Can Generative Artificial Intelligence Survive Data Contamination? Theoretical Guarantees under Contaminated Recursive Training

Kevin Wang, Hongqian Niu, Didong Li*
Department of Biostatistics, University of North Carolina at Chapel Hill

Abstract

As artificial intelligence (AI)-generated content proliferates, models are increasingly trained on their own outputs, risking progressive degradation or collapse. In this article, we provide the first positive, rigorous theoretical results, to the best of our knowledge, showing that under model-agnostic mild conditions, the model converges to the true data-generating distribution. The convergence rate is the minimum of the model's intrinsic rate and the fraction of real data at each training iteration, revealing a phase transition between data-limited and model-limited regimes. We further show that, for biased real data, correcting the bias prevents the persistence and amplification of early bias over training iteration. Extensive experiments across simulations, real images and texts validate our theoretical framework, establishing quantitative conditions for long-term AI stability in contaminated environments.

Keywords: Generative AI, recursive training, synthetic data, convergence rates

1 Introduction

Recent analyses suggest that Artificial Intelligence (AI) generated text, images, and code constitute an increasingly large share of online content. Journalistic investigations have documented widespread use of AI-generated text across platforms such as Wikipedia and hobbyist communities (Froio, 2025), as well as coordinated campaigns on social media (BBC News, 2025). In parallel, empirical research has documented the growing presence of AI-generated content in scientific and academic writing (Brooks et al., 2024; Liang et al., 2024).

Consequently, modern generative models are increasingly trained on data streams that contain both human-generated and model-generated content. Such recursive training is already occurring, both unintentionally through web-scale data collection and intentionally in applications that incorporate synthetic data (Bowles et al., 2018; Tobin et al., 2017; Billot et al., 2023; Xie et al., 2018; Papernot et al., 2016; Jordon et al., 2018; Lopes et al., 2017;

*didongli@unc.edu

Yin et al., 2020; Lee et al.; Xie et al., 2020; Arazo et al., 2020). Because synthetic content is often difficult to reliably identify, simply filtering it from training data is unlikely to be practical (Nist, 2024). Synthetic content is therefore becoming an intrinsic part of the digital information ecosystem, influencing both the information encountered by users and the data pipelines that shape future models. The critical question then is: *will recursive training on mixtures of real and synthetic data generated by the model's predecessors lead to progressive degradation, or can learning still improve over time?* This question is central to understanding the statistical behavior of learning algorithms trained on recursively contaminated data.

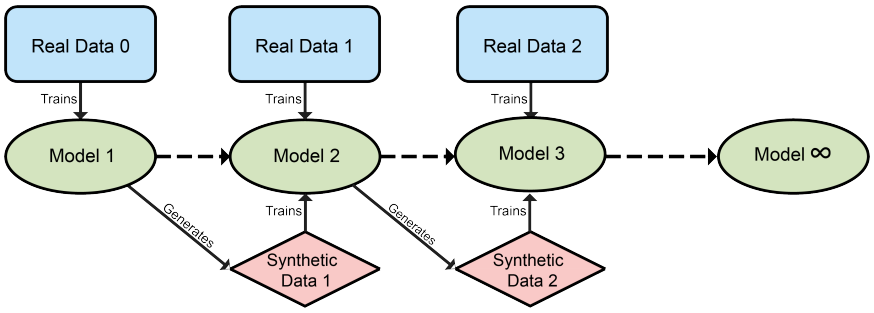


Figure 1: The Contaminated Recursive Training (CRT) framework is illustrated via a diagram. An initial dataset of real data (Real Data 0) trains the first model. After deployment, the model produces synthetic content (Synthetic Data 1) while new real data continue to appear online (Real Data 1). Subsequent models are trained on the accumulated mixture of real and synthetic data.

Recent work has begun to examine the behavior of models trained on synthetic data. Theoretical studies have characterized model collapse (where performance deteriorates and generated outputs become increasingly distorted or less diverse) when training relies entirely on synthetic data (Shumailov et al., 2023, 2024; Suresh et al., 2024). Empirical evidence suggests that incorporating real data at each iteration can mitigate degradation (Zhang et al., 2026), and theoretical analyses support this stabilizing effect in parametric settings (Gerstgrasser et al., 2024; Dohmatob et al., 2024a). However, general theoretical conditions with realistic assumptions under which recursive training converges to the underlying data distribution remain unknown. Without such guarantees, the long-term behavior of models trained on mixtures of real and synthetic data remains fundamentally uncertain.

We formalize this setting as contaminated recursive training (CRT). At each iteration, fresh samples from the target distribution and synthetic samples from the current generator are added to an accumulated training set. A learning procedure is then applied to the full accumulated dataset to produce the next generator. This formulation captures the case in which real data continue to arrive over time, but the training corpus is increasingly contaminated by synthetic data that may not be reliably separated. This framework is illustrated in Figure 1.

In addition to degradation caused by synthetic contamination, recursive training may

also propagate systematic errors arising from bias in real data. Generative models trained on non-representative datasets are known to produce systematically distorted outputs (Mehrabi et al., 2021; Zhou et al., 2024), and such biases have been extensively documented in widely used training corpora (Lucy and Bamman, 2021; Sheng et al., 2019). This raises the concern that when synthetic content is reused for training, biases introduced early in the process may persist or be amplified across successive generations, creating long-term feedback effects. Although numerous methods have been developed to mitigate sampling bias and distribution shift (He and Garcia, 2009; Cortes and Mohri, 2014; Chen et al., 2023b), the long-term behavior of bias in the context of recursive training has not been formally characterized. A central question is *whether early distortions can be corrected over time, or whether they instead become permanently embedded in future models, gradually shaping what people see, read, and rely on online.*

This paper addresses these questions by providing, to the best of our knowledge, the first general theoretical analysis of recursive training that establishes convergence rates to the true data-generating distribution under data contamination. We show that generative models trained on mixtures of real and synthetic data converge to the target distribution without requiring parametric assumptions on the data-generating distribution, relying only on general convergence properties of the underlying learning procedure. Moreover, the convergence rate is determined by the minimum of the convergence rate of the base AI model and the fraction of real data available at each iteration, revealing a phase transition between data-limited and model-limited regimes. We further show that sufficiently accurate bias-correction prevents the persistence and amplification of initial sampling bias. We validate these theoretical findings through extensive experiments across simulations, real images and texts, using generative adversarial networks (GANs), diffusion models, and large language models (LLMs). These experiments confirm the convergence of the recursively trained models as well as the predicted phase transitions, demonstrating that the theoretical results hold in practical, complex settings. Section 2 reviews existing work; Section 3 and 4 contain main results for CRT and BCRT respectively; Section 5-7 presents experimental results. All proofs and additional experimental details are provided in the Appendix.

2 Previous Work

Prior work on recursive training can be broadly divided into three settings: synthetic-only recursion, recursion with continued access to real data, and contaminated training with both real and synthetic data. Most theoretical and empirical work has focused on the synthetic-only setting, where various forms of model collapse have been established. Existing positive convergence results rely on growing the original real dataset and the parametric model setting. In contrast, the contaminated setting, in which new real and synthetic data are introduced simultaneously over time, has received comparatively little theoretical attention despite its relevance to modern generative AI systems.

Theoretically in the synthetic-only setting, Shumailov et al. (2024) analyze iterative training under both discrete and Gaussian models trained via maximum likelihood, proving

that synthetic-only recursive training inevitably leads to model collapse. Their theoretical results are supported by empirical evidence demonstrating severe degradation. Building on this, Suresh et al. (2024) provide explicit rates of collapse in similar settings, quantifying how quickly the learned distribution diverges from the target.

Related work by Shumailov et al. (2023) also considers synthetic-only training and establishes collapse in this regime. Notably, they introduce a "partial-refresh" setting in which a small fraction of the original real data is reintroduced at each iteration. While this slows degradation, it does not prevent eventual collapse. Furthermore, Dohmatob et al. (2024a, 2025) derive explicit rates of performance degradation in recursive training loops that rely solely on synthetic data within parametric model classes. Complementing this line of work, Dohmatob et al. (2024b) characterize model collapse through changes in tail-diversity scaling laws in certain language models, while Feng et al. (2025) show that sufficiently effective verification of synthetic data can mitigate collapse. Related work by Dey and Donoho (2024) studies an asymptotic universality phenomenon in recursive training under accumulated-data settings.

Beyond synthetic-only regimes, several works study recursive training as a form of data contamination, where real data is progressively mixed with model-generated samples. In a two-generation setting, Hataya et al. (2023) show empirically that training a second-generation model on a mixture of real and synthetic data leads to degradation relative to the first-generation model. Extending this to multiple iterations, Martínez et al. (2023) demonstrate that recursively adding synthetic data while keeping the original dataset fixed results in cumulative performance decline. Bertrand et al. (2023) show that when a sufficiently large proportion of the original data is retained, collapse can be avoided entirely, though these analyses assume that no new data is introduced over time. Moreover, Briesch et al. (2023) study a training loop for LLMs where generated synthetic data as well as new real data are repeatedly incorporated into subsequent training sets. Their experiments suggest that semantic coherence is largely preserved while output diversity declines over iterations, though they do not theoretically establish convergence to the true data distribution.

