Data-Copying in Generative Models: A Formal Framework

Robi Bhattacharjee [1] **Sanjoy Dasgupta** [1] **Kamalika Chaudhuri** [1]

Abstract

There has been some recent interest in detecting and addressing memorization of training data by deep neural networks. A formal framework for memorization in generative models, called "data-copying" was proposed by Meehan et. al (2020). We build upon their work to show that their framework may fail to detect certain kinds of blatant memorization. Motivated by this and the theory of non-parametric methods, we provide an alternative definition of data-copying that applies more locally. We provide a method to detect data-copying, and provably show that it works with high probability when enough data is available. We also provide lower bounds that characterize the sample requirement for reliable detection.

1. Introduction

Deep generative models have shown impressive performance. However, given how large, diverse, and uncurated their training sets are, a big question is whether, how often, and how closely they are memorizing their training data. This question has been of considerable interest in generative modeling (Lopez-Paz & Oquab, 2016; Xu et al., 2018) as well as supervised learning (Brown et al., 2021; Feldman, 2020). However, a clean and formal definition of memorization that captures the numerous complex aspects of the problem, particularly in the context of continuous data such as images, has largely been elusive.

For generative models, (Meehan et al., 2020) proposed a formal definition of memorization called "data-copying", and showed that it was orthogonal to various prior notions of overfitting such as mode collapse (Thanh-Tung & Tran, 2020), mode dropping (Yazici et al., 2020), and precision-recall (Sajjadi et al., 2018). Specifically, their definition looks at three datasets – a training set, a set of generated example, and an independent test set. Data-copying happens when the training points are considerably closer on average to the generated data points than to an independently drawn test sample. Otherwise, if the training points are further on average to the generated points than test, then there is underfitting. They proposed a three sample test to detect this kind of data-copying, and empirically showed that their test had good performance.

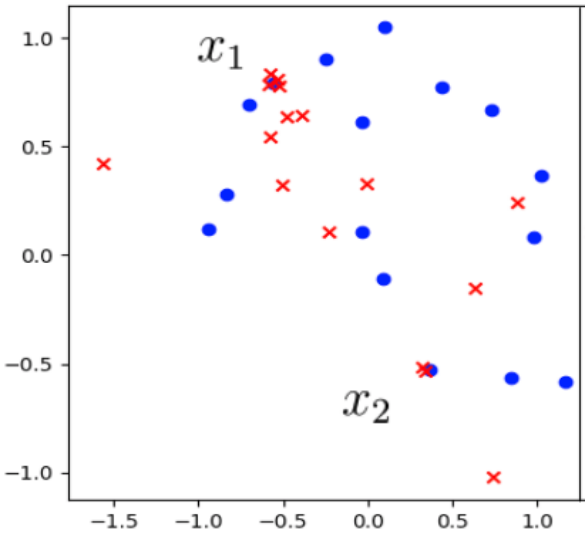


Figure 1. In this figure, the blue points are sampled from the halfmoons dataset (with Gaussian noise). The red points are sampled from a generated distribution that is a mixture of (40 %) blatant data copier (that outputs a random subset of the training set), and (60 %) a noisy underfit version of halfmoons. Although the generated distribution is clearly doing some form of copying at points x_1 and x_2, detecting this is challenging because of the canceling effect of the underfit points.

However, despite its practical success, this method may not capture even blatant cases of memorization. To see this, consider the example illustrated in Figure 1, in which a generated model for the halfmoons dataset outputs one of its training points with probability 0.4, and otherwise outputs a random point from an underfit distribution. When the test of (Meehan et al., 2020) is applied to this distribution, it is unable to detect any form of data copying; the generated samples drawn from the underfit distribution are sufficient to cancel out the effect of the memorized examples. Nevertheless, this generative model is clearly an egregious memorizer

[*]Equal contribution [1]UCSD. Correspondence to: Robi Bhattacharjee <rcbhatta@ucsd.edu>.

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

as shown in points x_1 and x_2 of Figure 1.

This example suggests a notion of *point-wise* data copying, where a model q can be thought of as copying a given training point x. Such a notion would be able to detect q's behavior nearby x_1 and x_2 regardless of the confounding samples that appear at a global level. This stands in contrast to the more global distance based approach taken in Meehan et. al. which is unable to detect such instances. Motivated by this, we propose an alternative point-by-point approach to defining data-copying.

We say that a generative model q data-copies an individual training point, x, if it has an unusually high concentration in a small area centered at x. Intuitively, this implies q is highly likely to output examples that are very similar to x. In the example above, this definition would flag q as copying x_1 and x_2.

To parlay this definition into a global measure of data-copying, we define the overall *data-copying rate* as the total fraction of examples from q that are copied from some training example. In the example above, this rate is 40%, as this is the fraction of examples that are blatant copies of the training data.

Next, we consider how to detect data-copying according to this definition. To this end, we provide an algorithm, Data_Copy_Detect, that outputs an estimate for the overall data-copying rate. We then show that under a natural smoothness assumption on the data distribution, which we call *regularity*, Data_Copy_Detect is able to guarantee an accurate estimate of the total data-copying rate. We then give an upper bound on the amount of data needed for doing so.

We complement our algorithm with a lower bound on the minimum amount of a data needed for data-copying detection. Our lower bound also implies that some sort of smoothness condition (such as regularity) is necessary for guaranteed data-copying detection; otherwise, the required amount of data can be driven arbitrarily high.

1.1. Related Work

Recently, understanding failure modes for generative models has been an important growing body of work e.g. (Salimans et al., 2016; Richardson & Weiss, 2018; Sajjadi et al., 2018). However, much of this work has been focused on other forms of overfitting, such as mode dropping or mode collapse.

A more related notion of overfitting is *memorization* (Lopez-Paz & Oquab, 2016; Xu et al., 2018; Chatterjee, 2018), in which a model outputs exact copies of its training data. This has been studied in both supervised (Brown et al., 2021; Feldman, 2020) and unsupervised (van den Burg & Williams, 2021; Bai et al., 2021) contexts. Memorization has also been considered in language generation models (Carlini et al., 2022).

The first work to explicitly consider the more general notion of *data-copying* is (Meehan et al., 2020), which gives a three sample test for data-copy detection. We include an empirical comparison between our methods in Section 5.2, where we demonstrate that ours is able to capture certain forms of data-copying that theirs is not.

Finally, we note that this work focuses on detecting natural forms of memorization or data-copying, that likely arises out of poor generalization, and is not concerned with detecting *adversarial* memorization or prompting, such as in (Carlini et al., 2019), that are designed to obtain sensitive information about the training set. This is reflected in our definition and detection algorithm which look at the specific generative model, and not the algorithm that trains it. Perhaps the best approach to prevent adversarial memorization is training the model with differential privacy (Dwork, 2006), which ensures that the model does not change much when one training sample changes. However such solutions come at an utility cost.

2. A Formal Definition of Data-Copying

We begin with the following question: what does it mean for a generated distribution q to copy a single training example x? Intuitively, this means that q is guilty of overfitting x in some way, and consequently produces examples that are very similar to it.

However, determining what constitutes a 'very similar' generated example must be done contextually. Otherwise the original data distribution, p, may itself be considered a copier, as it will output points nearby x with some frequency depending on its density at x. Thus, we posit that q data copies training point x if it has a significantly higher concentration nearby x than p does. We express this in the following definition.

Definition 2.1. Let p be a data distribution, $S \sim p^n$ a training sample, and q be a generated distribution trained on S. Let $x \in S$ be a training point, and let $\lambda > 1$ and $0 < \gamma < 1$ be constants. A generated example $x' \sim q$ is said to be a (λ, γ)**-copy** of x if there exists a ball B centered at x (i.e. $\{x' : ||x' - x|| \leq r\}$) such that following hold:

- $x' \in B$.
- $q(B) \geq \lambda p(B)$
- $p(B) \leq \gamma$

Here $q(B)$ and $p(B)$ denote the probability mass assigned to B by p and q respectively.

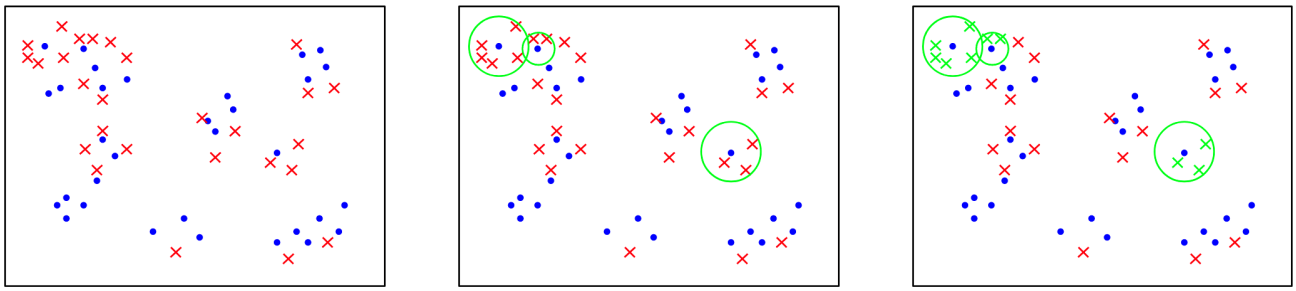


Figure 2. In the three panels above, the blue points are a training sample from p, and the red points are generated examples from q. In the middle panel, we highlight in green regions that are defined to be *data-copying regions*, as q overrepresents them with comparison to p. In the third panel, we then color all points from q that are considered to be copied green.

The parameters λ and γ are user chosen parameters that characterize data-copying. λ represents the rate at which q must overrepresent points close to x, with higher values of λ corresponding to more egregious examples of data-copying. γ represents the maximum size (by probability mass) of a region that is considered to be data-copying – the ball B represents all points that are "copies" of x. Together, λ and γ serve as practitioner controlled knobs that characterize data-copying about x.

Our definition is illustrated in Figure 2 – the training data is shown in blue, and generated samples are shown in red. For each training point, we highlight a region (in green) about that point in which the red density is much higher than the blue density, thus constituting data-copying. The intuition for this is that the red points within any ball can be thought of as "copies" of the blue point centered in the ball.

Having defined data-copying with respect to a single training example, we can naturally extend this notion to the entire training dataset. We say that $x' \sim q$ is copied from training set S if x' is a (λ, γ)-copy of some training example $x \in S$. We then define the *data-copy rate* of q as the fraction of examples it generates that are copied from S. Formally, we have the following:

Definition 2.2. Let p, S, q, λ, and γ be as defined in Definition 2.1. Then the **data-copy rate**, $cr\left(q, \lambda, \gamma\right)$ of q (with respect to p, S) is the fraction of examples from q that are (λ, γ)-copied. That is,

$$cr\left(q, \lambda, \gamma\right) = \Pr_{x' \sim q}\left[q\ (\lambda, \gamma)\text{-copies } x'\right].$$

In cases where λ, γ are fixed, we use $cr_q = cr(q, \lambda, \gamma)$ to denote the data-copy rate.

Despite its seeming global nature, cr_q is simply an aggregation of the point by point data-copying done by q over its entire training set. As we will later see, estimating cr_q is often reduced to determining which subset of the training data q copies.

2.1. Examples of data-copying

We now give several examples illustrating our definitions. In all cases, we let p be a data distribution, S, a training sample from p, and q, a generated distribution that is trained over S.

The uniform distribution over S: In this example, q is an egregious data copier that memorizes its training set and randomly outputs a training point. This can be considered as the canonical *worst* data copier. This is reflected in the value of cr_q – if p is a continuous distribution with finite probability density, then for any $x \in S$, there exists a ball B centered at x for which $q(B) >> p(B)$. It follows that q (λ, γ)- copies x for all $x \in S$ which implies that $cr_q = 1$.

The perfect generative model: $q = p$: In this case, $q(B) = p(B)$ for all balls, B, which implies that q does not perform any data-copying (Definition 2.1). It follows that $cr_q = 0$, matching the intuition that q does not data-copy at all.

Kernel Density Estimators: Finally, we consider a more general situation, where q is trained by a *kernel density estimator* (KDE) over $S \sim p^n$. Recall that a kernel density estimator outputs a generated distribution, q, with pdf defined by

$$q(x) = \frac{1}{n\sigma_n} \sum_{x_i \in S} K\left(\frac{x - x_i}{\sigma_n}\right).$$

Here, K is a kernel similarity function, and σ_n is the bandwidth parameter. It is known that for $\sigma_n = O(n^{-1/5})$, q converges towards p for sufficiently well behaved probability distributions.

Despite this guarantee, KDEs intuitively appear to perform some form of data-copying – after all they implicitly include each training point in memory as it forms a portion of their outputted pdf. However, recall that our main focus is in

understanding *overfitting* due to data-copying. That is, we view data-copying as a function of the outputted pdf, q, and not of the training algorithm used.

To this end, for KDEs the question of data-copying reduces to the question of whether q overrepresents areas around its training points. As one would expect, this occurs *before* we reach the large sample limit. This is expressed in the following theorem.

Theorem 2.3. *Let $1 < \lambda$ and $\gamma > 0$. Let σ_n be a sequence of bandwidths and K be any regular kernel function. For any $n > 0$ there exists a probability distribution π with full support over $\mathbb{R}^d$ such that with probability at least $\frac{1}{3}$ over $S \sim \pi^n$, a KDE trained with bandwidth σ_n and kernel function K has data-copy rate $cr_q \geq \frac{1}{10}$.*

This theorem completes the picture for KDEs with regards to data-copying – when n is too low, it is possible for the KDE to have a significant amount of data-copying, but as n continues to grow, this is eventually smoothed out.

The Halfmoons dataset Returning to the example given in Figure 1, observe that our definition exactly captures the notion of data-copying that occurs at points x_1 and x_2. For even strict choices of λ and γ, Definition 2.1 indicates that the red distribution copies both x_1 and x_2. Furthermore, the data-copy rate, cr_q, is 40% by construction, as this is the proportion of points that are outputted nearby x_1 and x_2.

2.2. Limitations of our definition

Definition 2.1 implicitly assumes that the goal of the generator is to output a distribution q that approaches p in a mathematical sense; a perfect generator would output q so that $q(M) = p(M)$ for all measurable sets. In particular, instances where q outputs examples that are far away from the training data are considered completely irrelevant in our definition.

This restriction prevents our definition from capturing instances in which q memorizes its training data and then applies some sort of transformation to it. For example, consider an image generator that applies a color filter to its training data. This would not be considered a data-copier as its output would be quite far from the training data in pixel space. Nevertheless, such a generated distribution can be very reasonably considered as an egregious data copier, and a cursory investigation between its training data and its outputs would reveal as much.

The key difference in this example is that the generative algorithm is no longer trying to closely approximate p with q – it is rather trying to do so in some kind of transformed space. Capturing such interactions is beyond the scope of our paper, and we firmly restrict ourselves to the case where a generator is evaluated based on how close q is to p with respect to their measures over the input space.

3. Detecting data-copying

Having defined cr_q, we now turn our attention towards *estimating it.* To formalize this problem, we will require a few definitions. We begin by defining a generative algorithm.

Definition 3.1. A **generative algorithm**, A, is a potentially randomized algorithm that outputs a distribution q over $\mathbb{R}^d$ given an input of training points, $S \subset \mathbb{R}^d$. We denote this relationship by $q \sim A(S)$.

This paradigm captures most typical generative algorithms including both non-parametric methods such as KDEs and parametric methods such as variational autoencoders.

As an important distinction, in this work we define data-copying as a property of the generated distribution, q, rather than the generative algorithm, A. This is reflected in our definition which is given solely with respect to q, S, and p. For the purposes of this paper, A can be considered an arbitrary process that takes S and outputs a distribution q. We include it in our definitions to emphasize that while S is an i.i.d sample from p, it is *not* independent from q.

Next, we define a *data-copying detector* as an algorithm that estimates cr_q based on access to the training sample, S, along with the ability to draw any number of samples from q. The latter assumption is quite typical as sampling from q is a purely computational operation. We do not assume any access to p beyond the training sample S. Formally, we have the following definition.

Definition 3.2. A **data-copying detector** is an algorithm D that takes as input a training sample, $S \sim p^n$, and access to a sampling oracle for $q \sim A(S)$ (where A is an arbitrary generative algorithm). D then outputs an estimate, $D(S, q) = \hat{cr}_q$, for the data-copy rate of q.

Naturally, we assume D has access to $\lambda, \gamma > 0$ (as these are practitioner chosen values), and by convention don't include λ, γ as formal inputs into D.

The goal of a data-copying detector is to provide accurate estimates for cr_q. However, the precise definition of cr_q poses an issue: data-copy rates for varying values of λ and γ can vastly differ. This is because λ, γ act as thresholds with everything above the threshold being counted, and everything below it being discarded. Since λ, γ cannot be perfectly accounted for, we will require some tolerance in dealing with them. This motivates the following.

