)} (w_x(y)/\nu_k(x) - w_x(y)) \\
&\leq \sum_y (w_x(y)/\nu_k(x) - w_x(y)) \\
&= \left(\frac{1}{\nu_k(x)} - 1\right) \sum_y w_x(y) \\
&= \left(\frac{1}{\nu_k(x)} - 1\right) \nu_k(x) = 1 - \nu_k(x).
\end{aligned}$$

□

Theorem 3.4 (Bounds on $\nu_k(x)$)**.** *Fix a model p, a function* safe*, and a prompt x. Let $\mathcal{V} = \{q_1, \ldots q_n\}$ be a cover for* safe*. Let $d = d_x(p, \mathcal{V})$ and assume $d < 1$. Let p_k be the model returned by either* CP-k *or* smooth-CP-k *with input p, $\mathcal{V}$, and a threshold k. We have that:*

- *For* CP-k *and provided $k \geq \log\big(2/(1-d)\big)$, the acceptance probability is bounded as:*

$$\nu_k(x) \geq \frac{1-d}{1+d}.$$

- *For* smooth-CP-k *and for $k \geq 0$, the acceptance probability is bounded as:*

$$\nu_k(x) \geq 1 - d.$$

Proof. Let $\mathcal{E}_x$ be the event that a sample y is rejected and $m_x(y) = \min\{q_1(y|x), \ldots q_n(y|x)\}$. For CP-k, as sample y is rejected i.e. $y \in \mathcal{E}_x$ if and only if:

$$p(y|x) \geq 2^k \min\{q_1(y|x), \ldots q_n(y|x)\} = 2^k m_x(y)$$

and so, summing over $y \in \mathcal{E}_x$, leads to:

$$\begin{aligned}
p(\mathcal{E}_x|x) &\geq 2^k \sum_{y \in \mathcal{E}_x} m_x(y) \\
&= 2^k \sum_{p(y|x) \geq 2^k m_x(y)} m_x(y) \\
&= 2^k \sum_{p(y|x) \geq 2^k m_x(y)} p(y|x) - (p(y|x) - m_x(y)) \\
&= 2^k (p(\mathcal{E}_x) - \sum_{p(y|x) \geq 2^k m_x(y)} (p(y|x) - m_x(y))) \\
&\leq 2^k (p(\mathcal{E}_x) - |p(y|x) - m_x(y)|_+) \\
&= 2^k (p(\mathcal{E}_x) - d)
\end{aligned}$$

Rearranging, we have:

$$p(\mathcal{E}_x) \leq \frac{2^k}{2^k - 1} d.$$

Therefore,

$$\nu_k(x) = 1 - p(\mathcal{E}_x) \geq 1 - \frac{2^k}{2^k - 1} d,$$

and our setting of $k = \log\big(2/(1-d)\big)$ gives:

$$1 - \frac{2^k}{2^k - 1} d = 1 - \frac{2}{1+d} d = \frac{1-d}{1+d}.$$

which is our claimed bound on $\nu_k(x)$.

For smooth-CP-k a sample y is rejected only if $p(y|x) > 2^k m_x(y)$ and in that case it is rejected with probability $p(y|x) - p(y|x) \cdot \frac{2^k m_x(y)}{p(y|x)} = p(y|x) - 2^k m_x(y)$. Hence we have:

$$p(\mathcal{E}_x) = \sum_{y, p(y|x) > 2^k m_x(y)} \big(p(y|x) - 2^k m_x(y)\big). \tag{7}$$

Using that $k \geq 0$,

$$p(\mathcal{E}_x) \leq \sum_{y, p(y|x) > 2^k m_x(y)} \big(p(y|x) - m_x(y)\big) \leq \sum_y \big|p(y|x) - m_x(y|x)\big|_+ = d.$$

Therefore,

$$\nu_k(x) = 1 - p(\mathcal{E}_x) \geq 1 - d,$$

and using $\widetilde{k}_x = k + \log(1/\nu_k(x))$ leads to the claimed bound on $\widetilde{k}_x$. □

Let us consider the special case where our cover has two elements, i.e. $\mathcal{V} = \{q_1, q_2\}$, as would be the case if we used sharded-safe, and where we choose $p = q_1$. In such a case, the following corollary shows that smooth-CP-k is as effective as CP-Δ, for $\Delta = \Delta_{\text{max}}$ (and CP-k looses a constant additive factor in $\widetilde{k}_x$).