Additional theoretical support for these findings is provided by Gerstgrasser et al. (2024), which analyzes linear models trained via least squares and shows that when maintaining access to the full original dataset, collapse is not inevitable. Barzilai and Shamir (2026) analyze iterative maximum-likelihood estimation for parametric models in an accumulating-data setting and establish conditions under which consistency is preserved, with stability depending on continued growth of the initial real-data sample.

Taken together, existing theoretical analyses either focus on synthetic-only recursion or establish stability in restricted accumulating-data settings, typically under parametric assumptions and continued access to a growing corpus of real data. By contrast, our setting allows fresh real and synthetic samples to arrive simultaneously, permits accumulation of all historical data, and accommodates general target distributions and generators. The next section introduces this framework formally and develops corresponding convergence guarantees.

3 Recursive Training under Data Contamination

In this section we begin by rigorously defining the generative models and the recursive training under data contamination scheme, and present theorems for the theoretical convergence under such frameworks.

Definition 3.1 (Generative model)**.** Let $\mathbb{P}_0$ be the target distribution. Given a model class $\mathcal{G}$ and a distance metric d between distributions, the goal is to learn a generative model $\mathrm{G} \in \mathcal{G}$ such that

$$\operatorname*{argmin}_{\mathrm{G}\in\mathcal{G}} d(\mathrm{G}, \mathbb{P}_0).$$

Typically, one will not have access to $\mathbb{P}_0$ directly, but rather through n i.i.d. samples from $\mathbb{P}_0$. In this case, while the goal of minimizing the distance remains the same, the algorithm for learning the generator will often not minimize this quantity directly as its objective but some empirical counterpart.

Definition 3.2 (Convergence rate)**.** Let $\widehat{\mathbb{P}}_n$ be an estimated distribution from a generative model $\mathcal{G}$ trained from n i.i.d. samples drawn from $\mathbb{P}_0$. We say that $\widehat{\mathbb{P}}_n$ converges at a (polynomial) rate $p > 0$ if $d(\widehat{\mathbb{P}}_n, \mathbb{P}_0) \lesssim n^{-p}$, i.e., there exists a constant $C > 0$ such that $d(\widehat{\mathbb{P}}_n, \mathbb{P}_0) \leq Cn^{-p}$.

In this paper, when referring to setting where the model is trained with no contamination, we call the convergence rate the baseline rate. For generality, we leave the method of finding the $\widehat{\mathbb{P}}_n$ unspecified, as our theory does not depend on the specific model but only on its convergence rate p. Moreover, if we change the notion to convergence in probability, all theorems hold in probability as well. Next, we introduce our setting of recursive training under data contamination.

Definition 3.3 (Contaminated Recursive Training (CRT))**.** Let X_0 be an initial dataset with m_1 i.i.d. samples drawn from $\mathbb{P}_0$. Let the initial estimator $\widehat{\mathbb{P}}_0$ be trained on X_0. For each step $t \geq 1$, perform the following operations:

1. **Recursive generation.** From the previous estimator $\widehat{\mathbb{P}}_{t-1}$, draw an i.i.d. synthetic sample Y_t of size m_2. Also draw a new i.i.d. real dataset X_t of size m_1 from $\mathbb{P}_0$. Let $\alpha \coloneqq \frac{m_1}{m_1+m_2} \in (0,1)$ be the real-data fraction.
2. **Contaminated data accumulation.** Accumulate all real datasets $X_0, \ldots, X_t$ and all previously generated synthetic datasets $Y_1, \ldots, Y_t$ into one dataset.
3. **Recursive update.** Train the learner on the accumulated data, producing the new generator $\widehat{\mathbb{P}}_t$.

$\{\widehat{\mathbb{P}}_t\}_{t\geq 0}$ is called the *contaminated recursively trained* (CRT) sequence of generators.

Comparison to prior recursive training settings. Our setup differs from the synthetic-only recursive schemes studied in the model-collapse literature (Shumailov et al., 2024, 2023; Suresh et al., 2024; Dohmatob et al., 2025), where each new model is trained primarily on synthetic samples and real data from earlier rounds is discarded. Such synthetic-only recursion drives the training distribution progressively away from the target distribution and leads to collapse. Our setting also differs from replacement-style contamination models (Hataya et al., 2023), where each iteration trains on a mixture of real and synthetic data but without accumulating all past real samples. In contrast, both real and synthetic data are continuously uploaded to the internet. Additionally, internet archives retain essentially all data that has been uploaded, both real and synthetic. Thus, it is imperative to study the setting where each iteration's generator is trained on a new sample of real data, a new sample of synthetic data from the most recent generator, and the running data from all previous iterations.

We then state the following two standard assumptions for our theoretical results.

Assumption A1 (Polynomial rate for baseline generative models). Let $\mathcal{Q}$ be a convex class of distributions. Then there exists a constant $0 < M < \infty$ such that for any $\mathbb{P}_0 \in \mathcal{Q}$ the baseline generative model based on a sample of size n satisfies $d(\widehat{\mathbb{P}}_n, \mathbb{P}_0) \leq Mn^{-p}$.

Assumption A2 (Convex distance metric). The distance metric $d(\cdot, \cdot)$ between distributions is convex, i.e., for any distributions P, Q_1, Q_2, and any scalars $\lambda_1, \lambda_2 > 0$ with $\lambda_1 + \lambda_2 = 1$,

$$d(P,\ \lambda_1 Q_1 + \lambda_2 Q_2) \leq \lambda_1\, d(P, Q_1) + \lambda_2 d(P, Q_2).$$

We emphasize that both assumptions are relatively weak in the literature and are satisfied under standard conditions commonly used in the theory of generative modeling. For example, Assumption A1 requires a uniform polynomial convergence rate over a distribution class $\mathcal{Q}$ that is closed under convex combinations. Such uniform guarantees are classical for estimators such as kernel density estimators (KDE, Devroye, 1985) and Dirichlet process mixture models (DPMM, Ghosal and Van der Vaart, 2017). Moreover, modern deep generative models, including Variational AutoEncoders (VAE, Chérief-Abdellatif et al., 2022; Mbacke et al., 2023), Generative Adversarial Networks (GAN, Uppal et al., 2019; Puchkin et al., 2024), Diffusion Models (De Bortoli, 2022; Oko et al., 2023; Chen et al., 2023a; Tang et al., 2024), and certain classes of Large Language Models (Lotfi et al., 2024; Rawat et al., 2024), are also known to satisfy such uniform convergence guarantees when the generator architecture is subject to standard regularity constraints. Finally, Assumption A2 is satisfied by most widely used statistical distances, including total variation, Kolmogorov–Smirnov, Wasserstein-p, energy distance, maximum mean discrepancy (MMD), and other integral probability metrics.

With these two assumptions, we present our first main theorem about convergence rate of CRT.

Theorem 3.4 (Convergence rate under CRT)**.** *Suppose Assumption A1 and A2 hold. Let*

$\{\widehat{\mathbb{P}}_t\}_{t\geq 0}$ *be the sequence of CRT trained generators, then*

$$d(\widehat{\mathbb{P}}_t, \mathbb{P}_0) \lesssim \begin{cases} t^{-\alpha}, & p > \alpha, \\ t^{-\alpha} \log t, & p = \alpha, \\ t^{-p}, & p < \alpha, \end{cases}$$

Equivalently, up to logarithms,

$$d(\widehat{\mathbb{P}}_t, \mathbb{P}_0) \lesssim t^{-\min\{p,\, \alpha\}}.$$

Additionally, if Assumption A1 is weakened to convergence in probability, the same rates hold in probability.

This theorem shows that should the real-data fraction α be greater than the baseline rate p, then we may achieve the same convergence rate as the baseline. However, if the real-data fraction α is smaller than the baseline rate p, the convergence rate is slowed down. In the case where they are equal ($p = \alpha$), we see a phase transition.

4 Biased Recursive Training under Data Contamination

In a closely related problem, we consider the case where the sampling distribution of the real data may be biased to begin with. Because generative models trained on biased datasets exhibit biased generation, this poses a significant risk in real-world settings. Furthermore, recursively contaminated training processes may further propagate errors present in the initial model stemming from this biased sampling for the training data, which subsequently poses two important questions: what happens when early model iterations are trained from biased data, and can such initially biased models be corrected in future rounds of training? We find that such questions can be studied by naturally extending our framework to the case we now refer to as biased contaminated recursive training (BCRT). In essence, one might imagine that while the original data may come from a biased source, over time these biases may be recognized and slowly corrected in the subsequent real samples. Below we study sufficient conditions on the sampling distribution in the recursive training paradigm to yield convergence as in the previous section under these circumstances.