Definition 3.3. Let $0 < \epsilon$ be a tolerance parameter. Then the **approximate data-copy rates**, $cr_q^{-\epsilon}$ and cr_q^{ϵ}, are defined as the values of cr_q when the parameters (λ, γ) are shifted by a factor of $(1 + \epsilon)$ to respectively decrease and

increase the copy rate. That is,

$$cr_q^{-\epsilon} = cr\left(q, \lambda(1+\epsilon), \gamma(1+\epsilon)^{-1}\right),$$

$$cr_q^{\epsilon} = cr\left(q, \lambda(1+\epsilon)^{-1}, \gamma(1+\epsilon)\right).$$

The shifts in λ and γ are chosen as above because increasing λ and decreasing γ both reduce cr_q seeing as both result in more restrictive conditions for what qualifies as data-copying. Conversely, decreasing λ and increasing γ has the opposite effect. It follows that

$$cr_q^{-\epsilon} \leq cr_q \leq cr_q^{\epsilon},$$

meaning that $cr_q^{-\epsilon}$ and cr_q^{ϵ} are lower and upper bounds on cr_q.

In the context of data-copying detection, the goal is now to estimate cr_q in comparison to $cr_q^{\pm\epsilon}$. We formalize this by defining *sample complexity* of a data-copying detector as the amount of data needed for accurate estimation of cr_q.

Definition 3.4. Let D be a data-copying detector and p be a data distribution. Let $\epsilon, \delta > 0$ be standard tolerance parameters. Then D has **sample complexity**, $m_p(\epsilon, \delta)$, with respect to p if for all $n \geq m_p(\epsilon, \delta)$, $\lambda > 1$, $0 < \gamma < 1$, and generative algorithms A, with probability at least $1 - \delta$ over $S \sim p^n$ and $q \sim A(S)$,

$$cr_q^{-\epsilon} - \epsilon \leq D(S, q) \leq cr_q^{\epsilon} + \epsilon.$$

Here the parameter ϵ takes on a somewhat expanded as it is both used to additively bound our estimation of cr_q and to multiplicatively bound λ and γ.

Observe that there is no mention of the number of calls that D makes to its sampling oracle for q. This is because samples from q are viewed as *purely computational*, as they don't require any natural data source. In most cases, q is simply some type of generative model (such as a VAE or a GAN), and thus sampling from q is a matter of running the corresponding neural network.

4. Regular Distributions

Our definition of data-copying (Definition 2.1) motivates a straightforward point by point method for data-copying detection, in which for every training point, x_i, we compute the largest ball B_i centered at x_i for which $q(B_i) \geq \lambda p(B_i)$ and $p(B_i) \leq \gamma$. Assuming we compute these balls accurately, we can then query samples from q to estimate the total rate at which q outputs within those balls, giving us our estimate of cr_q.

The key ingredient necessary for this idea to work is to be able to reliably estimate the masses, $q(B)$ and $p(B)$ for any ball in $\mathbb{R}^d$. The standard approach to doing this is through *uniform convergence*, in which large samples of points are drawn from p and q (in p's case we use S), and then the mass of a ball is estimated by counting the proportion of sampled points within it. For balls with a sufficient number of points (typically $O(d \log n)$), standard uniform convergence arguments show that these estimates are reliable.

However, this method has a major pitfall for our purpose – in most cases the balls B_i will be very small because data-copying intrinsically deals with points that are very close to a given training point. While one might hope that we can simply ignore all balls below a certain threshold, this does not work either, as the sheer number of balls being considered means that their union could be highly non-trivial.

To circumvent this issue, we will introduce an interpolation technique that estimates the probability mass of a small ball by scaling down the mass of a sufficiently large ball with the same center. While obtaining a general guarantee is impossible – there exist pathological distributions that drastically change their behavior at small scales – it turns out there is a relatively natural condition under which such interpolation will work. We refer to this condition as *regularity,* which is defined as follows.

Definition 4.1. Let $k > 0$ be an integer. A probability distribution p is **k-regular** the following holds. For all $\epsilon > 0$, there exists a constant $0 < p_\epsilon \leq 1$ such that for all x in the support of p, if $0 < s < r$ satisfies that $p(B(x, r)) \leq p_\epsilon$, then

$$\left(1 + \frac{\epsilon}{3}\right)^{-1} \frac{r^k}{s^k} \leq \frac{p(B(x, r))}{p(B(x, s))} \leq \left(1 + \frac{\epsilon}{3}\right) \frac{r^k}{s^k}.$$

Finally, a distribution is **regular** if it is k-regular for some integer $k > 0$.

Here we let $B(x, r) = \{x' : ||x - x'|| \leq r\}$ denote the closed ℓ_2 ball centered at x with radius r.

The main intuition for a k-regular distribution is that at a sufficiently small scale, its probability mass scales with distance according to a power law, determined by k. The parameter k dictates how the probability density behaves with respect to the distance scale. In most common examples, k will equal the *intrinsic dimension* of p.

As a technical note, we use an error factor of $\frac{\epsilon}{3}$ instead of ϵ for technical details that enable cleaner statements and proofs in our results (presented later).

4.1. Distributions with Manifold Support

We now give an important class of k-regular distributions.

Proposition 4.2. *Let p be a probability distribution with support precisely equal to a compact k dimensional submanifold (with or without boundary) of $\mathbb{R}^d$, M. Additionally,*

suppose that p has a continuous density function over M. Then it follows that p is k-regular.

Proposition 4.2 implies that most data distributions that adhere to some sort of manifold-hypothesis will also exhibit regularity, with the regularity constant, k, being the intrinsic dimension of the manifold.

4.2. Estimation over regular distributions

We now turn our attention towards designing estimation algorithms over regular distributions, with our main goal being to estimate the probability mass of arbitrarily small balls. We begin by first addressing a slight technical detail – although the data distribution p may be regular, this does not necessarily mean that the regularity constant, k, is known. Knowledge of k is crucial because it determines how to properly interpolate probability masses from large radius balls to smaller ones.

Luckily, estimating k turns out to be an extremely well studied task, as for most probability distributions, k is a measure of the *intrinsic dimension*. Because there is a wide body of literature in this topic, we will assume from this point that k has been correctly estimated from S using any known algorithm for doing so (for example (Block et al., 2022)). Nevertheless, for completeness, we provide an algorithm with provable guarantees for estimating k (along with a corresponding bound on the amount of needed data) in Appendix B.

We now return to the problem of $p(B(x,r))$ for a small value of r, and present an algorithm, $Est(x,r,S)$ (Algorithm 1), that estimates $p(B(x,r))$ from an i.i.d sample $S \sim p^n$.

Algorithm 1: $Est(x,r,S)$

$n \leftarrow |S|$
$b \leftarrow O\left(\frac{d \ln \frac{n}{\delta}}{\epsilon^2}\right)$
$r_* = \min\{s > 0, |S \cap B(x,s)| = b\}$.
if $r_* > r$ **then**
  Return $\frac{br^k}{nr_*^k}$
else
  Return $\frac{|T \cap B(x,r)|}{n}$

Est uses two ideas: first, it leverages standard uniform convergence results to estimate the probability mass of all balls that contain a sufficient number of training examples. This is what leads to the specific value of b that is chosen. Second, it estimates the mass of smaller balls by interpolating from its estimates from larger balls. The k-regularity assumption is crucial for this second step as it is the basis on which such interpolation is done.

Est has the following performance guarantee, which follows from standard uniform convergence bounds and the definition of k-regularity.

Proposition 4.3. *Let p be a regular distribution, and let $\epsilon > 0$ be arbitrary. Then if $n = O\left(\frac{d \ln \frac{d}{\delta \epsilon p_\epsilon}}{\epsilon^2 p_\epsilon}\right)$ with probability at least $1-\delta$ over $S \sim p^n$, for all $x \in \mathbb{R}^d$ and $r > 0$,*

$$\left(1+\frac{\epsilon}{2}\right)^{-1} \leq \frac{Est(x,r,S)}{p(B(x,r))} \leq \left(1+\frac{\epsilon}{2}\right).$$

5. A Data-copy detecting algorithm

Algorithm 2: $DataCopyDetect(S,q,m)$

1 $m \leftarrow O\left(\frac{dn^2 \ln \frac{nd}{\delta\epsilon}}{\epsilon^4}\right)$
2 Sample $T \sim q^m$
3 $\{x_1, x_2, \ldots, x_n\} \leftarrow S$
4 $\{z_1, z_2, \ldots, z_m\} \leftarrow T$
5 **for** $i = 1, \ldots, n$ **do**
6 Let $p_i(r)$ denote $Est(x_i, r, S)$
7 Let $q_i(r)$ denote $\frac{|B(x_i,r) \cap T|}{m}$
8 $radii \leftarrow \{||z - x_i|| : z \in T\} \cup \{0\}$
9 $radii \leftarrow \{r : p_i(r) \leq \gamma, r \in radii\}$
10 $r_i^* \leftarrow \max\{r : q_i(r) \geq \lambda p_i(r), r \in radii\}$
11 **end**
12 Sample $U \sim q^{20/\epsilon^2}$
13 $V \leftarrow U \cap (\bigcup_{i=1}^n B(x_i, r_i^*))$
14 Return $\frac{|V|}{|U|}$.

We now now leverage our subroutine, Est, to construct a data-copying detector, $Data_Copy_Detect$ (Algorithm 2), that has bounded sample complexity when p is a regular distribution. Like all data-copying detectors (Definition 3.2), $Data_Copy_Detect$ takes as input the training sample S, along with the ability to sample from a generated distribution q that is trained from S. It then performs the following steps:

1. (line 1) Draw an i.i.d sample of $m = O\left(\frac{dn^2 \ln \frac{nd}{\delta\epsilon}}{\epsilon^4}\right)$ points from q.
2. (lines 6 - 10) For each training point, x_i, determine the largest radius r_i for which
$$\frac{|B(x_i,r_i) \cap T|}{m} \geq \lambda Est(x_i, r_i, S),$$
$$Est(x_i, r_i, S) \leq \gamma.$$
3. (lines 12 - 13) Draw a fresh sample of points from $U \sim q^{O(1/\epsilon^2)}$, and use it to estimate the probability mass under q of $\cup_{i=1}^n B(x_i, r_i)$.

In the first step, we draw a *large* sample from q. While this is considerably larger than the amount of training data we have, we note that samples from q are considered free, and thus do not affect the sample complexity. The reason we need this many samples is simple – unlike p, q is not necessarily regular, and consequently we need enough points to properly estimate q around every training point in S.

The core technical details of $Data_Copy_Detect$ are contained within step 2, in which data-copying regions surrounding each training point, x_i, are found. We use $Est(x, r, S)$ and $\frac{|B(x,r)\cap T|}{m}$ as proxies for p and q in Definition 2.1, and then search for the maximal radius r_i over which the desired criteria of data-copying are met for these proxies.

The only difficulty in doing this is that this could potentially require checking an infinite number of radii, r_i. Fortunately, this turns out not to be needed because of the following observation – we only need to check radii at which a new point from T is included in the estimation $q_i(r)$. This is because these our estimation for $q_i(r)$ does not change between them meaning that our estimate of the ratio between q and p is maximal nearby these points.

Once we have computed r_i, all that is left is to estimate the data-copy rate by sampling q once more to find the total mass of data-copying region, $\cup_{i=1}^{n} B(x_i, r_i)$.

5.1. Performance of Algorithm 2

We now show that given enough data, $Data_Copy_Detect$ provides a close approximation of cr_q.

Theorem 5.1. *$Data_Copy_Detect$ is a data-copying detector (Definition 3.2) with sample complexity at most*

$$m_p(\epsilon, \delta) = O\left(\frac{d \ln \frac{d}{\delta \epsilon p_\epsilon}}{\epsilon^2 p_\epsilon}\right),$$

for all regular distributions, p.

Theorem 2 shows that our algorithm's sample complexity has standard relationships with the tolerance parameters, ϵ and δ, along with the input space dimension d. However, it includes an additional factor of $\frac{1}{p_\epsilon}$, which is a distribution specific factor measuring the regularity of the probability distribution. Thus, our bound cannot be used to give a bound on the amount of data needed without having a bound on p_ϵ.

We consequently view our upper bound as more akin to a convergence result, as it implies that our algorithm is guaranteed to converge as the amount of data goes towards infinity.

5.2. Applying Algorithm 2 to Halfmoons

We now return to the example presented in Figure 3 and empirically investigate the following question: is our algorithm able to outperform the one given in (Meehan et al., 2020) over this example?

To investigate this, we test both algorithms over a series of distributions by varying the parameter ρ, which is the proportion of points that are "copied." Figure 3 demonstrates a case in which $\rho = 0.4$. Additionally, we include a parameter, c, for (Meehan et al., 2020)'s algorithm which represents the number of clusters the data is partitioned into (with c-means clustering) prior to running their test. Intuitively, a larger number of clusters means a better chance of detecting more localized data-copying.

The results are summarized in the following table where we indicate whether the algorithm determined a statistically significant amount of data-copying over the given generated distribution and corresponding training dataset. Full experimental details can be found in Sections A and A.3 of the appendix.

Table 1. Statistical Significance of data-copying Rates over Halfmoons

Algo	$\mathbf{q = p}$	$\rho = 0.1$	**0.2**	**0.3**	**0.4**
Ours	no	yes	yes	yes	yes
$\mathbf{c = 1}$	no	no	no	no	no
$\mathbf{c = 5}$	no	no	no	no	yes
$\mathbf{c = 10}$	no	no	no	no	yes
$\mathbf{c = 20}$	no	no	no	yes	yes

As the table indicates, our algorithm is able to detect statistically significant data-copying rates in all cases it exists. By contrast, (Meehan et al., 2020)'s test is only capable of doing so when there is a large data-copy rate and when the number of clusters, c, is quite large.

6. Is smoothness necessary for data copying detection?

Algorithm 2's performance guarantee requires that the input distribution, p, be regular (Definition 4.1). This condition is essential for the algorithm to successfully estimate the probability mass of arbitrarily small balls. Additionally, the parameter, p_ϵ, plays a key role as it serves as a measure of how "smooth" p is with larger values implying a higher degree of smoothness.

This motivates a natural question – can data copying detection be done over unsmooth data distributions? Unfortunately, the answer turns out to be no. In the following result, we show that if the parameter, p_ϵ is allowed to be arbitrarily

small, then this implies that for any data-copy detector, there exists p for which the sample complexity is arbitrarily large.

Theorem 6.1. *Let B be a data-copying detector. Let $\epsilon = \delta = \frac{1}{3}$. Then, for all integers $\kappa > 0$, there exists a probability distribution p such that $\frac{1}{9\kappa} \leq p_\epsilon \leq \frac{1}{\kappa}$, and $m_p(\epsilon, \delta) \geq \kappa$, implying that*

$$m_p(\epsilon, \delta) \geq \Omega\left(\frac{1}{p_\epsilon}\right).$$

Although Theorem 6.1 is restricted to regular distributions, it nevertheless demonstrates that a bound on smoothness is essential for data copying detection. In particular, non-regular distributions (with no bound on smoothness) can be thought of as a degenerate case in which $p_\epsilon = 0$.

Additionally, Theorem 6.1 provides a lower bound that complements the Algorithm 2's performance guarantee (Theorem 5.1). Both bounds have the same dependence on p_ϵ implying that our algorithm is optimal at least in regards to p_ϵ. However, our upper bound is significantly larger in its dependence on d, the ambient dimension, and ϵ, the tolerance parameter itself.

While closing this gap remains an interesting direction for future work, we note that the existence of a gap isn't too surprising for our algorithm, $Data_Copy_Detect$. This is because $Data_Copy_Detect$ essentially relies on manually finding the entire region in which data-copying occurs, and doing this requires precise estimates of p at all points in the training sample.

Conversely, detecting data-copying only requires an *overall* estimate for the data-copying rate, and doesn't necessarily require finding all of the corresponding regions. It is plausible that more sophisticated techniques might able to estimate the data-copy rate *without* directly finding these regions.

7. Conclusion

In conclusion, we provide a new modified definition of "data-copying" or generating memorized training samples for generative models that addresses some of the failure modes of previous definitions (Meehan et al., 2020). We provide an algorithm for detecting data-copying according to our definition, establish performance guarantees, and show that at least some smoothness conditions are needed on the data distribution for successful detection.