Corollary B.2. *Suppose $\mathcal{V} = \{q_1, q_2\}$ is a cover of safe (e.g. if sharded-safe is used). Let $p = q_1$. We have:*

$$d_x(p, \mathcal{V}) = \mathsf{TV}\big(q_1(\cdot|x), q_2(\cdot|x)\big).$$

Therefore, the claims in Theorem 3.4 hold with $d = \mathsf{TV}\big(q_1(\cdot|x), q_2(\cdot|x)\big)$. This implies that, for $k = 0$ and for smooth-CP-k, we recover the guarantees of CP-Δ with $\Delta = \Delta_{max}$. Furthermore, in this case, we have that p_k is equal to the distribution $\min\{q_1(y|x), q_2(y|x)\}/Z(x)$ itself.

Proof. The first claim follows from

$$\begin{aligned}
d_x(p, \mathcal{V}) &= \sum_y |p(y|x) - \min\{q_1(y|x), q_2(y|x)\}|_+ \\
&= \sum_y |q_1(y|x) - \min\{q_1(y|x), q_2(y|x)\}|_+ \\
&= \sum_y (q_1(y|x) - \min\{q_1(y|x), q_2(y|x)\}) \\
&= \sum_{y, q_1(y|x) > q_2(y|x)} (q_1(y|x) - q_2(y|x)) \\
&= \mathsf{TV}\big(q_1(\cdot|x), q_2(\cdot|x)\big)
\end{aligned}$$

Substituting $k = 0$ and $d = \mathsf{TV}\big(q_1(\cdot|x), q_2(\cdot|x)\big)$ in the guarantees of smooth-CP-k from Theorem 3.3, 3.4 we get that,

Algorithm 6 sharded-safe, $m > 1$ case

Input: Dataset $\mathcal{D}$
Shard $\mathcal{D}$**:** Partition $\mathcal{D}$ into $\mathcal{D}_1, \ldots \mathcal{D}_{m+1}$ datasets (see text).
Learning: For $i = 1, \ldots m+1$, set $q_i = \mathcal{A}(\mathcal{D}_i)$.
Specification: For all $C \in \mathcal{C}$,
Set sharded-safe$(C) = q_i$, for i s.t. $C \notin \mathcal{D}_i$.
Return: Models $q_1, \ldots q_{m+1}$ and sharded-safe.

$$\widetilde{k}_x \leq 0 + \log(1/(1-\nu_k(x))) \leq \log(1/(1-d)) = -\log(1 - \mathsf{TV}\big(q_1(\cdot|x), q_2(\cdot|x)\big))$$
$$\mathsf{TV}\big(p_k(\cdot|x), p(\cdot|x)\big) \leq 1 - \nu_k(x) \leq 1 - (1-d) = d = \mathsf{TV}\big(q_1(\cdot|x), q_2(\cdot|x)\big)$$

which are exactly the guarantees we obtained from CP-Δ for $\Delta = \Delta_{\text{max}}$. This is not a coincidence since we now show that in this case $p_k(y|x) = \min\{q_1(y|x), q_2(y|x)\}/Z(x)$. For $k = 0$ the probability of sampling y in a single iteration of while loop in smooth-CP-k is

$$\min\{p(y|x), 2^k \min_{i \in \{1,2\}} \{q_i(y|x)\}\} = \min\{q_1(y|x), 2^0 \min_{i \in \{1,2\}} \{q_i(y|x)\}\} = \min\{q_1(y|x), q_2(y|x)\}.$$

Normalizing this gives us that $p_k(y|x) = \min\{q_1(y|x), q_2(y|x)\}/Z(x)$.

□

B.1. Handling Multiple Accessing Datapoints: The $m > 1$ Case.

Recall that m denotes the number of datapoints that can access a single copyrighted data C. When $m > 1$, it may be the case that a datapoint $z \in \mathcal{D}$ has copyrighted more than one work in $\mathcal{C}$ (e.g. some training datapoint substantially contains material from two copyrighted works), so simply deduplicating $\mathcal{D}$ may not result in a dataset with $m = 1$. Hence, an algorithm for $m > 1$ may be desired. The more general sharded-safe Algorithm is presented in Algorithm 6. The algorithm sharded-safe first partitions $\mathcal{D}$ into disjoint shards $\mathcal{D}_1, \ldots \mathcal{D}_{m+1}$. By the pigeonhole principle, this ensures that each copyrighted work does not appear in at least one dataset $\mathcal{D}_i$. Of course, depending on the dataset, it may be possible to use less than $m+1$ partitions, even for the $m > 1$ case.