Definition 4.1 (Biased contaminated recursive training (BCRT))**.** Let $(\mathbb{P}_t^{\text{bias}})_{t\geq 0}$ be a sequence of (possibly biased) distributions, and $\mathbb{P}_0$ be our target distribution. Let X_0 be an initial sample drawn from $\mathbb{P}_0^{\text{bias}}$ and the initial estimator be $\widehat{\mathbb{P}}_0$. For each step $t \geq 1$, perform the following operations:

1. **Recursive generation.** From the previous estimator $\widehat{\mathbb{P}}_{t-1}$, draw an i.i.d. synthetic sample Y_t of size m_2. Also draw an i.i.d. real dataset X_t of size m_1 from $\mathbb{P}_t^{\text{bias}}$. Let $\alpha \coloneqq \frac{m_1}{m_1+m_2} \in (0,1)$ be the real-data fraction.
2. **Biased contaminated data accumulation.** Accumulate all real datasets $X_0, \ldots, X_t$ and all previously generated synthetic datasets $Y_1, \ldots, Y_t$ into one dataset.

3. **Recursive update.** Train the learner on the accumulated hybrid data, producing the new generator $\widehat{\mathbb{P}}_t$.

The sequence $\{\widehat{\mathbb{P}}_t\}_{t\geq 0}$ is called the *biased contaminated recursively trained* (BCRT) sequence of generators.

As a direct consequence of Theorem 3.4, we can characterize what happens when the real samples are drawn from a fixed biased distribution and no effort is made to improve the sampling procedure or correct the bias. In this case, the recursive contamination process simply converges to the biased distribution itself. That is, the procedure will indeed converge, but to the biased distribution rather than the true one.

Corollary 4.2 (Convergence to a biased distribution)**.** *If the real datasets X_t are drawn i.i.d. from a fixed biased distribution $\mathbb{P}_t^{\text{bias}} = \mathbb{P}_0^{\text{bias}} \neq \mathbb{P}_0$ and Assumption A1 and A2 are satisfied, then the recursive procedure converges to $\mathbb{P}_0^{\text{bias}}$ rather than $\mathbb{P}_0$, with the same rates as in Theorem 3.4.*

In many realistic settings, practitioners apply bias-correction techniques or improved sampling strategies (see, e.g., (He and Garcia, 2009; Cortes and Mohri, 2014; Chen et al., 2023b) so that the real-data distribution evolves over time and gradually approaches the desired target distribution. When such corrections are applied across iterations, it is not obvious whether the recursive contamination process still converges, and if so, at what rate and to which limit. The following result addresses this setting by analyzing a biased recursive contamination process in which the real-data distributions move toward the true distribution over time. To account for this, we add an assumption on the sequence of biased distributions.

Assumption A3 (Bias decay). The biased distributions $\mathbb{P}_t^{\text{bias}} \in \mathcal{Q}$ and the bias decays at a polynomial rate q, i.e.,

$$d\Big(\mathbb{P}_t^{\text{bias}}, \mathbb{P}_0\Big) \lesssim t^{-q}.$$

This assumption captures the case where the practitioner performs any of, or any combination of: (1) improving their sampling procedures, (2) implementing bias-correction methods, or (3) working with subsequent distributions that are naturally unbiased relative to the target distribution. We note that the plausibility of this assumption in the first two cases is supported by the biased sampling and covariate shift literature, such as (He and Garcia, 2009; Cortes and Mohri, 2014; Chen et al., 2023b). We also emphasize that, in these, the rate is not necessarily determined by the model itself, but rather by the quality of the external adjustments made by the practitioner. The third case covers settings where the assumption is naturally satisfied, such as fine-tuning on a subpopulation of corpora. Additionally, the assumption that $\mathbb{P}_t^{\text{bias}} \in \mathcal{Q}$ simply ensures that Assumption A1 applies to each of the new distributions under study. Our next result shows that Theorem 3.4 can be extended to the BCRT paradigm.

Theorem 4.3 (Convergence rate under BCRT)**.** *Assume that Assumptions A1, A2, and A3 hold. Let $\{\widehat{\mathbb{P}}_t\}_{t\geq 0}$ be the sequence of BCRT trained generators, then*

$$d(\widehat{\mathbb{P}}_t, \mathbb{P}_0) \lesssim \begin{cases} t^{-\alpha}, & \min(p,q) > \alpha, \\ t^{-\alpha} \log t, & \min(p,q) = \alpha, \\ t^{-\min(p,q)}, & \min(p,q) < \alpha. \end{cases}$$

Equivalently, up to logarithms,

$$d(\widehat{\mathbb{P}}_t, \mathbb{P}_0) \lesssim t^{-\min\{p,\, q,\, \alpha\}}.$$

Like before, if either Assumption A1 or A3 are weakened to convergence in probability, the above rates hold in probability.

This theorem shows that the biased setting mirrors the unbiased case, with an additional limitation on the rate determined by how quickly the bias in the real data distribution decays.

5 Simulations

We first conduct simulation studies to validate our theoretical findings. We examined both the CRT Section 3 and BCRT Section 4 settings under controlled conditions in which the target distribution (Figure 3) and thus the baseline rate are known. This design allowed a direct comparison between the convergence rates in Theorem 3.4 and Theorem 4.3, and those observed empirically. We consider three generative models, Empirical Cumulative Distribution Function (ECDF), Kernel Density Estimator (KDE), and Wasserstein Generative Adversarial Network (WGAN). Convergence was evaluated using two discrepancy metrics: Wasserstein–1 (W_1) and maximum mean discrepancy (MMD). For BCRT, we introduce bias into the real-data distribution and allow the bias to decrease over iterations. This design enabled independent control of all three quantities determining the convergence rate: the baseline rate, the real-data fraction, and the bias-correction rate. For further details about the theoretical rates and the simulations, see Appendix Section B and Section D.

We first consider the CRT setting from Section 3 using KDEs with varying bandwidths. The density used is a mixture of two univariate Gaussians with known parameters,

$$\mathbb{P}_0 = w_1 \mathcal{N}(\mu_1, \sigma_1^2) + (1 - w_1)\mathcal{N}(\mu_2, \sigma_2^2),$$

with a density function shown in Figure 2.

The density for the BCRT setting from *Section* 4 is chosen to be the same but with a small biasing perturbation, whose rate of decay we may control:

$$\mathbb{P}_t^{\text{bias}} = (1 - \text{bias}_t)\mathbb{P}_0 + \text{bias}_t \mathbb{P}_{\text{bias}} \mathcal{N}(\mu_3, \sigma_3^2),$$

where $\mathbb{P}_0$ is defined above and $\text{bias}_t = 0.2(t+5)^{-q}$.

For both the CRT and BCRT settings, we present three generators: the empirical cumulative distribution function (ECDF), the standard Gaussian KDE with a bandwidth

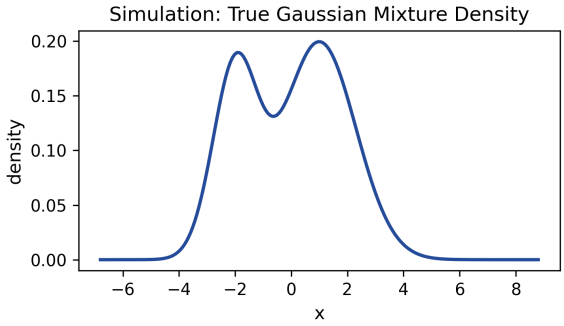


Figure 2: Density function of a two-Gaussian mixture with parameters $w_1 = 0.35$, $(\mu_1, \sigma_1) = (-2.0, 0.8)$, and $(\mu_2, \sigma_2) = (1.0, 1.3)$ used in Simulation of Section 5.1.

decaying on a deterministic schedule, and the WGAN. At each iteration t, we draw a new true sample of 50 real data from $\mathbb{P}_0$ and a synthetic sample of $\frac{1-\alpha}{\alpha}50$ from the previous model, and apply the CRT setting as in Definition 3.3. Then, either an ECDF, KDE, or WGAN is used to form the next model in the sequence and the W_1 and MMD losses from the ground truth are estimated by generating a sample and comparing the quantiles of the generated data to the quantiles of the true distribution on a uniform grid of 200 points. A simple linear fit to the log-log transformed data is then used to estimate the observed order of convergence. Each experiment is performed for 2000 iterations. Each experiment is repeated for 50 replicates and the mean is reported for each distance metric. The results for the CRT setting are summarized in Figure 3A and the results for the BCRT setting are summarized in Figure 3B, where for each $\alpha = \{0.1, 0.2, \ldots, 1\}$ and metric, the rate predicted by our theory and the rate observed in the simulation are compared.

Despite practical factors such as optimization dynamics and finite-sample effects, the empirical results are broadly consistent with the theoretical results. In particular, the observed convergence rates were limited by α in data-sparse regimes and by q when bias improved slowly, reproducing the predicted phase-transition behavior between model-limited, data-limited, and bias-limited regimes. Additional details may be found in Appendix Section B and C.

6 Image Experiments

Thus far, we have demonstrated the empirical rates of convergence under CRT and BCRT in simulated settings where the true distribution $\mathbb{P}_0$ is known and where exact empirical formulations of W_1 and MMD are available in one dimension.