With regards to future work, one important direction is in addressing the limitations discussed in section 2.2. Our definition and algorithm are centered around the assumption that the goal of a generative model is to output q that is close to p in a mathematical sense. As a result, we are unable to handle cases where the generator tries to generate *transformed* examples that lie outside the support of the training distribution. For example, a generator restricted to outputting black and white images (when trained on color images) would remain completely undetected by our algorithm regardless of the degree with which it copies its training data. To this end, we are very interested in finding generalizations of our framework that are able to capture such broader forms of data-copying.

Acknowledgments

We thank NSF under CNS 1804829 for research support.

References

Bai, C., Lin, H., Raffel, C., and Kan, W. C. On training sample memorization: Lessons from benchmarking generative modeling with a large-scale competition. In Zhu, F., Ooi, B. C., and Miao, C. (eds.), *KDD '21: The 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, Singapore, August 14-18, 2021*, pp. 2534–2542. ACM, 2021. doi: 10.1145/3447548.3467198. URL https://doi.org/10.1145/3447548.3467198.

Block, A., Jia, Z., Polyanskiy, Y., and Rakhlin, A. Intrinsic dimension estimation using wasserstein distances. *Journal of machine learning research*, 1533-7928, 2022.

Brown, G., Bun, M., Feldman, V., Smith, A., and Talwar, K. When is memorization of irrelevant training data necessary for high-accuracy learning? In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*. ACM, jun 2021. doi: 10.1145/3406325.3451131. URL https://doi.org/10.1145%2F3406325.3451131.

Carlini, N., Liu, C., Erlingsson, Ú., Kos, J., and Song, D. The secret sharer: Evaluating and testing unintended memorization in neural networks. In Heninger, N. and Traynor, P. (eds.), *28th USENIX Security Symposium, USENIX Security 2019, Santa Clara, CA, USA, August 14-16, 2019*, pp. 267–284. USENIX Association, 2019.

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.

Chatterjee, S. Learning and memorization. In Dy, J. and Krause, A. (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 755–763. PMLR, 10–15 Jul 2018.

Dwork, C. Differential privacy. In Bugliesi, M., Preneel, B., Sassone, V., and Wegener, I. (eds.), *Automata,*

Languages and Programming, 33rd International Colloquium, ICALP 2006, Venice, Italy, July 10-14, 2006, Proceedings, Part II, volume 4052 of *Lecture Notes in Computer Science*, pp. 1–12. Springer, 2006.

Feldman, V. Does learning require memorization? a short tale about a long tail. In Makarychev, K., Makarychev, Y., Tulsiani, M., Kamath, G., and Chuzhoy, J. (eds.), *Proccedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing, STOC 2020, Chicago, IL, USA, June 22-26, 2020*, pp. 954–959. ACM, 2020. doi: 10.1145/3357713.3384290. URL https://doi.org/10.1145/3357713.3384290.

Lopez-Paz, D. and Oquab, M. Revisiting classifier two-sample tests. *arXiv preprint arXiv:1610.06545*, 2016.

Meehan, C., Chaudhuri, K., and Dasgupta, S. A three sample hypothesis test for evaluating generative models. In Chiappa, S. and Calandra, R. (eds.), *The 23rd International Conference on Artificial Intelligence and Statistics, AISTATS 2020, 26-28 August 2020, Online [Palermo, Sicily, Italy]*, volume 108 of *Proceedings of Machine Learning Research*, pp. 3546–3556. PMLR, 2020.

Richardson, E. and Weiss, Y. On gans and gmms. In Bengio, S., Wallach, H. M., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pp. 5852–5863, 2018.

Sajjadi, M. S. M., Bachem, O., Lucic, M., Bousquet, O., and Gelly, S. Assessing generative models via precision and recall. In Bengio, S., Wallach, H. M., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pp. 5234–5243, 2018.

Salimans, T., Goodfellow, I. J., Zaremba, W., Cheung, V., Radford, A., and Chen, X. Improved techniques for training gans. *CoRR*, abs/1606.03498, 2016. URL http://arxiv.org/abs/1606.03498.

Thanh-Tung, H. and Tran, T. Catastrophic forgetting and mode collapse in gans. In *2020 International Joint Conference on Neural Networks, IJCNN 2020, Glasgow, United Kingdom, July 19-24, 2020*, pp. 1–10. IEEE, 2020. doi: 10.1109/IJCNN48605.2020.9207181. URL https://doi.org/10.1109/IJCNN48605.2020.9207181.

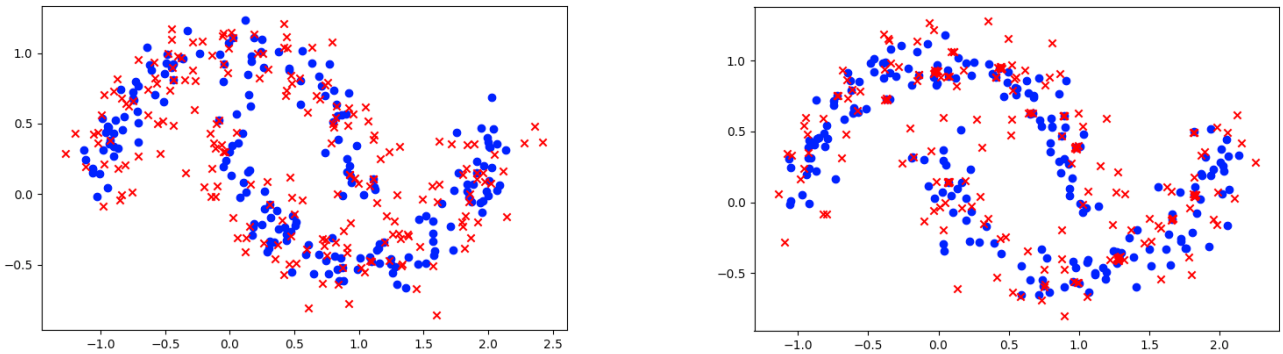
van den Burg, G. and Williams, C. On memorization in probabilistic deep generative models. In Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 27916–27928. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper/2021/file/eae15aabaa768ae4a5993a8a4f4fa6e4-Paper.pdf.

Xu, Q., Huang, G., Yuan, Y., Guo, C., Sun, Y., Wu, F., and Weinberger, K. Q. An empirical study on evaluation metrics of generative adversarial networks. *CoRR*, abs/1806.07755, 2018. URL http://arxiv.org/abs/1806.07755.

Yazici, Y., Foo, C., Winkler, S., Yap, K., and Chandrasekhar, V. Empirical analysis of overfitting and mode drop in gan training. In *IEEE International Conference on Image Processing, ICIP 2020, Abu Dhabi, United Arab Emirates, October 25-28, 2020*, pp. 1651–1655. IEEE, 2020. doi: 10.1109/ICIP40778.2020.9191083. URL https://doi.org/10.1109/ICIP40778.2020.9191083.

A. An Example over the Halfmoons dataset

In this section, we give an overview of our experiments over the Halfmoons dataset. Further details can be found in sec



(a) $\rho = 0.1$ (b) $\rho = 0.4$

Figure 3. In the two panels above, the blue points are a training sample from p, and the red points are generated examples from q. The parameter ρ is the proportion of examples of q that are generated by q_{copy}, with the rest of the examples being drawn from $q_{underfit}$. As ρ increases, the rate of data-copying increases, which can be seen as the red points become increasingly clustered on top of a scattering of blue ones. However, due to $q_{underfit}$, there are still many red points that are relatively scattered from the blue points. At a global level, these effects average out making data-copying detection difficult for (Meehan et al., 2020)'s method.

Our theoretical results show that given enough data, Algorithm 2 is guaranteed to detect data-copying. By contrast, the non-parametric test provided in (Meehan et al., 2020) can only guarantee detection in cases in which data-copying *globally* occurs. For more local instances of data-copying, they rely on k-means clustering to partition the input space into localized regions, and then run their global test over each region separately.

Their approach clearly cannot detect all forms of data-copying – a pathological generative distribution might copy in complex regions that are impossible to find using k-means clustering. However, for many practical examples considered in their paper, (Meehan et al., 2020) demonstrated considerable success with this approach.

This motivates the following question:

> Do there exist natural data distributions over which Algorithm 2 offers a meaningful advantage?

We provide a partial answer to this question by experimentally comparing our approach with (Meehan et al., 2020)'s over a simple example on the half moons dataset.

A.1. Experimental Setup

Data Distribution: Our data distribution, p, is the Halfmoon dataset with Gaussian noise ($\sigma = 0.1$).

Generated Distribution: Our generated distribution, q, is trained from an i.i.d sample of 2000 points from p, $S \sim p^{2000}$. Because our focus is on distinguishing *data-copy detection* algorithms, we design q to have a large amount of data-copying that is nevertheless subtle to detect. The key idea is to let q be a mixture of two distributions, q_{copy} and $q_{underfit}$. q_{copy} will be an egregious data copier, and $q_{underfit}$ will be designed to average away the effects of q_{copy}.

To construct q_{copy}, we first select a subset, $S' \subset S$, of 20 training examples. Then, we define q_{copy} to randomly output points from S' combined with a small amount of spherical noise (with radius 0.02). Thus, q_{copy} can be sampled from by sampling a point, x, from S' at uniform, and returning $x + \eta$ where η is drawn at uniform from a disk of radius 0.02.

To construct $q_{underfit}$, we combine our original data distribution, p, with a moderate amount of spherical noise (with radius 0.25). Thus, $q_{underfit}$ can be sampled from by first sampling $x \sim p$, and returning $x + \eta$ where η is drawn at uniform from a disk of radius 0.25. This distribution is meant to represent a fairly noisy and thus underfit version of p.

Finally, we define q as a mixture of q_{copy} and $q_{underfit}$, with q outputting a point from q_{copy} with probability ρ. In total, we have

$$q = \rho \cdot q_{copy} + (1 - \rho) \cdot q_{good}.$$

We let, ρ, the weight of q_{copy} within the mixture, be a varying parameter that gives rise to different generated distributions. Intuitively, the larger ρ is, the higher the data-copying rate. This is illustrated in Figure 3. In the both panels, we plot a sample of 200 training points p along with 200 points from q. In the left panel, we let $\rho = 0.1$ in the right, we use $\rho = 0.4$. Although both cases show examples of data-copying, the right panel shows a visibly higher level of it. This is expected, as it is drawn from a distribution in which q_{copy} is much more likely to be queried.

Data-copying Detection Algorithms: We run our algorithm, Data_Copy_Detect, on (S, q), We fix $\lambda = 20$ and $\gamma = 0.00025$ as constants for data-copy detection. λ represents a healthy level of data-copying, and $\gamma = 0.00025$ ensures that our condition for 'copying' is quite stringent. Full details of our implementation (including our practical choices for parameters such as b and m) are given in Appendix 5.2.

For comparison, we also include an implementation of (Meehan et al., 2020)'s algorithm with varying amounts of clusters being used for the initial k-means clustering. To avoid confusion with the intrinsic dimension, k, we let c denote the number of clusters, and consider $c \in \{1, 5, 10, 20\}$.

A.2. Results

The results are summarized in Table 2, with each column corresponding to a given choice of p, q (determined by the parameter ρ), and each row corresponding to a separate data-copying detection algorithm. As a baseline, we include the case where $q = p$ (meaning we have a perfect generated distribution) in the first column.

We run our algorithm with parameters λ and γ fixed as 20 and 0.00025 in all cases. For (Meehan et al., 2020)'s algorithm, we consider their data-copy detection score over the most egregious cluster.

Although our algorithm outputs real number estimates of the true data-copying rate, cr_q, (Meehan et al., 2020)'s algorithm outputs a score indicating the statistical significance of their metric under a null hypothesis of no data-copying occurring. To facilitate a simple comparison between our methods, for all algorithms, we simply output a simple yes or no to indicate whether our results were statistically significant up to the $p = 0.05$ level. We include full results of our experiments along with several extensions (with varying parameters) in section A.3.

As expected, neither of our algorithms detect data-copying on the baseline, $q = p$. However, in all other cases, our algorithm successfully detects data-copying. On the other hand, for the smaller values of ρ, (Meehan et al., 2020)'s does not. Their algorithm is only able to achieve detection when the weight of $\rho = 0.4$, and even in this case they are unable to consistently do so.

These results match the simple intuition of our algorithms. As seen in Figure 3, the red data is sometimes very close to the blue data (when it comes from q_{copy}) but at other times fairly distant (when it comes from $q_{underfit}$). These effects have a strong canceling effect in (Meehan et al., 2020)'s test. However, our test is able to adjust for this by considering each training example separately.

Table 2. Statistical Significance of data-copying Rates over Halfmoons

Algo	$\mathbf{q = p}$	$\rho = \mathbf{0.1}$	**0.2**	**0.3**	**0.4**
Ours	no	yes	yes	yes	yes
$\mathbf{c = 1}$	no	no	no	no	no
$\mathbf{c = 5}$	no	no	no	no	yes
$\mathbf{c = 10}$	no	no	no	no	yes
$\mathbf{c = 20}$	no	no	no	yes	yes

A.3. Further Experimental Details

We begin by reviewing the definitions of p and q. p is the Halfmoons dataset with Gaussian noise $(\sigma = 0.1)$. To define q, we have a mixture of two distributions, q_{copy} and $q_{underfit}$, which are defined as follows.

We draw $S \sim p^{2000}$ i.i.d, and then randomly select $S' \subset S$ with $|S'| = 20$. These points will form a basis for the support of q_{copy}. To sample $x \sim q_{copy}$, we take the following two steps.

1. Sample $z \sim S'$ at uniform.
2. Sample $\eta \sim U(B(0, 0.02))$, where $U(B(0, r))$ denotes the uniform distribution over the ball of radius r.
3. Output $x = z + \eta$.

q_{copy} can be thought of as an egregious data memorizer that injects a small amount of noise to give its inputs some (paltry) variety.

By contrast, to sample $x \sim q_{underfit}$, we do the following:

1. Sample $z \sim p$.
2. Sample $\eta \sim U(B(0, 0.25))$.
3. Output $x = z + \eta$.

In this case, the larger amount of noise serves to induce *underfitting*, in which q_{copy} does not assign the support of p enough probability mass.

Finally, to sample from q, we do the following.

1. With probability ρ, sample $x \sim q_{copy}$.
2. With probability $1 - \rho$, sample $x \sim q_{underfit}$.

(Meehan et al., 2020)'s test: Their test works as follows. Let S denote the original training sample, Q denote a sample of generated examples, with $Q \sim q^n$, and P denote a fresh set of test examples, with $P \sim p^n$. They then check to see if Q is systematically closer to S than P, (thus suggesting data copying). To do so, they use a statistical test as follows:

1. Let $S = \{x_1, x_2, \ldots, x_n\}$, $P = \{y_1, y_2, \ldots, y_n\}$, $Q = \{z_1, z_2, \ldots, z_n\}$.
2. Let Δ denote the number of pairs (i, j) for which $d(y_i, S) < d(z_j, S)$. A large value of Δ indicates that a *small* amount of data copying, as it implies that Q is further from S than P. A small value of Δ indicates a **large** amount of data copying.
3. Reflecting this, let $Z = \frac{\Delta - \frac{n^2}{2}}{\sqrt{\frac{n^2(2n+1)}{12}}}$. This gives a Z-score of Δ. (Meehan et al., 2020) show that, $p = q$, then the probability of results as significant as $Z < -5$ would be at most the probability of getting a -5σ event when sampling from a Gaussian. We use $Z < -3$ to indicate *statistically significant results*, and output the corresponding P-values ($P = 0.0027$ being significant) in our results.

Finally, to account for data copying occurring within specific regions, (Meehan et al., 2020) perform a preprocessing step in which they cluster the training data, S into c regions using k-means clustering. They then run their test separately on each region by assigning points from P and Q into the regions containing them. We output the *lowest* Z-score over any region, and vary the number of clusters with $c = 1, 5, 10, 20$.

Our test: We run Algorithm 2 with input (S, q) with a few adjustments.

1. We directly set $m = 200,000$. While the theoretical value of m is significantly higher (growing $O(n^2)$), we note that this is primarily done for achieving theoretical guarantees. In practice, often a much lower amount of data is needed.
2. For $Est(x, r, S)$, we set $b = 400$, which is a bit lower than the theoretically predicted value. As for m, we do this because for practical (and well-behaved) datasets, $Est(x, r, S)$ converges much more quickly than theory suggests.

3. We set $\lambda = 20$ and $\gamma = \frac{1}{4000}$, giving relatively stringent conditions on data copying.

Finally, our test outputs, $\hat{cr}_q$, which is an estimate of the data copy rate. Technically, any non-zero of $\hat{cr}_q$ indicates a degree of data copying. To facilitate a more direct comparison with (Meehan et al., 2020), we convert our results into statistical tests by doing the following.