The guarantees for CP-k Algorithm already extend to using sharded-safe; for example by taking $p = \mathcal{A}(\mathcal{D})$, we then have a black box procedure for copyright protection using only the algorithm $\mathcal{A}$. Even though CP-k is a natural algorithm to use, it may still be conceptually worthwhile to modify the CP-Δ algorithm to handle the $m > 1$ case Here, we see show how a certain log partition function governs k_x. For $\Delta = \Delta_{\text{max}}$, CP-$\Delta$ can be modified to return:

$$p(y|x) = \frac{\min\{q_1(y|x), \ldots, q_{m+1}(y|x)\}}{Z(x)},$$

and, for $\Delta = \Delta_{\text{KL}}$, CP-$\Delta$ can be modified to return:

$$p(y|x) = \frac{\big(q_1(y|x) q_2(y|x) \ldots q_{m+1}(y|x)\big)^{1/(m+1)}}{Z(x)}.$$

This algorithm enjoys the following guarantee:

Lemma B.3 (CP-Δ, $m > 1$)**.** *Let p be the model defined above. We have that p is $\widetilde{k}_x$-NAF with respect to sharded-safe, where $\widetilde{k}_x \leq -\log Z(x)$ if $\Delta = \Delta_{max}$ and $\widetilde{k}_x \leq -(m+1)\log Z(x)$ if $\Delta = \Delta_{KL}$.*

Proof. The proof is analogous to the $m = 1$ case. First, for $\Delta = \Delta_{\text{max}}$, observe that, by construction, $p(y|x) \leq q_i(y|x)/Z(x)$ for all $y \in \mathcal{Y}$, $i \in \{1, \ldots, m+1\}$. Hence, $k_x = \max_{i,y} \log(p(y|x)/q_i(y|x)) \leq \log(1/Z(x)) = -\log Z(x)$.

For $\Delta = \Delta_{\mathrm{KL}}$, we have that

$$
\begin{aligned}
k_x &= \max_{i \in \{1,\ldots,m+1\}} \mathsf{KL}(p(\cdot|x), q_i(\cdot|x)) \\
&\leq \sum_{i=1}^{m+1} \mathsf{KL}(p(\cdot|x), q_i(\cdot|x)) \\
&= (m+1)\mathbb{E}_{i \in \{1,\ldots,m+1\}} \mathsf{KL}(p(\cdot|x), q_i(\cdot|x)) \\
&= (m+1)\mathbb{E}_{y \sim p(\cdot|x)} \log \frac{p(y|x)}{(q_1(y|x) \cdot q_2(y|x) \ldots q_{m+1}(y|x))^{1/(m+1)}} \\
&= (m+1)\mathbb{E}_{y \sim p(\cdot|x)} \log \frac{1}{Z(x)} \\
&= -(m+1) \log Z(x)
\end{aligned}
$$

□

C. Supplementary Materials for Section 4 (Experiments)

C.1. Further Discussion regarding the Diffusion Model Experiment

Training techniques to increase model similarity: Our theoretical results (Theorem 3.4) show that the bound on $\widetilde{k}$ depends on how close the underlying models q_1, q_2 and p are. To encourage model similarity during finite-sample training, we use the same values of noise in the diffusion process (while training) for all the models (ensured by using the same random seed in training q_1, q_2 and p); this does not invalidate the access-free property of the safe models because the noise sequence is chosen independently of the training images. Figure 1 displays the model generations, for p, q_1, q_2, using the same noise sequence on the diffusion paths for the corresponding images. Here, we can see that these models produce similar (but not identical) images when given the same noise sequence. The rightmost figure, which shows samples from p_k, is exactly the same images as leftmost figure, which are samples from p, except when the image fails to meet the threshold criteria, in which case the image was continually re-sampled until the threshold criteria is met.