We now consider a more realistic setting by training a state-of-the-art diffusion model on MNIST images. Although Diffusion Models are known to converge under W_1 and MMD loss and therefore fit within the CRT framework of Section 3, observing the precise theoretical rates in practice is challenging due to complex neural network optimization dynamics,

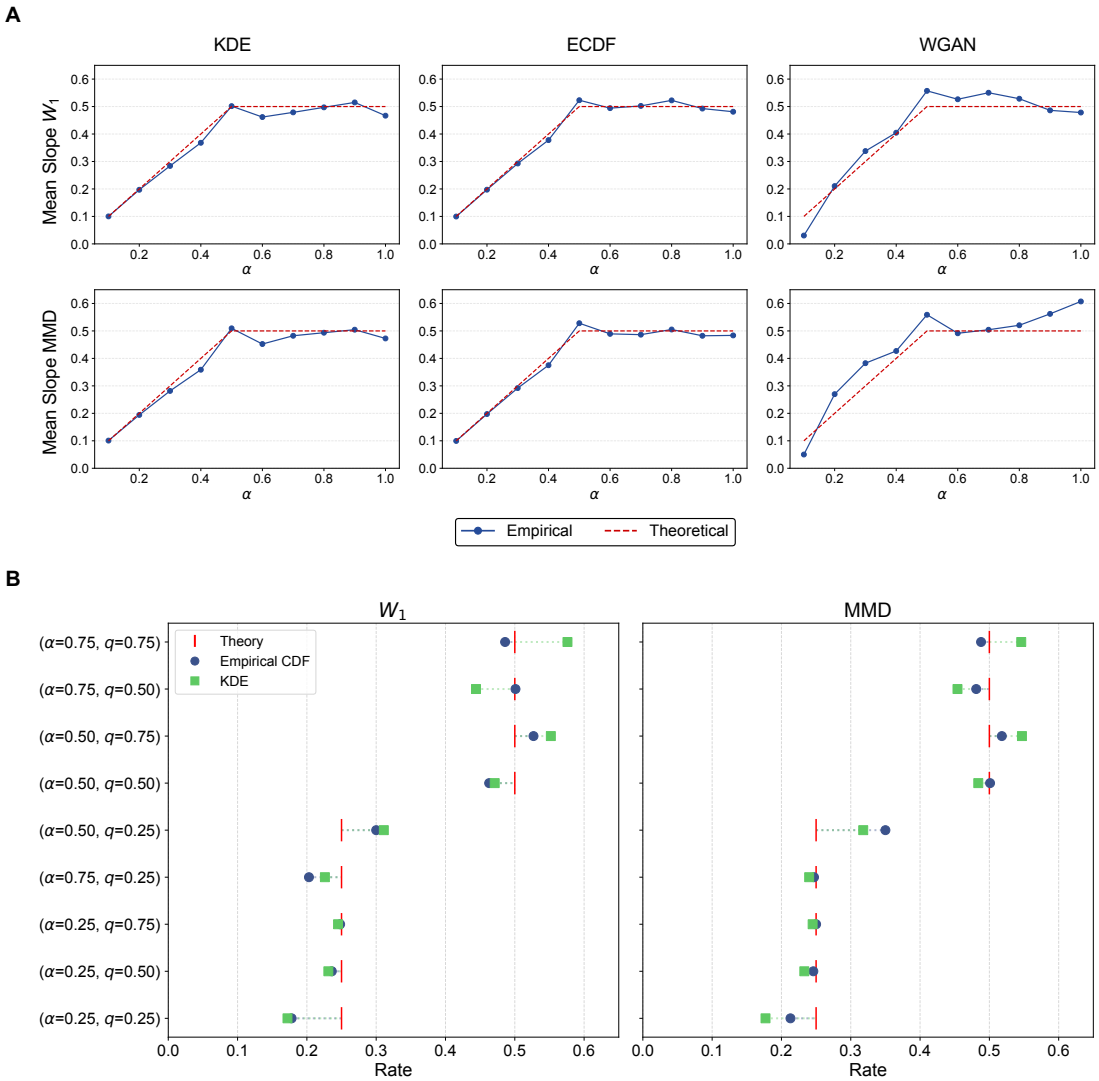


Figure 3: Simulation results. (A) Rates of KDE (left), ECDF (middle), and WGAN (right) measured using W_1 (top) and MMD (bottom) under CRT with varying α, with empirical rates in blue and theoretical rates in red, where the target density is a 1-dimensional mixture of two Gaussians. (B) Rates for ECDF and KDE under BCRT with varying α and q measured by W_1 and MMD. For both CRT and BCRT, the empirical rates match the theoretical rates.

numerous architectural and training hyperparameters, and the lack of a computationally tractable exact W_1 metric in high-dimensional spaces. Moreover, unlike the synthetic experiments, the true data distribution $\mathbb{P}_0$ is not explicitly known.

Consequently, we evaluate model quality using the Fréchet Inception Distance (FID), a

standard metric for assessing the similarity between generated and real images. While FID is not itself a convergence metric, sustained improvements in FID indicate that generated samples are becoming progressively closer to the real-data distribution. The resulting FID trajectories are shown in Figure 4B, while representative generated samples are displayed in Figure 4C.

Hence, our objective in this experiment is not to verify exact convergence rates, but rather to investigate whether the qualitative convergence behavior predicted by Theorem 3.4 persists in a realistic high-dimensional setting. In particular, we examine whether Diffusion Models trained under CRT continue to improve even when the real-data fraction at each iteration is substantially less than 0.5, corresponding to a setting with significant recursive contamination.

At each iteration, a training set of 300 images is constructed by sampling a fraction α of images from the MNIST dataset without replacement and a fraction $1-\alpha$ of synthetic images generated by the model trained in the previous iteration. The diffusion model is then trained for 150 epochs on the resulting dataset. This procedure is repeated for 300 iterations, with the model parameters carried forward between iterations rather than being reinitialized. Consequently, data accumulates through the optimizer, and each real training sample is used an equal number of times throughout the recursive training process, consistent with the assumptions of Theorem 3.4. Due to the high dimensionality of the image space, W_1 distances are not evaluated. Additional experimental details may be found in Section E.

Overall, the results demonstrate that even when $\alpha < 1$, the diffusion model trained under CRT continues to improve according to FID and produces increasingly realistic samples. These findings provide empirical evidence that the convergence behavior predicted by Theorem 3.4 extends beyond the idealized settings considered earlier and remains observable in practical image-generation tasks.

7 Large Language Model Experiments

We next investigate whether the qualitative behavior predicted by CRT persists in Large Language Models. Starting from a pretrained GPT-125M model, we perform recursive fine-tuning on WikiText-103 following the CRT procedure.

At iteration t, a total of $m_{\text{total}} = 1024$ token blocks of length 1024 are selected. A fraction $\alpha \in \{0, 0.1, 1\}$ is drawn from the real dataset without replacement, while the remaining fraction $1-\alpha$ consists of synthetic blocks generated by the model obtained from the previous iteration. The model is then fine-tuned for exactly one epoch on the resulting dataset. This procedure is repeated for $T = 50$ iterations. As in the image experiments of Section 6, the model parameters are carried forward between iterations rather than reinitialized, so data accumulates through the optimizer. Consequently, each real data block is utilized an equal number of times throughout training, consistent with the assumptions of Theorem 3.4. The results are summarized in Figure 5.