1. We compute $\hat{cr}_p$, which is an estimate for the data copying rate when the generated distribution exactly equals p over 1000 different instances (each instance corresponding to a freshly drawn training set S).
2. We then compute $\hat{cr}_q$ when q is as above.
3. We finally output the fraction of the time that $\hat{cr}_p > \hat{cr}_q$, thus giving us a P-value by giving us the rate at which the null-hypothesis gives results as significant as those that we observe.

Results: We give a more complete version of Table 2, with the P-values themselves being outputted in the table. For consistency, we output the median P-value obtained over 10 runs for each experiment.

Table 3. P-values of data-copying Rates over Halfmoons

Algo	$\mathbf{q = p}$	$\rho = \mathbf{0.1}$	**0.2**	**0.3**	**0.4**
Ours	1.000	0.000	0.000	0.000	0.000
$\mathbf{c = 1}$	0.5412	1.000	1.000	0.858	0.026
$\mathbf{c = 5}$	0.113	0.976	0.780	0.081	0.007
$\mathbf{c = 10}$	0.090	0.814	0.294	0.013	0.000
$\mathbf{c = 20}$	0.035	0.279	0.093	0.005	0.000

We also remark that the computed data-copying rates by our algorithm exactly match the value of ρ in all cases (up to 3 decimal points).

B. Estimating k

The main idea of our method is to simply pick any point x_i in the training sample, $S = \{x_1, x_2, \ldots, x_n\}$, choose two small balls centered at x_i, and then measure the ratio of their probability masses as well as their radii. For sufficiently small balls, these ratios will be related by a power of k, and we can consequently just solve for an estimate of k, $\hat{k}$. Finally, since for our purposes it is extremely important that our estimate be *exactly* correct, we round $\hat{k}$ to the nearest integer. While this clearly fails in cases that k is not an integer, for most distributions k precisely equals the dimension of the underlying data manifold (see for example Proposition 4.2). These steps are enumerated in the following algorithm, $Estimate_k(S)$.

Algorithm 3: $Estimate_k(S)$

$n \leftarrow |S|$
Pick $x \in S$ arbitrarily.
$b \leftarrow \frac{64(d+2)\ln\frac{16n}{\delta}}{\epsilon^2}$.
$r_* = \min\{r : |S \cap B(x,r)| = 2b\}$.
$s_* = \min\{s : |S \cap B(x,s)| = b\}$
$\hat{k} = round\left(\frac{1}{\log_2 \frac{r_*}{s_*}}\right)$
Return $\hat{k}$.

We now give sufficient conditions under which Algorithm 3 successfully recovers k.

Proposition B.1. *Let p be an k-regular distribution, and let $\delta > 0$ be arbitrary. Let $\phi = \frac{1}{2k}$. Then there exists a constant C such that if*

$$n \geq C \frac{d \ln \frac{d}{\delta \phi p_\phi}}{\phi^2 p_\phi},$$

with probability at least $1-\delta$ *over* $S \sim p^n$, $Estimate_k(S) = k$.

Proof. We begin by first applying standard uniform convergence over ℓ_2 balls in $\mathbb{R}^d$ (which have a VC dimension of at most $d+2$). To this end, let

$$\beta_n = \sqrt{\frac{4(d+2)\ln\frac{16n}{\delta}}{n}}.$$

Then by the standard result of Vapnik and Chervonenkis, with probability $1-\delta$ over $S \sim p^n$, for all $x \in \mathbb{R}^d$ and all $r > 0$,

$$\frac{|S \cap B(x,r)|}{n} - \beta_n\sqrt{\frac{|S \cap B(x,r)|}{n}} \leq p(B(x,r)) \leq \frac{|S \cap B(x,r)|}{n} + \beta_n^2 + \beta_n\sqrt{\frac{|S \cap B(x,r)|}{n}}. \tag{1}$$

Next, assume that

$$n \geq \frac{1776(d+2)\ln\left(\frac{28416(d+2)}{\delta\phi^2 p_\phi}\right)}{\phi^2 p_\phi}. \tag{2}$$

It is clear that for an appropriate constant, we have $n = O\left(\frac{d\ln\frac{d}{\delta\phi p_\phi}}{\phi^2 p_\phi}\right)$. Thus, it suffices to show that if Equation 1 holds, then $\hat{k} = k$ (as the former holds with probability $1-\delta$ over S). We now show the following claim.

Claim: Let $r > 0$ be any radius with $|S \cap B(x,r)| \geq b$. Then

$$\left(1+\frac{\phi}{9}\right)^{-1} \leq \frac{|S \cap B(x,r)|}{np(B(x,r))} \leq \left(1+\frac{\phi}{9}\right).$$

Proof. From the definition of b, we have that

$$\frac{b}{n} = \frac{400(d+2)\ln\frac{16n}{\delta}}{n\phi^2} = \frac{100\beta_n^2}{\phi^2}. \tag{3}$$

Let $c = \sqrt{\frac{b'}{n\beta_n^2}}$. Then $b' \geq b$ implies that $c \geq \frac{10}{\phi}$. It follows that

$$\frac{c+1}{c^2} \leq \frac{1}{c-1} \leq \frac{\phi}{9}. \tag{4}$$

Substituting Equations 3 and 4 into Equation 1, we have

$$\begin{aligned}\frac{b'}{np(B(x,r))} &\geq \frac{\frac{b'}{n}}{\frac{b'}{n} + \beta_n^2 + \beta_n\sqrt{\frac{k'}{n}}} \\ &= \frac{c^2}{c^2+1+c} \\ &= \left(1+\frac{c+1}{c^2}\right)^{-1} \\ &\geq \left(1+\frac{\phi}{9}\right)^{-1}\end{aligned} \tag{5}$$

and

$$\begin{aligned}\frac{b'}{np(B(x,r))} &\leq \frac{\frac{b'}{n}}{\frac{b'}{n} - \beta_n\sqrt{\frac{b'}{n}}} \\ &= \frac{c^2}{c^2-c} \\ &= 1+\frac{1}{c-1} \\ &\leq 1+\frac{\phi}{9},\end{aligned} \tag{6}$$

Together, Equations 5 and 6 imply our claim. □

We now return to the proof of Proposition 4.3. We now show that $p(B(x, s_*) \leq p(B(x, r_*)) \leq p_\phi$. To do so, we first bound β_n^2 as follows. We have,

$$\begin{aligned}
\beta_n^2 &= \frac{4(d+2)\ln(16n/\delta)}{n} \\
&= 4(d+2)\ln\left(\frac{28416(d+2)}{\delta\phi^2 p_\phi}\ln\left(\frac{28416(d+2)}{\delta\phi^2 p_\phi}\right)\right)\frac{\phi^2 p_\phi}{1776(d+2)\ln\left(\frac{28416(d+2)}{\delta\phi^2 p_\phi}\right)} \\
&\leq 8(d+2)\ln\left(\frac{28416(d+2)}{\delta\phi^2 p_\phi}\right)\frac{\phi^2 p_\phi}{1776(d+2)\ln\left(\frac{28416(d+2)}{\delta\phi^2 p_\phi}\right)} \\
&= \frac{p_\phi \phi^2}{222}.
\end{aligned} \tag{7}$$

Next, by Equations 1 and 7 along with the fact that $b = \frac{100\beta_n^2}{\phi^2}$ (Equation 3) that

$$\begin{aligned}
p(B(x, r_*)) &\leq \frac{|S \cap B(x, r_*)|}{n} + \beta_n^2 + \beta_n\sqrt{\frac{|S \cap B(x, r_*)|}{n}} \\
&= \frac{2b}{n} + \beta_n^2 + \beta_n\sqrt{\frac{2b}{n}} \\
&= \beta_n^2\left(\frac{200}{\phi^2} + 1 + \frac{20}{\phi}\right) \\
&\leq \frac{p_\phi\phi^2}{222}\frac{221}{\phi^2} = p_\phi.
\end{aligned}$$

It follows from Definition 4.1 that

$$\left(1 + \frac{\phi}{3}\right)^{-1}\frac{p(B(x, r_*))}{p(B(x, s_*))} \leq \frac{r_*^k}{s_*^k} \leq \left(1 + \frac{\phi}{3}\right)\frac{p(B(x, r_*))}{p(B(x, s_*)}. \tag{8}$$

However, $|S \cap B(x, s_*)| = b$ and $|S \cap B(x, r_*)| = 2b$, which means that we can safely apply our claim to both of these cases. By substituting Equations 5 and 6 (for both r_*, s_*) into Equation 8, along with the fact that $\left(1 + \frac{\phi}{3}\right)\left(1 + \frac{\phi}{9}\right) \leq \left(1 + \frac{\phi}{2}\right)$, it follows that

$$\left(1 + \frac{\phi}{2}\right)^{-1} \leq \frac{r_*^k}{2s_*^k} \leq \left(1 + \frac{\phi}{2}\right) \tag{9}$$

Finally, by taking logs of Equation 9 and simplifying, we have that

$$\frac{k}{1 + \log_2\left(1 + \frac{\phi}{2}\right)} \leq \frac{1}{\log_2\frac{r_*}{s_*}} \leq \frac{k}{1 - \log_2\left(1 + \frac{\phi}{2}\right)}$$

It consequently suffices to show that k is the unique integer between $\frac{k}{1+\log_2(1+8\epsilon)}$ and $\frac{k}{1-\log_2(1+2\epsilon)}$. However, this is simply a result of the assumption that $\phi = \frac{1}{2k}$ and standard manipulations, which completes the proof. □

C. Proofs

All proofs to theorems and propositions in the main body are in this section. For each result, we include a restatement for convenience.

C.1. Proof of Theorem 2.3

We prove a stronger version of Theorem 2.3.

Theorem C.1 (Theorem 2.3)**.** *Let $1 < \lambda$ and $\gamma > 0$. Let σ_n be a sequence of bandwidths and K be any regular kernel function. For any $n > 0$ there exists a probability distribution π with full support over $\mathbb{R}^d$ such for any $S \sim \pi^n$, a KDE trained with bandwidth σ_n and kernel function K has data-copy rate $cr_q \geq \frac{1}{2}$.*

We begin by giving necessary conditions for a kernel K to be regular.

Definition C.2. A kernel function, $K : \mathbb{R}^d \to \mathbb{R}_{\geq 0}$ is regular if it satisfies the following conditions.

1. K is radially symmetric. That is, there exists $h : \mathbb{R} \to \mathbb{R}$ such that $K(x) = h(||x||)$.
2. K is regularized. That is, $\int_{\mathbb{R}^d} K(x)dx = 1$.
3. K decays to 0. That is, $\lim_{t\to\infty} h(t) = \lim_{t\to-\infty} h(t) = 0$.

It is well known that under suitable choices of σ_n and several technical assumptions that a regular KDE converges towards the true data distribution in the large sample limit. We now prove Theorem 2.3.

Proof. Fix any n, and for convenience let denote σ_n by σ. Because K is non-negative, by condition 2. of Definition C.2, there exists $R > 0$ such that $\int_{||x||\leq R} K(x)dx = \frac{1}{2}$. Let

$$D = R\sigma \left(\max \left(2n\lambda, \frac{1}{\gamma} \right) \omega_d \right)^{1/d},$$

where ω_d denotes the volume of the unit ball in d dimensions. We let π denote the uniform distribution over $[0, D]^d$, and claim that this suffices.

Let $S \sim \pi^n$ be a training sample, with $S = \{x_1, x_2, \ldots, x_n\}$, and let q be a KDE trained from S with bandwidth σ and kernel function K. Suppose $x \sim q$ satisfies that $x \in B(x_i, R\sigma)$. We claim that q (λ, γ)-copies x.

To see this, it suffices to bound $\pi((B(x_i, R\sigma))$ and $q(B(x_i, R\sigma))$. The former quantity satisfies

$$\begin{aligned}
\pi((B(x_i, R\sigma)) &\leq \frac{vol(B(x_i, R\sigma))}{D^d} \\
&= \frac{\omega_d (R\sigma)^d}{D^d} \\
&= \frac{1}{\max\left(2n\lambda, \frac{1}{\gamma}\right)} \\
&\leq \min\left(\gamma, \frac{1}{2n\lambda}\right),
\end{aligned}$$

which implies that the third condition of Definition 2.1 is met. Meanwhile, $q((B(x_i, R\sigma))$ can be bounded as

$$\begin{aligned}
q((B(x_i, R\sigma)) &= \int_{B(x_i, R\sigma)} \frac{1}{n\sigma} \sum_{j=1}^{n} K\left(\frac{x - x_j}{\sigma}\right) dx \\
&\geq \int_{B(x_i, R\sigma)} \frac{1}{n\sigma} K\left(\frac{x - x_i}{\sigma}\right) dx \\
&= \int_{||u||\leq R} \frac{1}{n} K(u) du \\
&\geq \frac{1}{2n},
\end{aligned}$$

which implies that $q((B(x_i, R\sigma)) \geq \lambda p(B(x_i, R\sigma))$ giving the second condition of Definition 2.1. Thus, it follows that q (λ, γ)-copies all $x \in B(x_i, R\sigma)$. It consequently suffices to bound $q\left(\bigcup_{i=1}^{n} B(x_i, R\sigma)\right)$.

To do so, let η denote the probability distribution over $\mathbb{R}^d$ with probability density function $\eta(x) = \frac{1}{\sigma}K(\frac{x}{\sigma})$, and let $\hat{q}$ denote the probability density function induced by the following random process:

1. Select $1 \leq i \leq n$ at uniform.
2. Select $x \sim \eta$.
3. Output $x + x_i$.

The key observation is that $\hat{q}$ has precisely the same density function as q – qs density function is clearly a convolution of selecting x_i and then adding $x \sim \eta$. Applying this, we have

$$\begin{aligned}
\Pr_{x\sim q}\left[x \in \bigcup_{i=1}^{n} B(x_i, R\sigma)\right] &= \Pr_{x\sim \hat{q}}\left[x \in \bigcup_{j=1}^{n} B(x_j, R\sigma)\right] \\
&= \frac{1}{n}\sum_{i=1}^{n} \Pr_{x\sim\tau}\left[x \in \left(\bigcup_{j=1}^{n} B(x_j, R\sigma) - x_i\right)\right] \\
&\geq \frac{1}{n}\sum_{i=1}^{n} \Pr_{x\sim\tau}\left[x \in (B(x_i, R\sigma) - x_i)\right] \\
&= \int_{B(0,R\sigma)} \tau(x)dx \\
&= \int_{B(0,R\sigma)} \frac{1}{\sigma}K\left(\frac{x}{\sigma}\right)dx \\
&= \int_{B(0,R)} K(u)du \\
&\geq \frac{1}{2},
\end{aligned}$$

completing the proof.

□

C.2. Proof of Proposition 4.2

Proposition C.3 (Proposition 4.2)**.** *Let p be a probability distribution with support precisely equal to a smooth, compact, k-dimensional sub-manifold of $\mathbb{R}^d$, M. Additionally, suppose that p has a continuous density function over M. Then it follows that p is k-regular.*

To prove this, we begin with the following lemma.

Lemma C.4. *Let $k > 0$ be a constant. Let p be a probability distribution for which the following properties hold:*

1. The map $supp(p) \times \mathbb{R}^+ \to R^+$ defined by $(x, r) \mapsto p(B(x, r))$ is continuous.

2. The map $supp(p) \to \mathbb{R}^+$ defined by $x \mapsto \lim_{r\to 0} \frac{p(B(x,r)}{r^k}$ is well defined, continuous, and strictly positive over its domain.

3. p has compact support.

Then p is k-regular.

Proof. The map $r \to r^k$ is clearly continuous. It follows by properties (1.) and (2.), the following is a continuous map: $F : supp(p) \times \mathbb{R}^{\geq 0} \to \mathbb{R}^+$ where

$$F(x, r) = \begin{cases} \frac{p(B(x,r))}{r^k} & r > 0 \\ \lim_{s\to 0} \frac{p(B(x,s))}{s^k} & r = 0, \end{cases}.$$

Next, fix $\epsilon > 0$, as arbitrary. We desire to show that p_ϵ exists for which the conditions of Definition 4.1 hold. Without loss of generality, we can assume $\epsilon < 1$, as the case $\epsilon \geq 1$ can easily be handled by just using p_ϵ for a smaller value of ϵ.