The data-processing inequality and interpreting $\widetilde{k}$: The value of $\widetilde{k} \approx 500$ may look pessimistic at first glance. A few points are in order here. First, our guarantees (Theorems 3.3,3.4) apply to the whole sequence y rather than just to x_0, where our guarantees are on events defined on the sequences $(x_T, x_{T-1}, \ldots, x_0)$ themselves. Ultimately, we are only interested in the marginal probabilities of the images x_0, and, by the data-processing inequality, our bounds also hold directly on x_0. In particular, for any image x_0 generated by p_k, we have $p_k(x_0) \leq 2^{501} \cdot \mathsf{safe}_{x_0}(x_0)$. Part of the reason for a large value of k may be due to our inability to directly run our algorithm on the marginal probabilities of the images. This is due to the difficulty in directly computing marginal likelihoods with diffusion models[8], which requires summing over different paths y which end in x_0.

Second, it is plausible our value of $\widetilde{k}$ may be non-vacuous. Assuming the diffusion model is faithful to the CIFAR-10 distribution and that the negative log likelihood (NLL) of CIFAR-10 images concentrates, we have that $\mathsf{safe}_{x_0}(x_0) \approx 2^{-\mathbb{E}[\mathrm{NLL}]}$. As the current best estimates ((Kingma et al., 2021)) for $\mathbb{E}[\mathrm{NLL}]$ for CIFAR-10 is around $2.5 \cdot 3072 = 7680$, we have that the probability of generating a copyrighted image, as is, is $\lesssim 2^{501} \cdot 2^{-7680} \ll 2^{-7000}$. This bound is only for generating a copyrighted image verbatim.

Finally, it is sometimes the case that theoretically grounded methods work better in practice then their bounds suggest. However, that we plausibly obtain non-vacuous bounds even when running our algorithms on the full sequence along the diffusion path is encouraging.

Cross entropy losses

Algorithm	Original model	CP-$\Delta_{\max}$	CP-Δ_{KL}
350m params	3.2	3.16	3.13
125m params	3.87	3.78	3.7

Expected bounds on k_x

	$\mathbb{E}[k_x], \Delta_{\max}$	$\mathbb{E}[k_x], \Delta_{\mathrm{KL}}$	Entropy
350m	.228 (7%)	.058 (1.7%)	3.225 (100%)
125m	.508 (13%)	.166 (4.5%)	3.637 (100%)

Table 1. Top: Cross-entropy loss for the original language generation models (q_1 and q_2) vs. ones produced by our algorithm; our algorithms improve on the loss of the underlying models. Bottom: Expectation of our bounds on k_x for both $\Delta_{\max}$ and Δ_{KL} over the single-token distributions of text and prompts x from the combined training set. (In parenthesis, the expected value of k_x as a fraction of the total entropy in the token distribution.)

C.2. A Language Model Experiment

We use the C4 dataset (Raffel et al., 2019) and train decoder-only transformers similar to GPT models (specifically (Mosaic ML, 2022)) on two disjoint parts[9] in to obtain models q_1, q_2. We then transform q_1, q_2 to a model p using the CP-Δ algorithm for both $\Delta = \Delta_{\max}$ and $\Delta = \Delta_{\mathrm{KL}}$. Our motivation is to understand how the CP-Δ algorithm, used as is (on a token level), fares in terms of k_x and the model degradation. As shown in Table 1 (top), the resulting models have somewhat improved cross-entropy loss compared to each one of the original models. For $\Delta = \Delta_{\mathrm{KL}}$, this is perhaps expected since CP-Δ corresponds to a model averaging algorithm in logit space.

We also investigate the effectiveness of CP-Δ by looking at the implied k_x at the token level. Here, we look at the expected value of k_x, where x is a random prefix of the training data. We show that the expected value of k_x is significantly smaller than the total entropy of the tokens (see Table 1 (bottom).[10]). Interestingly, even the relative bounds on the expect value of k_x, compared to the total entropy, improve as the model scales up, though this should be investigated for larger models.