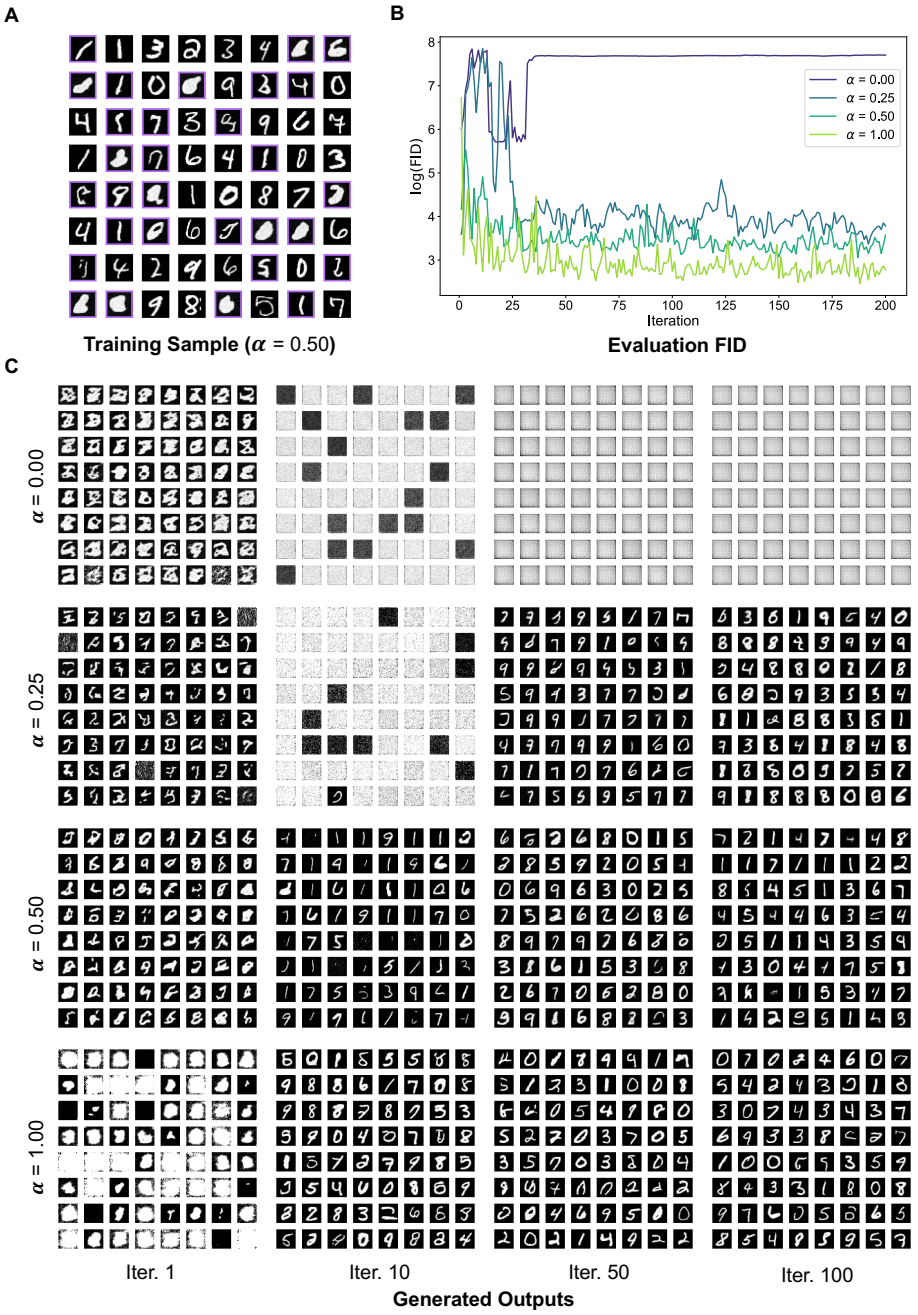


Figure 4: Image experiments under CRT using Diffusion Models. (A) Sample input data with $\alpha = 0.5$, where synthetically generated samples are highlighted. (B) FID at each training iteration for different values of α. FID decreases steadily across iterations only when $\alpha > 0$. (C) Samples generated by successive models in the CRT sequence for different values of α. When $\alpha > 0$, image quality improves across iterations, whereas training with $\alpha = 0$ leads to collapse, consistent with theoretical results.

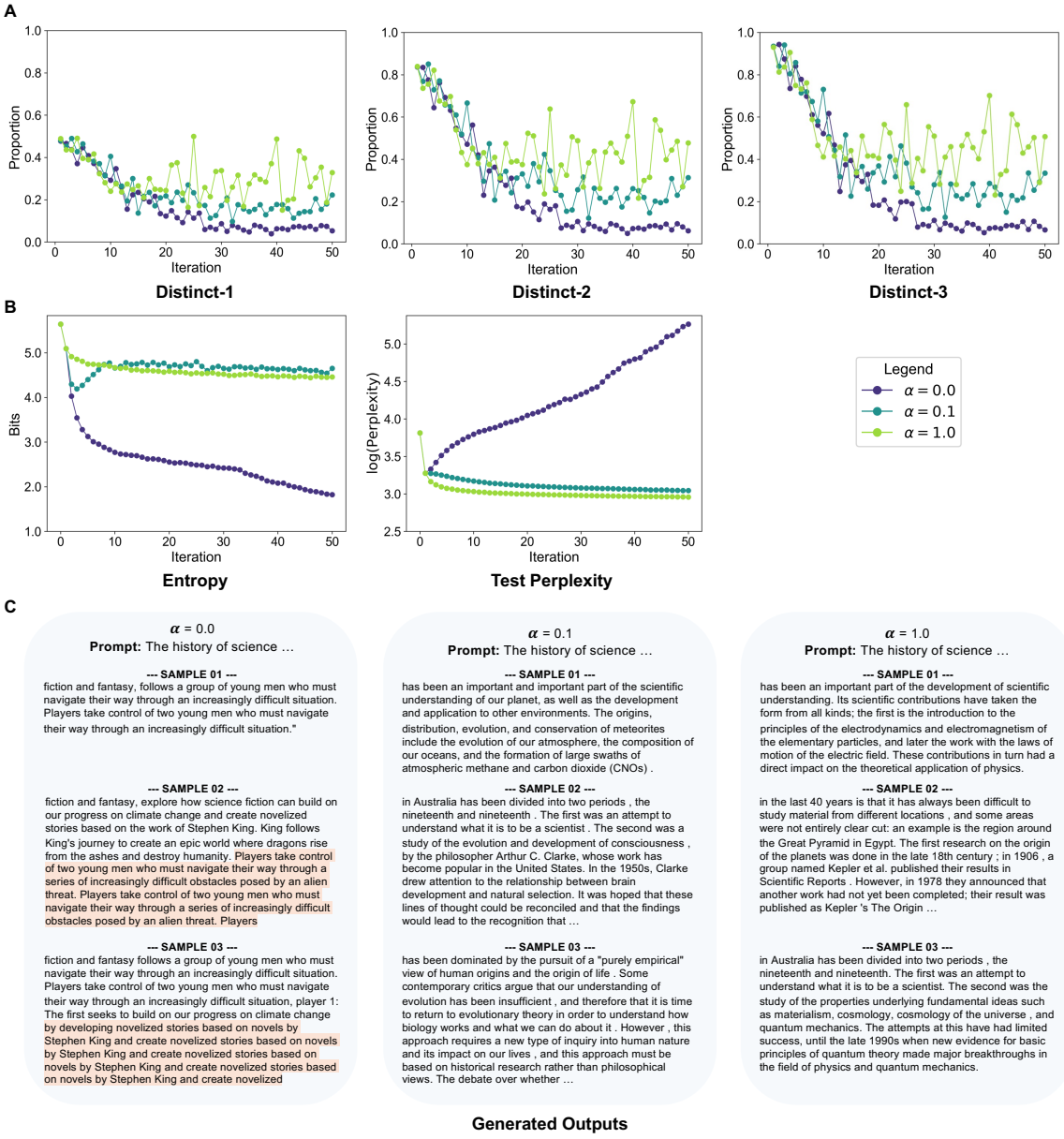


Figure 5: Text experiments under CRT using LLMs. (A) Output diversity of the fine-tuned LLMs over iterations with varying α's, measured using number of distinct k-grams to assess potential model collapse for $k = 1, 2, 3$. Diversity collapses rapidly when $\alpha = 0$, but is maintained when $\alpha = 0.1, 1$ cases. (B) Output entropy (left) and log prediction perplexity (middle) on a held-out WikiText test set measured at each iteration, with varying α's. Even with $\alpha = 0.1$, the test perplexity continues to decrease. (C) Variety of generated outputs of different generations of the model with different α, displaying model collapsed outputs for $\alpha = 0$ (highlighted in orange) and stabilized outputs for $\alpha > 0$.

To assess potential model collapse more directly, Figure 5 reports several complementary metrics. Figure 5A shows output diversity measured by the proportion of distinct k-grams. Figure 5B reports output entropy and test perplexity on a held-out WikiText test set. Because the underlying data distribution is unavailable, direct distributional distances cannot be computed. We therefore evaluate model quality using perplexity on a held-out WikiText test set. Although perplexity is only an indirect measure of distributional alignment, decreasing test perplexity indicates improved predictive performance and closer agreement with the test distribution. When $\alpha = 0$, diversity collapses nearly to zero, output entropy decreases substantially, and test perplexity steadily worsens, consistent with model collapse under purely synthetic recursion. In contrast, for both $\alpha = 0.1$ and $\alpha = 1$, diversity remains substantially preserved, entropy remains stable at levels comparable to the fully real-data setting, and test perplexity continues to improve across iterations. Together, these results suggest that even a modest influx of real data is sufficient to prevent the degenerative behavior observed in the synthetic-only setting and maintain stable recursive training.

We also present the generated outputs in Figure 5C after 50 iterations of CRT. When $\alpha = 0$, the outputs exhibit severe degeneration, both in terms of textual quality and relevance to the prompt. In contrast, when $\alpha = 1$, the generated outputs remain coherent and closely aligned with the prompt. Notably, even when $\alpha = 0.1$, incorporating only 10% real data at each iteration is sufficient to prevent collapse and maintain reasonable output quality.

These results indicate that recursive training can remain stable and continue improving at scale. They suggest that the dynamics predicted by the CRT framework may extend to large-scale language models operating in increasingly synthetic data environments.

8 Discussion and Future Work

In this article, we studied the long-term behavior of generative AI recursively trained on mixtures of real and synthetic data. Both our theoretical and empirical analysis shows that recursive training can survive data contamination as long as real data is included in each training iteration. Moreover, the asymptotic behavior is governed by three factors: the intrinsic rate of the AI model, real-data fraction in each iteration, and the rate at which bias in the data is corrected, with phase transitions, leading to model-limited, data-limited, and bias-limited regimes.