For any $x > 0$, since F is continuous, there exists $\rho_x > 0$ such that for any $x', \in B(x, \rho_x)$ and $r \leq \rho_x$,

$$|F(x', r) - F(x, 0)| < F(x, 0)\frac{\epsilon}{9}.$$

It follows for any such x' that

$$p(B(x', \rho_x)) = F(x, \rho_x)\rho_x^k \geq (F(x, 0))(1 - \frac{\epsilon}{9}), \tag{10}$$

and for any $0 < s < r < \rho_x$, we have

$$\begin{aligned}\frac{p(B(x', r))}{r^k} &= F(x', r) \\ &\leq F(x, 0)(1 + \frac{\epsilon}{9}) \\ &\leq F(x', s)\frac{1 + \frac{\epsilon}{9}}{1 - \frac{\epsilon}{9}} \\ &\leq F(x', s)\left(1 + \frac{\epsilon}{3}\right),\end{aligned}$$

and

$$\begin{aligned}\frac{p(B(x', r))}{r^k} &= F(x', r) \\ &\geq F(x, 0)(1 - \frac{\epsilon}{9}) \\ &\geq F(x', s)\frac{1 - \frac{\epsilon}{9}}{1 + \frac{\epsilon}{9}} \\ &\geq F(x', s)\left(1 + \frac{\epsilon}{3}\right)^{-1},\end{aligned}$$

which together imply that

$$\left(1 + \frac{\epsilon}{3}\right)^{-1}\frac{p(B(x, s))}{s^k} \leq \frac{p(B(x, r))}{r^k} \leq \left(1 + \frac{\epsilon}{3}\right)\frac{p(B(x, s))}{s^k}. \tag{11}$$

Finally, observe that the balls $B(x, r_x)$ cover the support of p. Since $supp(p)$ is compact, it follows that there exists a finite sub-cover of such balls, C. We finally let $p_\epsilon = \min_{B(x,r_x)\in C} F(x, 0)(1 - \frac{\epsilon}{9})$. It then follows by Equations 10 and 11, that p has met the criteria necessary for p to be k-regular, as desired. □

We are now prepared to prove Proposition 4.2.

Proof. It suffices to show that the conditions of Lemma C.4 hold. Conditions 1. and 3. immediately hold since the probability mass of the surface (i.e. points on the boundary) of a ball $B(x, r)$ will be 0 as its intersection with M would be a $(k-1)$-dimensional manifold.

Thus, it remains to verify condition 2. For any $x, y \in M$, let $d_M(x, y)$ denote the geodesic distance between x and y (with $||x - y||$ still denoting their euclidean distance in $\mathbb{R}^d$ as M is embedded in $\mathbb{R}^d$). Since M is a smooth, compact manifold, it follows that for any $x \in M$,

$$\lim_{r\to 0}\sup_{||x-y||=r}\frac{||x - y||}{d_M(x, y)} = 1.$$

In other words, at a small scale, the geodesic distance and the Euclidean distance converge. It follows that

$$\lim_{r\to 0}\frac{p(B(x, r))}{r^k} = \lim_{s\to 0}\frac{p(B_M(x, s))}{s^k},$$

where $B_M(x,s)$ denotes the geodesic ball of radius s centered at x on M. However, the latter quantity is precisely equal to the density function over M (up to a constant factor, since $\lim_{s\to 0}\frac{vol_M(B_M(x,s))}{s^k}=\omega_k$, where ω_k is the volume of the k-sphere). Since by assumption our density function is continuous and non-zero everywhere on the manifold, it follows that the map above must be well defined and continuous giving us condition 2. of Lemma C.4, as desired. □

C.3. Proof of Proposition 4.3

Proposition C.5 (Proposition 4.3)**.** *Let p be an k-regular distribution, and let $\epsilon>0$ be arbitrary. Then if $n=O\left(\frac{d\ln\frac{1}{\delta\epsilon p_\epsilon}}{\epsilon^2 p_\epsilon}\right)$ with probability at least $1-\delta$ over $S\sim p^n$, for all $x\in\mathbb{R}^d$ and $r>0$,*

$$\left(1+\frac{\epsilon}{2}\right)^{-1}p(B(x,r))\le Est(x,r,S)\le\left(1+\frac{\epsilon}{2}\right)p(B(x,r)). \tag{12}$$

Proof. We begin by first applying standard uniform convergence over ℓ_2 balls in $\mathbb{R}^d$ (which have a VC dimension of at most $d+2$). To this end, let

$$\beta_n=\sqrt{\frac{4(d+2)\ln\frac{16n}{\delta}}{n}}.$$

Then by the standard result of Vapnik and Chervonenkis, with probability $1-\delta$ over $S\sim p^n$, for all $x\in\mathbb{R}^d$ and all $r>0$,

$$\frac{|S\cap B(x,r)|}{n}-\beta_n\sqrt{\frac{|S\cap B(x,r)|}{n}}\le p(B(x,r))\le\frac{|S\cap B(x,r)|}{n}+\beta_n^2+\beta_n\sqrt{\frac{|S\cap B(x,r)|}{n}}. \tag{13}$$

Next, assume that

$$n\ge\frac{888(d+2)\ln\left(\frac{14208(d+2)}{\delta\min(\epsilon,1)^2p_\epsilon}\right)}{\min(\epsilon,1)^2p_\epsilon}. \tag{14}$$

It is clear that for an appropriate constant, we have $n=O\left(\frac{d\ln\frac{d}{\delta\epsilon p_\epsilon}}{\epsilon^2p_\epsilon}\right)$. Thus, it suffices to show that if Equation 13 holds for all x,r, then the desired bound, Equation 12, does as well.

To this end, let x,r be arbitrary, and let b be as defined in Algorithm 1. Let $b'=|S\cap B(x,r)|$ be the number of elements from S in $B(x,r)$. Then we have two cases.

Case 1: $b'\ge b$

It follows from Algorithm 1 that $Est(x,r,S)=\frac{b'}{n}$. We now set b as

$$\frac{b}{n}=\frac{400(d+2)\ln\frac{16n}{\delta}}{n\min(\epsilon,1)^2}=\frac{100\beta_n^2}{\epsilon^2}, \tag{15}$$

which clearly obeys the desired asymptotic bound given in Algorithm 1. Let $c=\sqrt{\frac{b'}{n\beta_n^2}}$. Then $b'\ge b$ implies that $c\ge\frac{10}{\min(\epsilon,1)}$. It follows that

$$\frac{c+1}{c^2}\le\frac{1}{c-1}\le\frac{\min(\epsilon,1)}{9}. \tag{16}$$

Substituting Equations 15 and 16 into Equation 13, we have

$$\begin{aligned}\frac{Est(x,r,S)}{p(B(x,r))}&\ge\frac{\frac{b'}{n}}{\frac{b'}{n}+\beta_n^2+\beta_n\sqrt{\frac{b'}{n}}}\\&=\frac{c^2}{c^2+1+c}\\&=\left(1+\frac{c+1}{c^2}\right)^{-1}\\&\ge\left(1+\frac{\min(\epsilon,1)}{9}\right)^{-1}\end{aligned} \tag{17}$$

and

$$\begin{aligned}\frac{Est(x,r,S)}{p(B(x,r))} &\leq \frac{\frac{b'}{n}}{\frac{b'}{n} - \beta_n\sqrt{\frac{b'}{n}}} \\ &= \frac{c^2}{c^2 - c} \\ &= 1 + \frac{1}{c-1} \\ &\leq 1 + \frac{\min(\epsilon, 1)}{9}.\end{aligned} \tag{18}$$

Together, Equations 17 and 18 imply that $Est(x, r, S)$ is sufficiently accurate.

Case 2: $b' < b$

We begin by bounding β_n^2 in terms of p_ϵ. We have,

$$\begin{aligned}\beta_n^2 &= \frac{4(d+2)\ln(16n/\delta)}{n} \\ &= 4(d+2)\ln\left(\frac{14208(d+2)}{\delta\min(\epsilon,1)^2 p_\epsilon}\ln\left(\frac{14208(d+2)}{\delta\min(\epsilon,1)^2 p_\epsilon}\right)\right)\frac{\min(\epsilon,1)^2 p_\epsilon}{888(d+2)\ln\left(\frac{14208(d+2)}{\delta\min(\epsilon,1)^2 p_\epsilon}\right)} \\ &\leq 8(d+2)\ln\left(\frac{14208(d+2)}{\delta\min(\epsilon,1)^2 p_\epsilon}\right)\frac{\min(\epsilon,1)^2 p_\epsilon}{888(d+2)\ln\left(\frac{14208(d+2)}{\delta\min(\epsilon,1)^2 p_\epsilon}\right)} \\ &= \frac{p_\epsilon\min(\epsilon,1)^2}{111}.\end{aligned} \tag{19}$$

Now, let r_* be as defined in Algorithm 1. Then $|S \cap B(x, r_*)| = b$. Our main idea will be to show that $p(B(x, r_*) \leq p_\epsilon$, and then use Equations 17 and 18 for r_* (which is possible since $|S \cap B(x, r_*)| = b$) along with the definition of p_ϵ (Definition 4.1) to bound $Est(x, r, S)$ in terms of $p(B(x, r))$. To this end, we have by Equations 13 and 19 along with the fact that $b = \frac{100\beta_n^2}{\min(\epsilon,1)^2}$ (Equation 15) that

$$\begin{aligned}p(B(x,r_*)) &\leq \frac{|S\cap B(x,r_*)|}{n} + \beta_n^2 + \beta_n\sqrt{\frac{|S\cap B(x,r_*)|}{n}} \\ &= \frac{b}{n} + \beta_n^2 + \beta_n\sqrt{\frac{b}{n}} \\ &= \beta_n^2\left(\frac{100}{\min(\epsilon,1)^2} + 1 + \frac{10}{\min(\epsilon,1)}\right) \\ &\leq \frac{p_\epsilon^2\min(\epsilon,1)^2}{111}\frac{111}{\min(\epsilon,1)^2} = p_\epsilon.\end{aligned}$$

It follows from Definition 4.1 that

$$\left(1+\frac{\epsilon}{3}\right)^{-1}\frac{p(B(x,r_*))r^k}{r_*^k} \leq p(B(x,r)) \leq \left(1+\frac{\epsilon}{3}\right)\frac{p(B(x,r_*))r^k}{r_*^k}. \tag{20}$$

Finally, by the definition of $Est(x, r, S))$ (Algorithm 1), we have that $Est(x, r, S) = \frac{Est(x,r_*,S)r^k}{r_*^k}$. Combining this with Equation 20 the definition of $Est(x, r, S)$ (Algorithm 1) along with Equations 17 and 18 (which can be safely applied to r_*

by reverting to Case 1), we have

$$\begin{aligned}\frac{Est(x,r,S)}{p(B(x,r))} &= \frac{\frac{Est(x,r_*,S)r^k}{r_*^k}}{p(B(x,r))} \leq \frac{\frac{Est(x,r_*,S)r^k}{r_*^k}\left(1+\frac{\epsilon}{3}\right)}{\frac{p(B(x,r_*))r^k}{r_*^k}} \\ &= \frac{Est(x,r_*,S)\left(1+\frac{\epsilon}{3}\right)}{p(B(x,r_*))} \leq \left(1+\frac{\epsilon}{3}\right)\left(1+\frac{\min(\epsilon,1)}{9}\right) \\ &\leq 1+\frac{\epsilon}{2},\end{aligned}$$

and

$$\begin{aligned}\frac{Est(x,r,S)}{p(B(x,r))} &= \frac{\frac{Est(x,r_*,S)r^k}{r_*^k}}{p(B(x,r))} \geq \frac{\frac{Est(x,r_*,S)r^k}{r_*^k}}{\frac{p(B(x,r_*))r^k}{r_*^k}\left(1+\frac{\epsilon}{3}\right)} \\ &= \frac{Est(x,r_*,S)}{p(B(x,r_*))\left(1+\frac{\epsilon}{3}\right)} \geq \left(1+\frac{\epsilon}{3}\right)^{-1}\left(1+\frac{\min(\epsilon,1)}{9}\right)^{-1} \\ &\geq \left(1+\frac{\epsilon}{2}\right)^{-1},\end{aligned}$$

which concludes the proof.

□

C.4. Proof of Theorem 5.1

Theorem C.6 (Theorem 5.1)**.** *Data_Copy_Detect is a data-copying detector (Definition 3.2) with sample complexity at most*

$$m_p(\epsilon,\delta) = O\left(\frac{d\ln\frac{d}{\delta\epsilon p_\epsilon}}{\epsilon^2 p_\epsilon}\right),$$

for all regular distributions, p.

Proof. Let C be the constant defined in Proposition 4.3, and let $n \geq C\frac{d\ln\frac{d}{\delta\epsilon p_\epsilon}}{\epsilon^2 p_\epsilon}$. Let $S \sim p^n$ be a set of n i.i.d training points, $\{x_1, x_2, \ldots, x_n\}$, and let $q \sim A(S)$ be an arbitrary generated distribution.

By Proposition 4.3, the subroutine $Est(x,r,S)$ is accurate over any x and r up to a factor of $(1+\epsilon)$ with probability at least $1-\frac{\delta}{3}$ (we can achieve this by simply making n a bit larger and substituting $\frac{\delta}{3}$ into Proposition 4.3). Suppose this holds, meaning that that for all $x \in \mathbb{R}^d$ and all $r > 0$, the condition of Proposition 4.3 holds, and

$$(1+\epsilon)^{-1}p(B(x,r)) \leq Est(x,r,S) \leq (1+\epsilon)p(B(x,r)). \tag{21}$$

We desire to show that

$$cr_q^{-\epsilon} - \epsilon \leq DataCopyDetect(q,S) \leq cr_q^{\epsilon} + \epsilon.$$

To do so, we begin applying uniform convergence over $T \sim q^m$. To this end, let

$$\beta_m = \sqrt{\frac{4(d+2)\ln\frac{48m}{\delta}}{m}}.$$

Then by the standard result of Vapnik and Chervonenkis, with probability $1-\frac{\delta}{3}$ over $T \sim q^m$, for all $x \in \mathbb{R}^d$ and all $r > 0$,

$$\frac{|T\cap B(x,r)|}{m} - \beta_m\sqrt{\frac{|T\cap B(x,r)|}{m}} \leq q(B(x,r)) \leq \frac{|T\cap B(x,r)|}{m} + \beta_m^2 + \beta_n\sqrt{\frac{|T\cap B(x,r)|}{m}}. \tag{22}$$

Observe that by the definition of m, we have

$$\begin{aligned}\beta_m^2 &= \frac{4(d+2)\ln(48m/\delta)}{m} \\ &= 4(d+2)\ln\left(\frac{98304n^2(d+2)}{\delta\epsilon^2\min(\epsilon,1)^2}\ln\left(\frac{98304n^2(d+2)}{\delta\epsilon^2\min(\epsilon,1)^2}\right)\right)\frac{\epsilon^2\min(\epsilon,1)^2}{2048n^2(d+2)\ln\left(\frac{98304n^2(d+2)}{\delta\epsilon^2\min(\epsilon,1)^2}\right)} \\ &\leq 8(d+2)\ln\left(\frac{98304n^2(d+2)}{\delta\epsilon^2\min(\epsilon,1)^2}\right)\frac{\epsilon^2\min(\epsilon,1)^2}{2048n^2(d+2)\ln\left(\frac{98304n^2(d+2)}{\delta\epsilon^2\min(\epsilon,1)^2}\right)} \\ &= \frac{\epsilon^2\min(\epsilon,1)^2}{256n^2}.\end{aligned} \tag{23}$$

Next, suppose x, r satisfy that $q(B(x,r)) \geq \frac{\epsilon}{2n}$. For convenience, let $q(\widehat{B(x,r)})$ denote $\frac{|T\cap B(x,r)|}{m}$. By applying Equations 22 and 23, it follows that

$$\begin{aligned}\frac{q(\widehat{B(x,r)})}{q(B(x,r))} &\leq \frac{q(B(x,r))+\beta_m}{q(B(x,r)} \\ &\leq 1+\frac{\beta_m}{q(B(x,r)} \\ &\leq 1+\frac{\min(\epsilon,1)}{8},\end{aligned}$$

and

$$\begin{aligned}\frac{q(B(x,r))}{q(\widehat{B(x,r)})} &\leq \frac{q(B(x,r))}{q(B(x,r))-\beta_m^2-\beta_m\sqrt{q(\widehat{B(x,r)})}} \\ &\leq \frac{q(B(x,r))}{q(B(x,r)-2\beta_m} \\ &= \frac{1}{1-\frac{2\beta_m}{q(B(x,r))}} \\ &\leq \frac{1}{1-\frac{\min(\epsilon,1)}{4}} \\ &\leq 1+\frac{\min(\epsilon,1)}{3}.\end{aligned}$$

Combining these, we have

$$\left(1+\frac{\min(\epsilon,1)}{3}\right)^{-1} \leq \frac{q(B(x,r)}{q(\widehat{B(x,r)})} \leq \left(1+\frac{\min(\epsilon,1)}{3}\right) \tag{24}$$

Next, for $1 \leq i \leq n$, let r_i^* be the radii defined in Algorithm 2. Define $r_i^{-\epsilon}$ and r_i^{ϵ} to be the maximal radii r for which q respectively $(\lambda(1+\epsilon), \gamma(1+\epsilon)^{-1})$-copies, and $(\lambda(1+\epsilon)^{-1}, \gamma(1+\epsilon))$-copies p about x_i. Then we have the following claims.