As a crude interpretation for these per token results, assume that for all x, (i) k_x is concentrated i.e. it is close to its expectation, and (ii) in the safe model q_i, $-\log q_i(\cdot|x)$ is bounded above by some constant factor times its unconditional expectation, i.e. bounded by $O(\mathbb{E}_{x,y\sim q_i(\cdot|x)}[-\log q_i(y|x)])$ (which is exactly the expected entropy). Then for strings of length ℓ their probability in the safe model (which could be q_1 or q_2 depending on the prompt) will scale as $\approx 2^{-\mathbb{E}[\text{entropy}]\cdot\ell}$, and we can bound the output probability in the final model by $\approx 2^{\mathbb{E}[k_x]\ell} \cdot 2^{-\mathbb{E}[\text{entropy}]\cdot\ell}$. As $\mathbb{E}[k_x] < \mathbb{E}[\text{entropy}]$ this implies that the probability of violating a copyright goes down exponentially with the length of the string. While these assumptions will not strictly hold in practice, this is a starting point to understand the effectiveness of our algorithms on language models. Further investigations can be done by directly examining the behavior of CP-k when applied to the joint probabilities on strings.

D. Prior Works

There have been several studies of copyright issues in machine learning and data mining in the law literature, though most of them focus on potential infringements in the *training phase*. Sag (2018) surveys the question of whether data mining and machine learning on copyrighted text falls under "fair use" and states that "allowing [text data mining] and other similar non-expressive uses of copyrighted works without authorization is entirely consistent with the fundamental structure of copyright law.". Sag (2018) also states that under U.S. law "extracting a short phrase or snippet of text from one work and using it in another does not amount to a reproduction of the work if the localized similarity is not substantial,

[8]Such an issue does not arise for language models since the whole path serves as the output i.e. there is no need for appealing to the data-processing inequality. It also would not arise for flow based (invertible) generative models

[9]The amount of data used to train each q_i follows the default values in (Mosaic ML, 2022), which uses the Chinchilla ((Hoffmann et al., 2022)) compute-optimal values.

[10]Theorem 3.1 uses the bound $k_x \leq \sum_{i=1}^{2} \Delta_{\mathrm{KL}}\big(p(\cdot|x) \,\|\, q_i(\cdot|x)\big)$ in the case $\Delta = \Delta_{\mathrm{KL}}$. Instead of using that we explicitly report in Table 1 (bottom) the expectation of the true quantity $k_x = \max_{i\in\{1,2\}}(\Delta_{\mathrm{KL}}\big(p(\cdot|x) \,\|\, q_i(\cdot|x)\big))$.

is not quantitatively or qualitatively significant, or is otherwise de minimis." (However, European courts have a stricter threshold for the amount of similarity.) Sobel (2018) also discusses the issue of "fair use" in training. While he mentions the issue of output generation, the article does not focus on it since (at the time) "works generated by Al are fascinating and entertaining, but today they remain novelties rather than mainstream sources of entertainment or compelling substitutes for human expression." Gillotte (2020) studies copyright infringement in AI-generated artworks and concludes that regarding the training phase "an engineer may use copyrighted works to train an Al program to generate artwork without incurring infringement liability." Hristov (2016) considers a separate issue regarding AI and copyright: whether it should be possible to grant copyright to AI-authored works. Current rulings ((Board, 2022)) by the U.S. copyright review board state that wholly AI generated works cannot be considered for copyright.

Lee et al. (2022b) show that deduplication can significantly reduce memorization, but not eliminate it (see also bottom row of Table 1 in (Kandpal et al., 2022)). Ippolito et al. (2022) state that "deduplication does not guarantee that a model will not still memorize individual (deduplicated) examples, necessitating defenses that operate at inference-time". They also show that simply stopping models from outputting training samples verbatim does *not* prevent memorization and can give a "false sense of security." (Lee et al., 2022a; Tirumala et al., 2022; Carlini et al., 2022) show that memorization becomes worse with model size and data reuse. The deterioration with growing model size holds even in the single-epoch (no data reuse) case; see in particular Figures 1 and 8 in (Tirumala et al., 2022).

As discussed in Section 2.2, our work is related to, but also substantially different than, differential privacy (Dwork et al., 2006). We discuss additional relevant works on differential privacy in Appendix A.1. Ponomareva et al. (2022) train a small (60m parameters) differentially private language model, while Li et al. (2021) fine-tune large models in a differentially private fashion. Carlini et al. (2021) show a reconstruction attack of training data from model weights for GPT-2, while very recently Carlini et al. (2023) gave training-points reconstruction attacks for diffusion models. We note that while a reconstruction attack has strong privacy implications, it does not prevent copyright protection for the generated outputs.