Our results suggest that the spread of AI-generated content across the internet does not necessarily cause AI models to deteriorate over time. What matters most is whether models continue to learn from new information about the real-world. In practice, this means that human-created material, such as journalism, photography, scientific articles, blog posts, and everyday online discussion, are central for shaping future AI models. As long as this stream of original content remains sufficient in both quality and quantity, AI models can continue to improve even in an environment where synthetic content is common.

The results also provide perspective on concerns about bias in AI models. When real datasets contain systematic distortions, for example, when certain groups, perspectives, or experiences are under or over represented, models trained on those data can reproduce

those patterns in their outputs. Under recursive training, it is feared that such distortions will reinforce themselves across successive generations of models, both perpetuating those biases onto internet users as well as solidifying their own presence in the next generation of AI. However, our analysis also shows that this process is not irreversible. Improvements in data collection and bias-correction can reduce these distortions over time, provided that the quality of incoming data improves sufficiently quickly. In practical terms, this suggests that efforts to broaden dataset representation, such as including more diverse sources, viewpoints, and communities, can still meaningfully improve future AI models even if earlier models were trained on imperfect data.

While our work establishes the foundations for recursively trained generative AI, several emerging aspects warrant future investigation. First, many important objectives used in modern generative modeling, including likelihood, cross-entropy, and Kullback-Leibler based methods, do not directly satisfy the assumptions of our theory. Future work may proceed in two complementary directions: extending the CRT framework to accommodate broader classes of discrepancies, or establishing conditions under which training procedures based on these objectives yield convergence in a discrepancy compatible with the current theory. Whether such connections exist, and under what assumptions, remains an open question.

Second, the real-world path from model generation to future training data is considerably more complex than the CRT framework. Model outputs are not transferred directly into the internet, but instead pass through multiple stages of selection, editing, publication, and amplification by human users and automated systems. Modern training pipelines further apply procedures such as dataset curation, annotation, supervised fine tuning, reinforcement learning from human feedback, and learned reward models. Consequently, future models are shaped not only by the content that enters the training corpus but also by the mechanisms used to select and learn from that content. Extending recursive-training theory to account for these processes remains an important direction for future work, as they may either amplify or mitigate the effects of recursive contamination.

Third, the distinction between real and synthetic data is becoming increasingly blurred. Many forms of online content, including research articles, software, blog posts, and digital media, are now produced through iterative interactions between humans and AI systems. In practice, content is often drafted, edited, refined, or curated through multiple rounds of human-AI collaboration before it is ultimately published. Such data cannot be naturally categorized as either purely human-generated or purely synthetic. As these forms of hybrid intelligence become increasingly prevalent, it will be important to develop recursive-training frameworks that explicitly account for data generated through human-AI interaction rather than assuming a clean separation between real and synthetic sources.

Finally, we note that a substantial literature leverages the intentional use of synthetic data for training. In data-scarce settings, training on synthetic data has enabled models to achieve strong performance on real data, both after subsequent retraining and, in some cases, even without further adaptation. In some settings, models are trained entirely on data generated from computer simulation and then deployed in the physical world with little

or no additional real-world training, such as in reinforcement learning with low tolerance for negative outcomes, such as controlling vehicles Sadeghi and Levine (2016). An important direction for future work is to extend our analysis to settings where synthetic data is deliberately introduced due to limited real data availability, and where one can control where, when, and how much synthetic data is incorporated.

References

Eric Arazo, Diego Ortego, Paul Albert, Noel E O'Connor, and Kevin McGuinness. Pseudo-labeling and confirmation bias in deep semi-supervised learning. In *2020 International joint conference on neural networks (IJCNN)*, pages 1–8. IEEE, 2020.

Daniel Barzilai and Ohad Shamir. When models don't collapse: On the consistency of iterative mle. *Advances in Neural Information Processing Systems*, 38:76813–76854, 2026.

BBC News. Bbc reveals web of spammers profiting from ai holocaust images, August 2025. URL `https://www.bbc.com/news/articles/ckg4xjk1g1xo`. Retrieved November 3, 2025.

Quentin Bertrand, Avishek Joey Bose, Alexandre Duplessis, Marco Jiralerspong, and Gauthier Gidel. On the stability of iterative retraining of generative models on their own data. *arXiv preprint arXiv:2310.00429*, 2023.

Benjamin Billot, Douglas N Greve, Oula Puonti, Axel Thielscher, Koen Van Leemput, Bruce Fischl, Adrian V Dalca, Juan Eugenio Iglesias, et al. Synthseg: Segmentation of brain mri scans of any contrast and resolution without retraining. *Medical image analysis*, 86:102789, 2023.

Christopher Bowles, Liang Chen, Ricardo Guerrero, Paul Bentley, Roger Gunn, Alexander Hammers, David Alexander Dickie, Maria Valdés Hernández, Joanna Wardlaw, and Daniel Rueckert. Gan augmentation: Augmenting training data using generative adversarial networks. *arXiv preprint arXiv:1810.10863*, 2018.

Martin Briesch, Dominik Sobania, and Franz Rothlauf. Large language models suffer from their own output: An analysis of the self-consuming training loop. 2023.

Creston Brooks, Samuel Eggert, and Denis Peskoff. The rise of ai-generated content in wikipedia. *arXiv preprint arXiv:2410.08044*, 2024.

Minshuo Chen, Kaixuan Huang, Tuo Zhao, and Mengdi Wang. Score approximation, estimation and distribution recovery of diffusion models on low-dimensional data. In *International Conference on Machine Learning*, pages 4672–4712. PMLR, 2023a.

Zhenpeng Chen, Jie M Zhang, Federica Sarro, and Mark Harman. A comprehensive empirical study of bias mitigation methods for machine learning classifiers. *ACM transactions on software engineering and methodology*, 32(4):1–30, 2023b.

Badr-Eddine Chérief-Abdellatif, Yuyang Shi, Arnaud Doucet, and Benjamin Guedj. On pac-bayesian reconstruction guarantees for vaes. In *International conference on artificial intelligence and statistics*, pages 3066–3079. PMLR, 2022.

Corinna Cortes and Mehryar Mohri. Domain adaptation and sample bias correction theory and algorithm for regression. *Theoretical Computer Science*, 519:103–126, 2014.

Valentin De Bortoli. Convergence of denoising diffusion models under the manifold hypothesis. *arXiv preprint arXiv:2208.05314*, 2022.

Luc Devroye. Nonparametric density estimation. *The L_1 View*, 1985.

Apratim Dey and David Donoho. Universality of the $\pi^2/6$ pathway in avoiding model collapse. *arXiv preprint arXiv:2410.22812*, 2024.

Elvis Dohmatob, Yunzhen Feng, and Julia Kempe. Model collapse demystified: The case of regression. *Advances in Neural Information Processing Systems*, 37:46979–47013, 2024a.

Elvis Dohmatob, Yunzhen Feng, Pu Yang, Francois Charton, and Julia Kempe. A tale of tails: Model collapse as a change of scaling laws. *arXiv preprint arXiv:2402.07043*, 2024b.

Elvis Dohmatob, Yunzhen Feng, Arjun Subramonian, and Julia Kempe. Strong model collapse. In *International Conference on Learning Representations*, volume 2025, pages 15656–15691, 2025.

Yunzhen Feng, Elvis Dohmatob, Pu Yang, Francois Charton, and Julia Kempe. Beyond model collapse: Scaling up with synthesized data requires verification. In *International Conference on Learning Representations*, volume 2025, pages 89702–89730, 2025.

Rémi Flamary, Nicolas Courty, Alexandre Gramfort, Mokhtar Z Alaya, Aurélie Boisbunon, Stanislas Chambon, Laetitia Chapel, Adrien Corenflos, Kilian Fatras, Nemo Fournier, et al. Pot: Python optimal transport. *The Journal of Machine Learning Research*, 22(1): 3571–3578, 2021.

Nicolas Fournier and Arnaud Guillin. On the rate of convergence in wasserstein distance of the empirical measure. *Probability theory and related fields*, 162(3):707–738, 2015.

Nicole Froio. Ai is ruining houseplant communities online, June 2025. URL `https://www.theverge.com/ai-artificial-intelligence/691355/ai-is-ruining-houseplant-communities-online`. Accessed: 2025-06-26.

Walter Gautschi. Some elementary inequalities relating to the gamma and incomplete gamma function. *J. Math. Phys*, 38(1):77–81, 1959.

Matthias Gerstgrasser, Rylan Schaeffer, Apratim Dey, Rafael Rafailov, Henry Sleight, John Hughes, Tomasz Korbak, Rajashree Agrawal, Dhruv Pai, Andrey Gromov, et al. Is model collapse inevitable? breaking the curse of recursion by accumulating real and synthetic data. *arXiv preprint arXiv:2404.01413*, 2024.

Subhashis Ghosal and Aad W Van der Vaart. *Fundamentals of nonparametric Bayesian inference*, volume 44. Cambridge University Press, 2017.

Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *The journal of machine learning research*, 13(1):723–773, 2012.

Ryuichiro Hataya, Han Bao, and Hiromi Arai. Will large-scale generative models corrupt future datasets? In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 20555–20565, 2023.