Claim 1: For $1 \leq i \leq n$, if $q(B(x, r_i^*)) \geq \frac{\epsilon}{2n}$, $r_i^* \leq r_i^{\epsilon}$.

Proof. Because $Est(x_i, r_i^*, S) \leq \gamma$, it follows by Equation 21 that $p(B(x_i, r_i^*)) \leq \left(1+\frac{\epsilon}{2}\right)\gamma$. Furthermore, by also applying Equation 24 we have that

$$\frac{q(x_i, r_i^*)}{p(x_i, r_i^*)} \geq \frac{\frac{|B(x_i,r_i^*)\cap T|}{m}}{Est(x_i, r_i^*, S)\left(1+\frac{\min(\epsilon,1)}{3}\right)\left(1+\frac{\epsilon}{2}\right)} \geq \lambda(1+\epsilon)^{-1}.$$

Thus q $(\lambda(1+\epsilon)^{-1}, \gamma(1+\epsilon))$-copies all points in $B(x_i, r_i^*)$ implying $r_i^* \leq r_i^{\epsilon}$. □

Claim 2: For $1 \leq i \leq n$, if $q(B(x, r_i^{-\epsilon})) \geq \frac{\epsilon}{2n}$, then $r_i^{-\epsilon} \leq r_i^*$.

Proof. For the left hand side, we use a similar argument. By Equation 21 along with the definition of r_i^ϵ, we have $Est(x_i, r_i^{-\epsilon}, S) \leq \gamma(1+\epsilon)^{-1}\left(1+\frac{\epsilon}{2}\right) \leq \gamma$. By Equations 21 and 24, we have

$$\frac{\frac{|B(x_i, r_i^{-\epsilon}) \cap T|}{m}}{Est(x_i, r_i^*, S)} \geq \frac{q(B(x_i, r_i^{-\epsilon}))}{p(B(x_i, r_i^{-\epsilon}))\left(1+\frac{\min(\epsilon,1)}{3}\right)\left(1+\frac{\epsilon}{2}\right)} \geq \lambda,$$

with the last inequality coming again from the definition of $r_i^{-\epsilon}$. Thus, $r_i^{-\epsilon}$ meets the criteria from Algorithm 2 required to be selected as r_i^*. As a technical note, because Algorithm 2 only considers finitely many radii, it may not consider precisely $r_i^{-\epsilon}$. However, this is not a problem, as the nearest considered radii to this point have nearly unchanged values of $Est(x, r, S)$ and $\frac{|B(x,r) \cap T|}{m}$, meaning that some similar radius will be considered. □

Finally, armed with our claims, we now consider the total region of points in which Algorithm 2 claimed data-copying occurs. Let S^1 and S^2 be the sets of indices for which the conditions are violated for claims 1 and 2 respectively. Then it follows from Claim 1 that

$$\begin{aligned} cr_q^\epsilon - q\left(\cup_{i=1}^n B(x_i, r_i^*)\right) &= q\left(\cup_{i=1}^n B(x_i, r_i^\epsilon)\right) - q\left(\cup_{i=1}^n B(x_i, r_i^*)\right) \\ &\geq q\left(\cup_{i=1}^n B(x_i, r_i^\epsilon)\right) - q\left(\cup_{i \notin S^1} B(x_i, r_i^*)\right) - q\left(\cup_{i \in S^1} B(x_i, r_i^*)\right) \\ &\geq -\frac{\epsilon}{2}. \end{aligned}$$

Here we are using Claim 1 to hand all terms that are not in S^1, and then crudely bounding the remaining terms with $\frac{\epsilon}{2n}$. Similarly, by Claim 2, we have

$$\begin{aligned} q\left(\cup_{i=1}^n B(x_i, r_i^*)\right) - cr_q^{-\epsilon} &= q\left(\cup_{i=1}^n B(x_i, r_i^*)\right) - q\left(\cup_{i=1}^n B(x_i, r_i^{-\epsilon})\right) \\ &\geq q\left(\cup_{i=1}^n B(x_i, r_i^*)\right) - q\left(\cup_{i \notin S^2} B(x_i, r_i^{-\epsilon})\right) - q\left(\cup_{i \in S^2} B(x_i, r_i^{-\epsilon})\right) \\ &\geq -\frac{\epsilon}{2}. \end{aligned}$$

Combining these, we see that

$$cr_q^{-\epsilon} - \frac{\epsilon}{2} \leq q\left(\cup_{i=1}^n B(x_i, r_i^*)\right) \leq cr_q^\epsilon + \frac{\epsilon}{2}.$$

All the remains is to show that our last step of Algorithm 2, in which we estimate this mass, is accurate up to a factor of $\frac{\epsilon}{2}$. However, this immediately follows from the fact that we use $\frac{20 \log \frac{1}{\delta}}{\epsilon^2}$ samples (last line of Algorithm 2). In particular, because this holds with probability $1 - \frac{\delta}{3}$, we can apply a union bound with our other two probabilistic events (Est being sufficiently close, and T yielding uniform convergence) to get a total failure probability of δ, as desired. □

C.5. Proof of Theorem 6.1

Theorem C.7 (Theorem 6.1)**.** *Let B be a data-copying detector. Let $\epsilon = \delta = \frac{1}{3}$. Then there exist 1-regular distributions p for which p_ϵ is arbitrarily small and B has sample complexity*

$$m_p(\epsilon, \delta) \geq \Omega(\frac{1}{p_\epsilon}).$$

More precisely, for all integers $\kappa > 0$, there exists a probability distribution p such that $\frac{1}{9\kappa} \leq p_\epsilon \leq \frac{1}{\kappa}$, and $m_p(\epsilon, \delta) > \Omega(\kappa)$.

Proof Outline: Let κ be a sufficiently large integer. Then we take the following steps.

1. We define the probability distribution p_T, where $T \subset [2\kappa] = \{1, 2, \ldots, 2\kappa\}$ is a subset with $|T| = \kappa$ that parametrizes our distribution. We then show that for all T, p_T is a 1-regular distribution satisfying $\frac{1}{9\kappa} \leq (p_T)_\epsilon \leq \frac{1}{\kappa}$.
2. We define a generative algorithms A_T and A'_T, where as before $T \subset [2\kappa]$ with $|T| = \kappa$. We then show that if $S \sim p_T^{O(\kappa)}$, $A_T(S)$ is likely to have a high data-copy rate with respect to p_T, whereas $A'_T(S)$ has a data-copy rate of 0.

3. We construct families

$$\mathcal{F} = \{(p_T, A_T) : T \subset [2\kappa], |T| = \kappa\} \text{ and } \mathcal{F}' = \{(p_T, A'_T) : T \subset [2\kappa], |T| = \kappa\},$$

and show that $(S, A(S))$ follows very similar distributions when S is drawn from $p^{O(\kappa)}$ and (p, A) is drawn from $\mathcal{F}$ and $\mathcal{F}'$ respectively, meaning that it is difficult to tell which family the pair (p, A) is drawn from.

4. We show that if B has sample complexity at most $O(\kappa)$, then by (2.) it would be able to distinguish $(S, A_T(S))$ from $(S, A'_T(S))$ thus contradicting (3.) We thus conclude B has sample complexity $\Omega(\kappa)$, as desired.

Proof. We follow the outline above proceeding step by step.

Step 1: constructing p_T

First, set $\gamma < 1$ arbitrarily, and let $\lambda = 13$. Note that these constants are chosen out of convenience, and for different values of ϵ, δ, different ones can be chosen.

Let $\kappa > 0$ be any integer, and let $[2\kappa] = \{1, 2, 3, \ldots, 2\kappa\}$. Let $C_1, C_2, \ldots, C_{2\kappa}$ be 2κ disjoint unit circles in $\mathbb{R}^d$ with distance at least 3 between any two circles. All data distributions, p_T, that we construct will have support over $\cup_{i=1}^{2\kappa} C_i$, and will further obey the constraint that their marginal distribution over any C_i is precisely the uniform distribution. Thus, a distribution p_T is uniquely specified by the probability mass it assigns to each circle. To this end, we define p_T as follows.

Definition C.8. Let $T \subset [2\kappa]$ be a subset of indices with $|T| = \kappa$. Then p_T is the unique probability distribution satisfying the criteria above such that

$$p_T(C_i) = \begin{cases} \frac{1}{3\kappa} & i \in T \\ \frac{2}{3\kappa} & i \notin T \end{cases}$$

Lemma C.9. *p_T is 1-regular, and satisfies $\frac{1}{9\kappa} \leq (p_T)_\epsilon \leq \frac{2}{3\kappa}$ when $\epsilon = \frac{1}{3}$.*

Proof. First, we observe that by Proposition 4.2, we immediately have that p_T is 1-regular as a union of disjoint circles is a 1 dimensional closed manifold, and the density function of p_T with respect to each circle is uniform and therefore continuous. For convenience, we let p denote p_T, as by symmetry, $(p_T)_\epsilon$ is equal for all values of T.

Next, for $r \leq 2$ and $x \sim p$, we compute $\frac{p(B(x,r)}{r}$. Suppose $x \in C_i$. The key observation is that the density of p over C_i is uniform, and thus since $r \leq 2$, the mass of $B(x, r)$ can be found by simply computing the arc length. It follows that

$$\frac{p(B(x,r))}{r} = p(C_i))\frac{4\arcsin(\frac{r}{2})}{2\pi r}. \tag{25}$$

By some basic properties about arcsin, it follows that $\frac{p(B(x,r)}{r}$ is monotonically increasing with $0 < r \leq 2$ and satisfies $\lim_{r\to 0^+} \frac{p(B(x,r)}{r} = \frac{p(C_i)}{\pi}$ and $\frac{p(B(x,2))}{2} = \frac{p(C_i)}{2}$. Using this, we now prove the upper and lower bounds for p_ϵ beginning with the upper bound.

Assume towards a contradiction that $p_\epsilon > \frac{2}{3\kappa}$. By Definition 4.1, this implies that for any sufficiently small $r > 0$, we have

$$\left(1 + \frac{\epsilon}{3}\right)^{-1} \frac{p(B(x,r))}{r} \leq \frac{p(B(x,2))}{2} \leq \left(1 + \frac{\epsilon}{3}\right) \frac{p(B(x,r))}{r},$$

as for any $x \sim p$, $p(B(x, 2)$ is at most $\frac{2}{3\kappa}$. Substituting equation 25 and taking the limit as $r \to 0^+$, it follows that $\frac{p(C_i)}{2} \leq \frac{7}{6}\frac{p(C_i)}{\pi}$, which is a contradiction giving us that $p_\epsilon \leq \frac{2}{3\kappa}$.

Next, for the lower bound, it suffices to show that for any x and any $0 < s \leq r$ with $p(B(x, r)) \leq \frac{1}{9\kappa}$ that

$$\left(1 + \frac{\epsilon}{3}\right)^{-1} \frac{p(B(x,s))}{s} \leq \frac{p(B(x,r))}{r} \leq \left(1 + \frac{\epsilon}{3}\right) \frac{p(B(x,s))}{s}. \tag{26}$$

Applying Equation 25 with $r = 1$, we have for any $x \sim p$,

$$\frac{p(B(x,1))}{1} = p(C_i)\frac{4\arcsin(\frac{1}{2})}{2\pi} = p(C_i)\frac{1}{3} \geq \frac{1}{3\kappa}\frac{1}{3} = \frac{1}{9\kappa}.$$

Since $\frac{p(B(x,r)}{r}$ is monotonic in r, it follows that $p(B(x,r)) \leq \frac{1}{9\kappa}$ only if $r \leq 1$. We are now prepared to prove Equation 26.

The left inequality immediately holds since $\frac{p(B(x,r)}{r}$ is monotonic in r. For the right inequality, we have that if r satisfies $p(B(x,r) \leq \frac{1}{9\kappa}$, then $r \leq 1$ implying for $x \in C_i$,

$$\begin{aligned}
\frac{p(B(x,r))}{r} &\leq \frac{p(B(x,1)}{1} \\
&= p(C_i)\frac{1}{3} \\
&\leq (1+\frac{1}{9})\frac{p(C_i)}{\pi} \\
&= \left(1+\frac{\epsilon}{3}\right)\lim_{t\to 0}\frac{p(B(x,t))}{t} \\
&\leq \left(1+\frac{\epsilon}{3}\right)\frac{p(B(x,s))}{s},
\end{aligned}$$

as desired. □

Step 2: defining A_T and A'_T

Having defined our probability distributions, p_T, we now define our generative algorithms A_T and A'_T. Recall that a generative algorithm, A, is any process that takes as input a set of points $S \in \mathbb{R}^d$ and then returns a probability distribution, $A(S)$ over $\mathbb{R}^d$. The algorithm is allowed to have randomization.

A_T and A'_T will always be constrained to output distributions that are similar to p_T in the sense that they have support over a disjoint union of circles, and their marginal distribution over any circle (within the support) is the uniform distribution. The only change is that we add one extra circle, C_0, that satisfies

$$||C_0 - C_i|| \geq 2 + \max_{i,j} ||C_i - C_j||,$$

meaning that it is very far from all C_i. Thus, any outputted distribution by A_T or A'_T can be specified by specifying the probability mass it assigns to each circle in $\{C_0, C_1, \ldots, C_{2\kappa}\}$.

Both A_T and A'_T will operate under the assumption that the training sample of points S is relatively well behaved. In the event that this does not hold, A_T and A'_T will output the uniform distribution over C_0 as a default. We now formally define this criteria upon S.

Definition C.10. Let S be a finite set of points and $T \subset [2\kappa]$ be a set of indices with $|T| = \kappa$. We say that S covers T the sets $L = \{i : i \in T, |C_i \cap S| = 1\}$ and $L' = \{i : i \notin T, |C_i \cap S| = 1\}$ both satisfy $|L|, |L'| \geq \frac{\kappa}{8}$.

Observe that this definition if symmetric with respect to complements meaning that S covers T if and only if S covers $[2\kappa] \setminus T$. We now use this to define A_T and A'_T beginning with A_T.

Definition C.11. Let $T \subset [2\kappa]$ be a subset of indices with $|T| = \kappa$, and let S be any set of points in $\mathbb{R}^d$. Then A_T consists of the following steps. We let q denote its output, and $A_T(S)$ denote the full distribution of potential generated distributions q.

1. If S does *not* cover T, then output the uniform distribution over C_0 as q.
2. Otherwise, let $L = \{i : i \in T, |C_i \cap S| = 1\}$ be as defined in Definition C.10.
3. Randomly select $L_* \subset L$ with $|L_*| = \frac{\kappa}{8}$ at uniform.
4. We then let q be the unique probability distribution satisfying the criteria above with

$$q(C_i) = \begin{cases} \frac{\lambda(1+\epsilon)}{3\kappa} & i \in L_* \\ 0 & i \in [2\kappa] \setminus L_* \\ 1 - \frac{\lambda(1+\epsilon)}{24} & i = 0 \end{cases}$$

Having defined A_T, we define A'_T by having $A'_T = A_{[2\kappa]\setminus T}$. That is,

Definition C.12. Let $T \subset [2\kappa]$ be a subset of indices with $|T| = \kappa$, and let S be any set of points in $\mathbb{R}^d$. Then $A'_T(S)$ is precisely $A_{[2\kappa]\setminus T}(S)$ where $[2\kappa] \setminus T$ is the complement of T.

Observe that if S covers T, then by Definitions C.11 and C.12, $A_T(S)$ and $A'_T(S)$ will both have supports non-trivially intersecting the set of circles over which p_T is based, $\cup_{i=1}^{2\kappa} C_i$. We now show that this condition is sufficient for our desired behavior with respect to data-copying.

Lemma C.13. *Let κ satisfy $\frac{1}{3\kappa} \leq \gamma$. For any $T \subset [2\kappa]$, let S be any set of points in the support of p_T that covers T. Then with probability 1 over the randomness of A_T and A'_T, $q_T \sim A_T(S)$ and $q'_T \sim A'_T(S)$ have respective data-copy rates $cr^{-\epsilon}_{q_T}$ and $cr^{\epsilon}_{q'_T}$ satisfying*

$$cr^{-\epsilon}_{q_T} \geq \frac{\lambda(1+\epsilon)}{24},$$
$$cr^{\epsilon}_{q'_T} = 0.$$

Proof. Let L and L' be as in Definition C.10. We begin with $cr^{-\epsilon}_{q_T}$, which was the data-copy rate of q_T with parameters $(\lambda(1+\epsilon), \gamma(1-\epsilon))$ (Definition 3.3).