Haibo He and Edwardo A Garcia. Learning from imbalanced data. *IEEE Transactions on knowledge and data engineering*, 21(9):1263–1284, 2009.

James Jordon, Jinsung Yoon, and Mihaela Van Der Schaar. Pate-gan: Generating synthetic data with differential privacy guarantees. In *International conference on learning representations*, 2018.

Dong-Hyun Lee et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks.

Weixin Liang, Zachary Izzo, Yaohui Zhang, Haley Lepp, Hancheng Cao, Xuandong Zhao, Lingjiao Chen, Haotian Ye, Sheng Liu, Zhi Huang, et al. Monitoring ai-modified content at scale: A case study on the impact of chatgpt on ai conference peer reviews. *PMLR*, 2024.

Raphael Gontijo Lopes, Stefano Fenu, and Thad Starner. Data-free knowledge distillation for deep neural networks. *arXiv preprint arXiv:1710.07535*, 2017.

Sanae Lotfi, Yilun Kuang, Marc Finzi, Brandon Amos, Micah Goldblum, and Andrew G Wilson. Unlocking tokens as data points for generalization bounds on larger language models. *Advances in Neural Information Processing Systems*, 37:9229–9256, 2024.

Li Lucy and David Bamman. Gender and representation bias in gpt-3 generated stories. In *Proceedings of the third workshop on narrative understanding*, pages 48–55, 2021.

Gonzalo Martínez, Lauren Watson, Pedro Reviriego, José Alberto Hernández, Marc Juarez, and Rik Sarkar. Combining generative artificial intelligence (ai) and the internet: Heading towards evolution or degradation? *arXiv preprint arXiv:2303.01255*, 2023.

Sokhna Diarra Mbacke, Florence Clerc, and Pascal Germain. Statistical guarantees for variational autoencoders using pac-bayesian theory. *Advances in Neural Information Processing Systems*, 36:56903–56915, 2023.

Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. A survey on bias and fairness in machine learning. *ACM computing surveys (CSUR)*, 54 (6):1–35, 2021.

Gaithersburg Md Nist. Reducing risks posed by synthetic content: An overview of technical approaches to digital content transparency. *NIST AI*, pages 100–41, 2024.

Kazusato Oko, Shunta Akiyama, and Taiji Suzuki. Diffusion models are minimax optimal distribution estimators. In *International Conference on Machine Learning*, pages 26517–26582. PMLR, 2023.

Nicolas Papernot, Martín Abadi, Ulfar Erlingsson, Ian Goodfellow, and Kunal Talwar. Semi-supervised knowledge transfer for deep learning from private training data. *arXiv preprint arXiv:1610.05755*, 2016.

Nikita Puchkin, Sergey Samsonov, Denis Belomestny, Eric Moulines, and Alexey Naumov. Rates of convergence for density estimation with generative adversarial networks. *Journal of Machine Learning Research*, 25(29):1–47, 2024.

Ankit Singh Rawat, Veeranjaneyulu Sadhanala, Afshin Rostamizadeh, Ayan Chakrabarti, Wittawat Jitkrittum, Vladimir Feinberg, Seungyeon Kim, Hrayr Harutyunyan, Nikunj Saunshi, Zachary Nado, et al. A little help goes a long way: Efficient llm training by leveraging small lms. *arXiv preprint arXiv:2410.18779*, 2024.

Walter Rudin. Principles of mathematical analysis. *3rd ed.*, 1976.

Fereshteh Sadeghi and Sergey Levine. Cad2rl: Real single-image flight without a single real image. *arXiv preprint arXiv:1611.04201*, 2016.

Emily Sheng, Kai-Wei Chang, Prem Natarajan, and Nanyun Peng. The woman worked as a babysitter: On biases in language generation. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pages 3407–3412, 2019.

Ilia Shumailov, Zakhar Shumaylov, Yiren Zhao, Yarin Gal, Nicolas Papernot, and Ross Anderson. The curse of recursion: Training on generated data makes models forget. *arXiv preprint arXiv:2305.17493*, 2023.

Ilia Shumailov, Zakhar Shumaylov, Yiren Zhao, Nicolas Papernot, Ross Anderson, and Yarin Gal. Ai models collapse when trained on recursively generated data. *Nature*, 631 (8022):755–759, 2024.

Ananda Theertha Suresh, Andrew Thangaraj, and Aditya Nanda Kishore Khandavally. Rate of model collapse in recursive training. *arXiv preprint arXiv:2412.17646*, 2024.

Rong Tang, Lizhen Lin, and Yun Yang. Conditional diffusion models are minimax-optimal and manifold-adaptive for conditional distribution estimation. *arXiv preprint arXiv:2409.20124*, 2024.

Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 23–30. IEEE, 2017.

Ananya Uppal, Shashank Singh, and Barnabás Póczos. Nonparametric density estimation & convergence rates for gans under besov ipm losses. *Advances in neural information processing systems*, 32, 2019.

Liyang Xie, Kaixiang Lin, Shu Wang, Fei Wang, and Jiayu Zhou. Differentially private generative adversarial network. *arXiv preprint arXiv:1802.06739*, 2018.

Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10687–10698, 2020.

Hongxu Yin, Pavlo Molchanov, Jose M Alvarez, Zhizhong Li, Arun Mallya, Derek Hoiem, Niraj K Jha, and Jan Kautz. Dreaming to distill: Data-free knowledge transfer via deepinversion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8715–8724, 2020.

Jinghui Zhang, Mochen Yang, Dandan Qiao, and Qiang Wei. Regurgitative training: The value of real data in training large language models. *Management Science*, 2026.

Mi Zhou, Vibhanshu Abhishek, Timothy Derdenger, Jaymo Kim, and Kannan Srinivasan. Bias in generative ai. *arXiv preprint arXiv:2403.02726*, 2024.

Appendix

This appendix contains the supplementary material for the paper "Can Generative Artificial Intelligence Survive Data Contamination? Theoretical Guarantees under Contaminated Recursive Training." In Section A we provide a link to all code necessary to produce the simulations and experiments of the paper. In Section B, we provide a more detailed and technical discussion of the Section 3, including the assumption, theorem, and proof. In Section C, we provide the corresponding technical details for Section 4. In Section D, we provide theoretical justification for the baseline rates assumed in our Section 5, as well as additional details for the simulations. In Section E, we provide additional experimental details for the image experiments in Section 6. Finally in Section F, we provide additional experimental details for the LLM experiments in Section 7.

A Code Availability

All code to run simulations, experiments, and to generate all figures are available at: `https://github.com/hong-niu/generative-recursive-training`.

B CRT: Theoretical Details and Proof

This section provides the formal proof of **Theorem 3.4 (Convergence under CRT)**. We follow the notation introduced in the main text.

Proof. Define

$$\mathbb{Q}_t = \frac{1}{M_t}\left[tm_1\mathbb{P}_0 + \sum_{j=1}^{t-1} m_2\widehat{\mathbb{P}}_j\right], \qquad M_t = tm_1 + (t-1)m_2.$$

Using Assumptions A1 and A2, we first use the triangle inequality to write

$$d(\widehat{\mathbb{P}}_t, \mathbb{P}_0) \le d(\widehat{\mathbb{P}}_t, \mathbb{Q}_t) + d(\mathbb{Q}_t, \mathbb{P}_0).$$

To treat $d(\widehat{\mathbb{P}}_t, \mathbb{Q}_t)$, note that $\widehat{\mathbb{P}}_t$ is learned from the accumulated data whose distribution is $\mathbb{Q}_t$. Using Assumption A1, which states that the baseline generative model has a uniform polynomial rate p, we treat $\mathbb{Q}_t$ as the distribution to be learned by $\widehat{\mathbb{P}}_t$ at iteration t, yielding $d(\widehat{\mathbb{P}}_t, \mathbb{Q}_t) \lesssim M_t^{-p}$. We write

$$d(\widehat{\mathbb{P}}_t, \mathbb{P}_0) \le d(\widehat{\mathbb{P}}_t, \mathbb{Q}_t) + d(\mathbb{Q}_t, \mathbb{P}_0) \lesssim M_t^{-p} + d(\mathbb{Q}_t, \mathbb{P}_0).$$

From Assumption A2, by convexity of d in its second argument,

$$\begin{aligned} d(\mathbb{Q}_t, \mathbb{P}_0) &= d\left(\mathbb{P}_0, \frac{tm_1}{M_t}\mathbb{P}_0 + \frac{m_2}{M_t}\sum_{j=1}^{t-1}\widehat{\mathbb{P}}_j\right) \\ &\le \frac{tm_1}{M_t} d(\mathbb{P}_0, \mathbb{P}_0) + \frac{m_2}{M_t}\sum_{j=1}^{t-1} d(\mathbb{P}_0, \widehat{\mathbb{P}}_j) \\ &= \frac{m_2}{M_t}\sum_{j=1}^{t-1} d(\mathbb{P}_0, \widehat{\mathbb{P}}_j). \end{aligned}$$