Since $|L| \geq \frac{\kappa}{8}$, there exists $L_* \subset L$ with $|L_*| = \frac{\kappa}{8}$ such that q_T has support over $C_0 \cup \{C_i\}_{i \in L_*}$. For any $i \in L_*$, let x_i denote the unique point in the intersection of C_i and S. Observe that by the definition of L, $p_T(B(x_i, 2)) = \frac{1}{3\kappa}$. On the other hand, we have $q_T(B(x_i, 2)) = q_T(C_i) = \frac{\lambda(1+\epsilon)}{3\kappa}$, with the first equality holding since C_i is the only circle that intersects $B(x_i, 2)$. It follows by Definition 2.1 that q_T $(\lambda(1+\epsilon), \gamma(1+\epsilon)^{-1})$-copies all $x \in C_i$. Taking the total measure (under q_T), we have

$$cr^{-\epsilon}_{q_T} \geq q_T(\cup_{i \in L_*} C_i) = \frac{\kappa}{8} \frac{\lambda(1+\epsilon)}{3\kappa} = \frac{\lambda(1+\epsilon)}{24\kappa}$$

as desired.

Next, we show $cr^{\epsilon}_{q'_T} = 0$. To do so, it suffices to show that for all $x \in S$ and $r > 0$,

$$q'_T(B(x,r)) < \lambda(1+\epsilon)^{-1} p_T(B(x,r)),$$

as this would imply that no points are $(\lambda(1+\epsilon)^{-1}, \gamma(1+\epsilon))$-copied.

Observe that $M = \cup_{1 \leq i \leq 2\kappa} C_i$ is a 1-dimensional manifold containing the entire support of p_T, and that furthermore the marginal distribution of $q'_T(S)$ over M has a well defined probability density with respect to M. Since $x \in S$ and $S \subset M$ (as $S \subset supp(p_T)$), we can consider two cases: if $B(x,r)$ intersects C_0 (the only region in the support of $A'_T(S)$ outside M), and if $B(x,r)$ does not intersect C_0.

Case 1: $B(x,r)$ intersects C_0 Observe that by the definition of C_0, $C_i \subset B(x,r)$ for all $1 \leq i \leq 2\kappa$. This is because C_0 is very far from all the other circles. However, this implies $M \subset B(x,r)$ meaning that $p_T(B(x,r)) \geq p_T(M) = 1$. However, $q'_T(B(x,r))$ is clearly at most 1, making the desired inequality trivially hold as $\lambda(1+\epsilon)^{-1} > 1$.

Case 2: $B(x,r)$ does not intersect C_0 Observe that this implies $supp(p_T) \cap B(x,r) = supp(q'_T \cap B(x,r) \subseteq M$, as both of these distributions only have support on M when outside of C_0. Since p_T and q'_T both have well defined probability densities over M, their masses over $B(x,r)$ can be found by integrating their densities over this region.

However, by the definition of A'_T, for any $y \in supp(q'_T)$, we have that $y \in C_i$ where $i \in [2\kappa] \setminus T$. By letting p_T and q'_T denote their respective density functions, it follows that

$$p_T(y) = \frac{2}{3\kappa(2\pi)}, \text{ and } q'_T(y) = \frac{\lambda(1+\epsilon)}{3\kappa(2\pi)}.$$

It follows that $\frac{q'_T(y)}{p_T(x)} = \frac{\lambda(1+\epsilon)}{2} < \lambda(1+\epsilon)^{-1}$. Thus, it follows from integrating as y goes over $B(x,r)$ that $q'_T(B(x,r)) < \lambda(1+\epsilon)^{-1} p_T(B(x,r))$ as desired.

As a slight technical detail, while this inequality will no longer be strict if $p_T(B(x,r)) = 0$, we know that this is never the case since $p_T(B(x,r))$ is strictly positive for all $x \in M$.

□

Next, we bound the probability that set of κ points drawn i.i.d. from p_T, $S \sim p_T^\kappa$, will cover T. To do so, we begin with a combinatorial lemma.

Lemma C.14. *Let m, n be an integers with $\frac{n}{4} \leq m \leq \frac{3n}{4}$. Suppose m numbers are chosen uniformly at random from $\{1, 2, \ldots, n\}$. Then with probability at least $1 - 2\exp\left(\frac{-n}{2048}\right)$, at least $\frac{n}{8}$ numbers in $\{1, 2, \ldots, n\}$ are selected exactly once.*

Proof. Let $b_1, b_2, \ldots, b_m$ denote our m numbers chosen from $\{1, 2, \ldots, n\}$. For $1 \leq i \leq m$, let X_i be an indicator variable for b_i being distinct from x_j for all $1 \leq j < i$, and let $Y_i = 1 - X_i$ be an indicator variable for the opposite. By convention we take $X_1 = 1$ and $Y_1 = 0$. Let $X = \sum_{i=1}^m X_i$ and $Y = \sum_{i=1}^m Y_i$. The key observation is that if Z denotes the number of elements in $\{1, \ldots, n\}$ that are selected exactly once, then $Z \geq X - Y$.

To see this, observe that if we maintain Z as a set while observing $b_1, b_2, \ldots, b_m$, then it follows that whenever $X_i = 1$, we append an element to Z (as its corresponding number b_i will have occurred for the first time and thus be chosen exactly once), and we remove an element from Z only when $Y_i = 1$, as a repeat of a number necessarily implies $Y_i = 1$. It follows that to bound Z, it suffices to bound $X - Y$.

To this end, observe that for any $1 \leq i \leq m$, *regardless* of the outcomes of $X_1, X_2, \ldots, X_{i-1}$, $\mathbb{E}[X_i] \geq \frac{n-i+1}{n}$, as there are at least $n - i + 1$ numbers in $\{1, \ldots, n\}$ that have not been chosen yet. It follows that if $X_i^* = \sum_{j=1}^i X_i - \frac{n-i+1}{n}$ for $1 \leq i \leq m$, then X_i^* is a sub-martingale (as each term in the sum has expected value at least 0) satisfying $|X_i^* - X_{i-1}^*| \leq 1$. Applying Azuma's inequality, we see that

$$\Pr[X_m^* \geq -\frac{n}{32}] \geq 1 - \exp\left(\frac{-n^2}{2048m}\right) \geq 1 - \exp\left(\frac{-n}{2048}\right).$$

We now apply a similar trick for $Y_1, \ldots, Y_m$. In this case, observe that for $1 \leq i \leq m$, *regardless* of the outcomes of $Y_1, \ldots, Y_{i-1}$, $\mathbb{E}[Y_i] \leq \frac{i-1}{m}$, as there can be at most $i - 1$ numbers that have already been chosen and $Y_i = 1$ if and only if the corresponding b_i is equal to one of those $i - 1$ numbers. It follows that $Y_i^* = \sum_{j=1}^i Y_i - \frac{i-1}{m}$ is a super-martingale (as each term has expected value at most 0) with $|Y_i^* - Y_{i-1}^*| \leq 1$. Applying Azuma's inequality, we see that

$$\Pr[Y_m^* \leq \frac{n}{32}] \geq 1 - \exp\left(\frac{-n^2}{2048m}\right) \geq 1 - \exp\left(\frac{-n}{2048}\right).$$

Applying a union bound, we see that with probability at least $1 - 2\exp\left(\frac{-n}{2048}\right)$, $X_m^* \geq \frac{-n}{32}$ and $Y_m^* \leq \frac{-n}{32}$. By substituting

these inequalities in, it follows that with probability $1 - 2\exp\left(\frac{-n}{2048}\right)$, Z satisfies

$$\begin{aligned}
Z &\geq X - Y \\
&= \sum_{i=1}^{m} X_i - \sum_{j=1}^{m} Y_i \\
&= \sum_{i=1}^{m}\left(X_i - \frac{n-i+1}{n}\right) + \sum_{i=1}^{m}\left(\frac{n-i+1}{n}\right) - \sum_{i=1}^{m}\left(Y_i - \frac{i-1}{n}\right) - \sum_{i=1}^{m}\left(\frac{i-1}{n}\right) \\
&= X_m^* - Y_m^* + \sum_{i=1}^{m}\left(\frac{n-i+1}{n}\right) - \sum_{i=1}^{m}\left(\frac{i-1}{n}\right) \\
&\geq -\frac{n}{32} - \frac{n}{32} + \sum_{i=1}^{m}\left(\frac{n-i+1}{n}\right) - \sum_{i=1}^{m}\left(\frac{i-1}{n}\right) \\
&= -\frac{n}{16} + m\left(\frac{n+(n-m+1)}{2n}\right) - \frac{m(m-1)}{2n} \\
&= -\frac{n}{16} + \frac{m}{2n}\left(2n - m + 1 - m + 1\right) \\
&= -\frac{n}{16} + \frac{m(n-m+1)}{n} \\
&\geq -\frac{n}{16} + \frac{3n}{16} = \frac{n}{8},
\end{aligned}$$

with the last inequality holding since $\frac{n}{4} \leq m \leq \frac{3n}{4}$. This concludes our proof since we have shown $Z \geq \frac{n}{8}$ with the desired probability.

□

We now apply Lemma C.14 to bound the probability that $S \sim p_T^\kappa$ covers T.

Lemma C.15. *Let $T \subset [2\kappa]$ be a set of κ indices, and let $S \sim p_T^\kappa$ be a set of κ i.i.d points . Then with probability at least $1 - 4\exp\left(-\frac{\kappa}{2048}\right)$, S covers T.*

Proof. Let $S = (x_1, x_2, \ldots, x_\kappa)$, and let $A = (a_1, a_2, \ldots, a_\kappa)$ be the unique indices such that $x_i \in a_i$. By Definition C.10, L and L' are the number of values in T and $[2\kappa] \setminus T$ that appear exactly once in A. We desire to bound the probability that $|L| \geq \frac{\kappa}{8}$ and $|L'| \geq \frac{\kappa}{8}$. To do so, the key idea is to condition on M, which we define as the number of $1 \leq i \leq \kappa$ such that $a_i \in T$.

Suppose that $M = m$. Observe that the conditional distribution of A (viewed as a multiset) given $M = m$ is precisely the distribution obtained by selecting m indices at uniform from T and $\kappa - m$ indices at uniform from $[2\kappa] \setminus m$. This holds because p_T is uniform when restricted to $\cup_{i \in T} C_i$ or $\cup_{i \in [2\kappa]\setminus T} C_i$. Suppose that $\frac{\kappa}{4} \leq m \leq \frac{3\kappa}{4}$. Then the same must hold for $\kappa - m$. it follows by applying Lemma C.14 to selecting m indices from T and $\kappa - m$ indices from $[2\kappa] \setminus T$ that with probability at least $1 - 2\exp\left(-\frac{\kappa}{2048}\right)$ that $|L| \geq \frac{\kappa}{8}$ and $|L|' \geq \frac{\kappa}{8}$. Thus, by summing over all such m, we see that

$$\begin{aligned}
\Pr_{S \sim p_T^\kappa}[|L| \geq \frac{\kappa}{8}, |L'| \geq \frac{\kappa}{8}] &= \sum_{m=1}^{\kappa} \Pr_{S \sim p_T^\kappa}(M = m) \Pr[|L| \geq \frac{\kappa}{8}, |L'| \geq \frac{\kappa}{8} | M = m] \\
&\geq \sum_{m=\kappa/4}^{3\kappa/4} \Pr_{S \sim p_T^\kappa}(M = m) \Pr[|L| \geq \frac{\kappa}{8}, |L'| \geq \frac{\kappa}{8} | M = m] \\
&\geq \sum_{m=\kappa/4}^{3\kappa/4} \Pr_{S \sim p_T^\kappa}(M = m) \left(1 - 2\exp\left(-\frac{\kappa}{2048}\right)\right) \\
&= \left(1 - 2\exp\left(-\frac{\kappa}{2048}\right)\right) \Pr_{S \sim p_T^\kappa}[\frac{\kappa}{4} \leq M \leq \frac{3\kappa}{4}].
\end{aligned}$$

To bound the latter probability, we simply apply a Chernoff bound, as $M = \sum_{i=1}^{\kappa} \mathbb{1}(a_i \in T)$ is the sum of κ independent indicator variables each with expected value $\frac{1}{3}$. Using a two sided Chernoff bound, we see that $\Pr[\frac{\kappa}{4} \leq M \leq \frac{3\kappa}{4}] \geq 1 - 2\exp\left(-\frac{\kappa}{144}\right)$. Substituting this, it follows that

$$\Pr_{S \sim p_t^\kappa}[|L| \geq \frac{\kappa}{8}, |L'| \geq \frac{\kappa}{8}] \geq \left(1 - 2\exp\left(-\frac{\kappa}{2048}\right)\right)\left(1 - 2\exp\left(-\frac{\kappa}{144}\right)\right) \geq 1 - 4\exp\left(-\frac{\kappa}{2048}\right).$$

□

Step 3: Constructing $\mathcal{F}$ and $\mathcal{F}'$

We start by defining $\mathcal{F}$ and $\mathcal{F}'$ as distributions of pairs (p, A) where p is a data distribution and A is a generative algorithm.

Definition C.16. $\mathcal{F}$ and $\mathcal{F}'$ are the uniform distributions over $\{(p_T, A_T) : T \subset [2\kappa], |T| = \kappa\}$ and $\{(p_T, A'_T) : T \subset [2\kappa], |T| = \kappa\}$ respectively.

Next, we use $\mathcal{F}$ and $\mathcal{F}'$ to construct distributions Q and Q' over pairs (S, q), where S is a set of points, and q is generated distribution.

Definition C.17. Let Q be the distribution of (S, q) where $(p_T, A_T) \sim \mathcal{F}$, $S \sim p_T^\kappa$, and $q \sim A_T(S)$. Similarly, let Q' be the distribution of (S, q) where $(p_T, A'_T) \sim \mathcal{F}'$, $S \sim p_T^\kappa$, and $q \sim A'_T(S)$.

Our goal will be to show that Q and Q' follow similar distributions. Our strategy will be to show that for the majority of (S, q) in their supports, they have similar probability masses. To this end, we first characterize the values of (S, q) that we are interested in considering.

Definition C.18. We say that (S, q) is nice if S is a sample of points from some p_T, and q is a generated distribution from either A_T or A'_T that has no support over C_0. More precisely, (S, q) is nice if the following conditions hold:

1. $S \subset \cup_{i=1}^{2\kappa} C_i$, with $|S| = \kappa$.
2. There exists a set of $\frac{\kappa}{8}$ distinct indices, $L_* \subset [2\kappa]$, such that for $0 \leq i \leq 2\kappa$,

$$q(C_i) = \begin{cases} \frac{\lambda(1+\epsilon)}{3\kappa} & i \in L_* \\ 0 & i \in [2\kappa] \setminus L_* \\ 1 - \frac{\lambda(1+\epsilon)}{24} & i = 0 \end{cases}$$

3. For every $i \in L_*$, $|S \cap C_i| = 1$, meaning exactly one element from S is in C_i.

We now prove a quick lemma relating nice pairs to instances in which S covers T.

Lemma C.19. *Let $T \subset [2\kappa]$ satisfy $|T| = \kappa$. Let $S \sim p_T^\kappa$ and let q and q' be generated distributions with $q = A_T(S)$ and $q' = A'_T(S)$. Then the following three are equivalent:*

1. *(S, q) is nice.*
2. *(S, q') is nice.*
3. *S covers T.*

Proof. Suppose S covers T. Then the sets L and L' (Definition C.10) each have size at least $\frac{\kappa}{8}$ implying that when running A_T or A'_T, the set L_* will be non-trivial. This in turn will imply that (q, S) and (q', S) are nice, regardless of the choice of L_*.

Otherwise, suppose S does not cover T. Then by Definition C.11, $A_T(S)$ and $A_{T'}(S)$ will both be the uniform distribution over C_0 thus violating Definition C.18. □

We now show that Q and Q' assign identical probability masses to nice pairs.

Lemma C.20. *Let (S, q) be a nice pair. Then $Q(S, q) = Q'(S, q)$ with these expressions denoting the probability that (S, q) is chosen over Q and Q' respectively.*

Proof. Let $S = \{x_1, x_2, \ldots, x_\kappa\}$. Let M denote the set of indices in $\{1, 2 \ldots, 2\kappa\}$ such that exactly one point of S lies in the corresponding circle. That is, $M = \{i : |S \cap C_i| = 1\}$. Let L_* be the set of indices in $\{1, 2, \ldots 2\kappa\}$ where q assigns non-trivial probability mass to the corresponding circle. That is, $L_* = \{i : q(C_i) > 0, 1 \leq i \leq 2\kappa\}$. Since (S, q) is a nice pair (Definition C.18), L_* is a subset of M, and satisfies $|L_*| = \frac{\kappa}{8}$. Furthermore, q is uniquely determined by L_*.

We now compute $Q(S, q)$ and $Q'(S, q)$ by summing the conditional probabilities of (S, q) given (p_T, A_T) and (p_T, A'_T) respectively as T ranges over all subsets. By utilizing the fact that (S, q) is nice (meaning it can only occur if S covers T) along with the definition of A_T, we have that

$$\begin{aligned}
Q(S, q) &= \sum_{|T|=\kappa : T \subset [2\kappa]} \frac{1}{\binom{2\kappa}{\kappa}} \Pr[(S, q)|p_T, A_T] \\
&= \sum_{|T|=\kappa : T \subset [2\kappa]} \frac{1}{\binom{2\kappa}{\kappa}} \Pr[S|p_T] \Pr[A_T(S) = q|S, T] \\
&= \sum_{T : S \text{ covers } T} \frac{1}{\binom{2\kappa}{\kappa}} \Pr[S|p_T] \Pr[A_T(S) = q|S, T]. \\
&= \sum_{T : S \text{ covers } T} \frac{1}{\binom{2\kappa}{\kappa}} \Pr[S|p_T] \frac{\mathbb{1}\left(L_* \subseteq T\right)}{\binom{|T \cap M|}{\kappa/8}}.
\end{aligned}$$

with the last equality holding because $A_T(S)$ randomly chooses a $\kappa/8$ element subset of $T \cap M$ for the support of q (see Definition C.11). The term $\mathbb{1}(L_* \subseteq T)$ is necessary because if $L_* \not\subseteq T$, then it is impossible for it to be chosen making the probability 0.

Similarly, letting T^c denote the complement of T, we have

$$Q'(S, q) = \sum_{T : S \text{ covers } T} \frac{1}{\binom{2\kappa}{\kappa}} \Pr[S|p_T] \frac{\mathbb{1}\left(L_* \subseteq T^c\right)}{\binom{|T^c \cap M|}{\kappa/8}},$$

with the only real difference being the support is chosen from $T^c \cap M$ rather than $T \cap M$.

To show that these sums are equal, we will further group the sums by using M to define an equivalence relation over $\{T : T \subset [2\kappa], |T| = \kappa\}$. For $T_1, T_2 \subset [2\kappa]$, we say they are equivalent if their intersections with $[2\kappa] \setminus M$, the complement of M, are equal. That is,

$$T_1 \sim T_2 \iff T_1 \cap ([2\kappa] \setminus M) = T_2 \cap ([2\kappa] \setminus M).$$

The usefulness of this equivalence relation is in the following claim.

Claim: Let $T_1 \sim T_2$ be equivalent subsets of κ indices. Then the following hold:

1. $\Pr[S|p_{T_1}] = \Pr[S|p_{T_2}]$.
2. $|T_1 \cap M| = |T_2 \cap M|$ and $T_1^c \cap M| = |T_2^c \cap M|$.
3. S covers T_1 if and only if S covers T_2.

Proof. (Of Claim) Let T be any set of indices, let $S = \{x_1, x_2, \ldots, x_\kappa\}$, and let $a_1, a_2, \ldots a_\kappa$ denote the respective indices of the circles that $x_1, \ldots, x_\kappa$ are on. Without loss of generality (relabeling if necessary), suppose that $a_1, a_2, \ldots, a_m$ are the unique indices that constitute M (defined above).

Since p_T has probability mass $\frac{1}{3\kappa}$ on every index in T and $\frac{2}{3\kappa}$ on the others, we have that the probability density of S

(denoted $\Pr[S|p_{T_1}]$) satisfies,

$$\begin{aligned}\Pr[S|p_T] &= \prod_{i=1}^{\kappa} \frac{2-\mathbb{1}(a_i \in T)}{\kappa}\frac{1}{2\pi} \\ &= \left(\prod_{i=1}^{m} \frac{2-\mathbb{1}(a_i \in T\cap M)}{\kappa}\frac{1}{2\pi}\right)\left(\prod_{i=m+1}^{\kappa} \frac{2-\mathbb{1}(a_i \in T\cap([2\kappa]\setminus M))}{\kappa}\frac{1}{2\pi}\right) \\ &= \left(\frac{2^{|T^c\cap M|}}{(2\pi\kappa)^m}\right)\left(\prod_{i=m+1}^{\kappa} \frac{2-\mathbb{1}(a_i \in T\cap([2\kappa]\setminus M))}{\kappa}\frac{1}{2\pi}\right),\end{aligned}$$

with the last equality exploiting the fact that $\{a_1, a_2, \ldots, a_m\}$ precisely equals M (by the definition of M). Next, observe that if $T_1 \sim T_2$, then by definition, $T_1 \cap [2\kappa] \setminus M = T_2 \cap [2\kappa \setminus M$ implying that the second part of the product is equal. However, since $|T_1| = |T_2| = \kappa$, the first part must be equal as well, as $|T^c \cap M| = \kappa - |T^c \cap [2\kappa] \setminus M|$. It follows that the probability densities are the same. Note that this observation also implies the second claim, that $|T_1 \cap M| = |T_2 \cap M|$ and $T_1^c \cap M| = |T_2^c \cap M|$

Finally, to show the second part of the claim, we simply observe that for a set T, the sets L and L' from Definition C.10 are precisely $T \cap M$ And $T^c \cap M$. For $T = T_1, T_2$, by the second claim, these have equal sizes. □

We now return to the proof of Lemma C.20. Having shown the claim, we now return to our original computation. Let $T_1, T_2, \ldots, T_r$ denote sets of κ indices with $[T_1], [T_2], \ldots, [T_r]$ denoting their respective equivalence classes such that $[T_1], \ldots, [T_r]$ partition $\{T : S \text{ covers } T\}$. This is possible from the third part of our claim.

For $1 \le i \le r$, let $m_i = |T_i \cap M|$ and $m_i' = |T_i^c \cap M|$ where T_i^c denotes the complement of T_i. It follows from second part of our claim that $|T \cap M|, |T^c \cap M|$ both equal m_i as well for all $T \in [T_i]$.

By partitioning our sum for $Q(S, q)$ in using $[T_1], \ldots, [T_r]$, we have

$$\begin{aligned}Q(S,q) &= \sum_{T:S \text{ covers } T} \frac{1}{\binom{2\kappa}{\kappa}} \Pr[S|p_T] \frac{\mathbb{1}\left(L_* \subseteq T\right)}{\binom{|T\cap M|}{\kappa/8}} \\ &= \sum_{i=1}^{r} \sum_{T\in[T_i]} \frac{1}{\binom{2\kappa}{\kappa}} \Pr[S|p_T] \frac{\mathbb{1}\left(L_* \subseteq T\right)}{\binom{|T\cap M|}{\kappa/8}} \\ &= \sum_{i=1}^{r} \frac{\Pr[S|p_{T_i}]}{\binom{2\kappa}{\kappa}} \sum_{T\in[T_i]} \frac{\mathbb{1}\left(L_* \subseteq T\right)}{\binom{m_i}{\kappa/8}} \\ &= \sum_{i=1}^{r} \frac{\Pr[S|p_{T_i}]}{\binom{2\kappa}{\kappa}} \frac{\binom{m-\kappa/8}{m_i-\kappa/8}}{\binom{m_i}{\kappa/8}},\end{aligned}$$

with the last equality coming by counting the number of $T \in [T_i]$ such that $L_* \subseteq T$. This counting problem essentially forces all $\kappa/8$ elements in L_* to be in T leaving us to choose the remaining elements in M that can be part of T.

By using the exact same line of reasoning for $Q'(S, q)$, we have

$$\begin{aligned}Q'(S,q) &= \sum_{T:S \text{ covers } T} \frac{1}{\binom{2\kappa}{\kappa}} \Pr[S|p_T] \frac{\mathbb{1}\left(L_* \subseteq T^c\right)}{\binom{|T^c\cap M|}{\kappa/8}} \\ &= \sum_{i=1}^{r} \sum_{T\in[T_i]} \frac{1}{\binom{2\kappa}{\kappa}} \Pr[S|p_T] \frac{\mathbb{1}\left(L_* \subseteq T^c\right)}{\binom{|T^c\cap M|}{\kappa/8}} \\ &= \sum_{i=1}^{r} \frac{\Pr[S|p_{T_i}]}{\binom{2\kappa}{\kappa}} \sum_{T\in[T_i]} \frac{\mathbb{1}\left(L_* \subseteq T^c\right)}{\binom{m_i'}{\kappa/8}} \\ &= \sum_{i=1}^{r} \frac{\Pr[S|p_{T_i}]}{\binom{2\kappa}{\kappa}} \frac{\binom{m-\kappa/8}{m_i'-\kappa/8}}{\binom{m_i'}{\kappa/8}},\end{aligned}$$

Here the only difference ends up being that we use m_i' instead of m_i since we have effectively replaced T with T^c. However, this replacement only takes place for q, the component of the probability that deals with S is identical for both Q and Q'.

Finally, based on these equations, it suffices to show that $\frac{\binom{m-\kappa/8}{m_i'-\kappa/8}}{\binom{m_i'}{\kappa/8}} = \frac{\binom{m-\kappa/8}{m_i-\kappa/8}}{\binom{m_i}{\kappa/8}}$. To do so, since $m_i = |T_i \cap M|$ and $m_i' = |T_i^c \cap M|$, it follows that $m_i + m_i' = m$. Using this, we have that

$$\begin{aligned}\frac{\binom{m-\kappa/8}{m_i-\kappa/8}}{\binom{m_i}{\kappa/8}} &= \frac{(m-\kappa/8)!\,(\kappa/8)!(m_i-\kappa/8)!}{(m_i-\kappa/8)!(m-m_i)!m_i!} \\ &= \frac{(m-\kappa/8)!(\kappa/8)!}{m_i'!m_i!}.\end{aligned}$$

Applying the same manipulation to $\frac{\binom{m-\kappa/8}{m_i'-\kappa/8}}{\binom{m_i'}{\kappa/8}}$ completes the proof.

□

Step 4: finishing the overall proof.

Let κ be a sufficiently large integer. It suffices to show that there exists a probability distribution p with $\frac{1}{9\kappa} \le p_\epsilon \le \frac{2}{3\kappa}$ such that $m_p(\epsilon,\delta) > \kappa$. Assume towards a contradiction that no such p exists, meaning that $m_p(\epsilon,\delta) \le \kappa$ for all p satisfying the above.

Let $T \subset [2\kappa]$ satisfy $T = [2\kappa]$. By Lemma C.9, $\frac{1}{9\kappa} \le (p_T)_\epsilon \le \frac{2}{3\kappa}$. It follows that with probability at least $1-\delta$ over $S \sim p_T^\kappa$ and $q \sim A_T(S)$ along with the randomness of B,

$$cr_q^{-\epsilon} - \epsilon \le B(S,q) \le cr_q^{\epsilon} + \epsilon,$$

with $cr_q^{-\epsilon}, cr_q^{\epsilon}$ denoting the appropriate data-copying rates for q with respect to p.

By Lemma C.13, if S covers T, then $cr_q^{-\epsilon} \ge \frac{\lambda(1+\epsilon)}{24} = \frac{13\frac{4}{3}}{24} > \frac{2}{3}$. By Lemma C.15, S covers T with probability at least $1 - 4\exp\left(-\frac{\kappa}{2048}\right)$. Substituting this, we have

$$\begin{aligned}1-\delta &\le \mathbb{E}_{S\sim p_T^\kappa}\mathbb{E}_{q\sim A_T(S)}\mathbb{E}_B \mathbb{1}\left(B(S,q) \ge cr_q^{-\epsilon} - \epsilon\right) \\ &= \mathbb{E}_{S\sim p_T^\kappa}\mathbb{1}\left(S \text{ does not cover } T\right) + \mathbb{E}_{S\sim p_T^\kappa}\mathbb{1}\left(S \text{ covers } T\right)\mathbb{E}_{q\sim A_T(S)}\mathbb{E}_B\mathbb{1}\left(B(S,q) > \frac{1}{3}\right) \\ &\le 4\exp\left(-\frac{\kappa}{2048}\right) + \mathbb{E}_{S\sim p_T^\kappa}\mathbb{1}\left(S \text{ covers } T\right)\mathbb{E}_{q\sim A_T(S)}\mathbb{E}_B\mathbb{1}\left(B(S,q) > \frac{1}{3}\right),\end{aligned}$$

with the substitutions for $cr_q^{-\epsilon} - \epsilon$ utilizing that $\epsilon = \frac{1}{3}$.

Applying this over the distribution, $\mathcal{F}$ (Definition C.16), which comprises of all (p_T, A_T) with T chosen at uniform over all subsets of size κ, and then substituting the definition of Q (Definition C.17), we have

$$\begin{aligned}1-\delta-4\exp\left(-\frac{\kappa}{2048}\right) &\le \mathbb{E}_{(p_T,A_T)\sim\mathcal{F}}\mathbb{E}_{S\sim p_T^\kappa}\mathbb{E}_{q\sim A_T(S)}\mathbb{1}\left(S \text{ covers } T\right)\mathbb{E}_B\mathbb{1}\left(B(S,q) > \frac{1}{3}\right) \\ &= \mathbb{E}_{(S,q)\sim Q}\mathbb{1}\left((S,q) \text{ is nice}\right)\mathbb{E}_B\mathbb{1}\left(B(S,q) > \frac{1}{3}\right) \\ &= \frac{1}{1-\Pr_{(S,q)\sim Q}[(S,q) \text{ is not nice}]}\mathbb{E}_{(S,q)\sim Q_*}\mathbb{E}_B\mathbb{1}\left(B(S,q) > \frac{1}{3}\right) \\ &\le \mathbb{E}_{(S,q)\sim Q_*}\mathbb{E}_B\mathbb{1}\left(B(S,q) > \frac{1}{3}\right),\end{aligned} \tag{27}$$

where Q_* denotes the marginal distribution of Q over all nice (Definition C.18) pairs (S,q). Note that the manipulation above holds because of Lemma C.19, which implies that (S,q) is nice if and only if S covers T.

Next, we apply the same exact reasoning to the pair (p_T, A'_T). To this end, we have that with probability at least $1-\delta$ over $S \sim p_T^\kappa$, $q \sim A'_T(S)$, along with the randomness of B,

$$cr_q^{-\epsilon} - \epsilon \leq D(S,q) \leq cr_q^{\epsilon} + \epsilon.$$

By Lemma C.13, if S covers T, then $cr_q^\epsilon = 0$. Applying the same argument as above using Lemma C.15, we have that

$$1 - \delta - 4\exp\left(-\frac{\kappa}{2048}\right) \leq \mathbb{E}_{S\sim p_T^\kappa} \mathbb{1}\left(S \text{ covers } T\right) \mathbb{E}_{q\sim A'_T(S)} \mathbb{E}_B \mathbb{1}\left(B(S,q) \leq \frac{1}{3}\right).$$

Applying this over the distribution $\mathcal{F}'$ (Definition C.16) and using a similar set of manipulations as we did with $\mathcal{F}$ and Q, we have that

$$\begin{aligned} 1 - \delta - 4\exp\left(-\frac{\kappa}{2048}\right) &\leq \mathbb{E}_{(p_T, A'_T)\sim\mathcal{F}'} \mathbb{E}_{S\sim p_T^\kappa} \mathbb{E}_{q\sim A'_T(S)} \mathbb{1}\left(S \text{ covers } T\right) \mathbb{E}_B \mathbb{1}\left(B(S,q) \leq \frac{1}{3}\right) \\ &\leq \mathbb{E}_{(S,q)\sim Q'_*} \mathbb{E}_B \mathbb{1}\left(B(S,q) \leq \frac{1}{3}\right), \end{aligned} \tag{28}$$

where Q'_* denotes the marginal distribution of Q' over nice pairs (S,q).

Finally, by Lemma C.20, Q'_* and Q_* follow the exact same distribution. This means that summing equations 27 and 28, we can combine the summands *inside* the expectation giving us that

$$2 - 2\delta - 8\exp\left(-\frac{\kappa}{2048}\right) \leq \mathbb{E}_{(S,q)\ simQ_*} \mathbb{E}_B \left(\mathbb{1}\left(B(S,q) > \frac{1}{3}\right) + \mathbb{1}\left(B(S,q) \leq \frac{1}{3}\right)\right) = 1.$$

This gives a contradiction as this equation is clearly false when κ is sufficiently large (as $\delta = \frac{1}{3}$).

□