Thus

$$d(\widehat{\mathbb{P}}_t, \mathbb{P}_0) \lesssim M_t^{-p} + \frac{m_2}{M_t}\sum_{j=1}^{t-1} d(\mathbb{P}_0, \widehat{\mathbb{P}}_j). \tag{1}$$

Let $S_{t-1} := \sum_{j=1}^{t-1} d(\mathbb{P}_0, \widehat{\mathbb{P}}_j)$. Applying (1) at step $t-1$ gives

$$d(\widehat{\mathbb{P}}_{t-1}, \mathbb{P}_0) \lesssim M_{t-1}^{-p} + \frac{m_2}{M_{t-1}} S_{t-2},$$

so

$$\begin{aligned} S_{t-1} &= d(\widehat{\mathbb{P}}_{t-1}, \mathbb{P}_0) + S_{t-2} \\ &\lesssim M_{t-1}^{-p} + \left(1 + \frac{m_2}{M_{t-1}}\right) S_{t-2}. \end{aligned}$$

Iterating this recursion yields

$$S_{t-1} \lesssim M_{t-1}^{-p} + \sum_{j=1}^{t-2} M_j^{-p} \prod_{k=j}^{t-2} \left(1 + \frac{m_2}{M_k}\right).$$

Substituting back into (1) gives

$$d(\widehat{\mathbb{P}}_t, \mathbb{P}_0) \lesssim M_t^{-p} + \frac{m_2}{M_t} M_{t-1}^{-p} + \frac{m_2}{M_t} \sum_{j=1}^{t-2} M_j^{-p} \prod_{k=j}^{t-2} \left(1 + \frac{m_2}{M_k}\right).$$

Since

$$M_t = tm_1 + (t-1)m_2 = (m_1 + m_2)t - m_2,$$

we have $M_t \asymp (m_1 + m_2)t$, hence there exist constants $c_1, c_2 > 0$ such that

$$c_1(m_1 + m_2)t \le M_t \le c_2(m_1 + m_2)t \quad \text{for all } t \ge 1.$$

Similarly,

$$M_t^{-p} \lesssim (m_1 + m_2)^{-p} t^{-p}, \qquad \frac{m_2}{M_t} \lesssim \frac{1-\alpha}{t},$$

with $\alpha = m_1/(m_1 + m_2)$, and similarly $M_j^{-p} \lesssim (m_1 + m_2)^{-p} j^{-p}$. Thus

$$\begin{aligned} d(\widehat{\mathbb{P}}_t, \mathbb{P}_0) &\lesssim (m_1 + m_2)^{-p} t^{-p} + m_2 (m_1 + m_2)^{-p-1} (t-1)^{-p-1} \\ &\quad + (1-\alpha) t^{-1} (m_1 + m_2)^{-p} \sum_{j=1}^{t-2} j^{-p} \prod_{k=j}^{t-2} \left(1 + \frac{1-\alpha}{k}\right) \qquad (2) \\ &\lesssim t^{-p} + t^{-p-1} + t^{-1} \sum_{j=1}^{t-2} j^{-p} \prod_{k=j}^{t-2} \left(1 + \frac{1-\alpha}{k}\right). \qquad (3) \end{aligned}$$

Up to multiplicative constants, we analyze the summation in Equation (3):

$$\begin{aligned} t^{-1} \sum_{j=1}^{t-2} j^{-p} \prod_{k=j}^{t-2} \left(1 + \frac{1-\alpha}{k}\right) &= \sum_{j=1}^{t-2} j^{-p} t^{-1} \left[\prod_{k=j}^{t-2} \left(\frac{k+1-\alpha}{k}\right)\right] \\ &= \sum_{j=1}^{t-2} j^{-p} t^{-1} \frac{t-1}{j} \left[\prod_{k=j}^{t-2} \left(\frac{k+1-\alpha}{k}\right)\left(\frac{k}{k+1}\right)\right] \\ &= \sum_{j=1}^{t-2} j^{-p} \left(\frac{t-1}{t}\right) j^{-1} \left[\prod_{k=j}^{t-2} \left(\frac{k+1-\alpha}{k+1}\right)\right], \end{aligned}$$

where we use the identity $\prod_{k=j}^{t-2} \frac{k}{k+1} = \frac{j}{t-1}$. Using the Gamma function, we have the exact identity

$$\prod_{k=j}^{t-2} \left(\frac{k+1-\alpha}{k+1} \right) = \frac{\Gamma(t+1-\alpha-1)}{\Gamma(j+1-\alpha)} \frac{\Gamma(j+1)}{\Gamma(t)} = \frac{\Gamma(t-\alpha)}{\Gamma(j+1-\alpha)} \frac{\Gamma(j+1)}{\Gamma(t)}.$$

We recall the following inequalities for the Gamma function. For $x > 0$ and $0 < s < 1$, Gautschi's inequality (Gautschi, 1959) states that

$$x^{1-s} < \frac{\Gamma(x+1)}{\Gamma(x+s)} < (x+1)^{1-s}.$$

Combined with standard Stirling-type bounds for Γ (Rudin, 1976), this implies that for any $\alpha \in (0,1)$ there exist constants $c_1, c_2, c_3, c_4 > 0$ such that for all integers $t, j \geq 1$,

$$c_1 t^{-\alpha} \leq \frac{\Gamma(t-\alpha)}{\Gamma(t)} \leq c_2 t^{-\alpha}, \qquad c_3 j^{\alpha} \leq \frac{\Gamma(j+1)}{\Gamma(j+1-\alpha)} \leq c_4 j^{\alpha}. \tag{4}$$

Using (4), the contribution of the last term (without constants) in Equation (3) is bounded by

$$\left(\frac{t-1}{t} \right) \frac{\Gamma(t-\alpha)}{\Gamma(t)} \sum_{j=1}^{t-2} j^{-p-1} \frac{\Gamma(j+1)}{\Gamma(j+1-\alpha)} \lesssim t^{-\alpha} \sum_{j=1}^{t-2} j^{-p-1+\alpha}.$$

Putting all terms together, we have shown that

$$d(\widehat{\mathbb{P}}_t, \mathbb{P}_0) \lesssim t^{-p} + (t-1)^{-p-1} + t^{-\alpha} \sum_{j=1}^{t-2} j^{-p-1+\alpha}. \tag{5}$$

Standard bounds for p-series yield

$$\sum_{j=1}^{t-2} j^{-1-(p-\alpha)} \lesssim \begin{cases} 1, & p > \alpha, \\ \log t, & p = \alpha, \\ t^{\alpha-p}, & p < \alpha. \end{cases}$$

We treat the three regimes separately, absorbing the first two terms of (5) into the dominant term by adjusting the constant if needed.

Case 1: $p > \alpha$**.** Then $\sum_{j=1}^{t-2} j^{-p-(1-\alpha)} \lesssim 1$, so the third term in (5) is

$$\lesssim t^{-\alpha}.$$

Moreover, since $p > \alpha$ we have $t^{-p} \leq t^{-\alpha}$ for all $t \geq 1$, and

$$(t-1)^{-p-1} \lesssim t^{-p-1} \leq t^{-p} \leq t^{-\alpha}.$$

Hence, all three terms in (5) are asymptotically bounded by $t^{-\alpha}$, and therefore

$$d(\widehat{\mathbb{P}}_t, \mathbb{P}_0) \lesssim t^{-\alpha}.$$

Case 2: $p = \alpha$**.** Then $\sum_{j=1}^{t-2} j^{-p-(1-\alpha)} = \sum_{j=1}^{t-2} j^{-1} \lesssim \log t$, and $t^{-p} = t^{-\alpha}$. Thus the third term in (5) is

$$\lesssim t^{-\alpha} \log t.$$

The first term in (5) is t^{-p}, which is dominated by $t^{-p} \log t$, and the second term $(t-1)^{-p-1}$ is of strictly smaller order. Therefore

$$d(\widehat{\mathbb{P}}_t, \mathbb{P}_0) \lesssim t^{-\alpha} \log t.$$

Case 3: $p < \alpha$**.** Then

$$\sum_{j=1}^{t-2} j^{-1-(p-\alpha)} \lesssim t^{1-1-(p-\alpha)} = t^{\alpha-p}.$$

Hence the third term in (5) is

$$t^{-\alpha} t^{\alpha-p} = t^{-p}.$$

The first term t^{-p} and the second term $(t-1)^{-p-1}$ are dominated by the third term, so all three terms are $O(t^{-p})$ up to multiplicative constants. Therefore

$$d(\widehat{\mathbb{P}}_t, \mathbb{P}_0) \lesssim t^{-p}.$$

Combining the three cases, we have proved that

$$d(\widehat{\mathbb{P}}_t, \mathbb{P}_0) \lesssim \begin{cases} t^{-\alpha}, & p > \alpha, \\ t^{-\alpha} \log t, & p = \alpha, \\ t^{-p}, & p < \alpha. \end{cases}$$

If Assumption A1 holds in probability, then all asymptotic inequalities above also hold in probability, which completes the proof.

□

C BCRT: Theoretical Details and Proof

This section provides a formal proof of **Theorem 4.3 (Convergence under BCRT)**. We again follow the notation and framework introduced in the main text. The following elementary lemma will be useful in the proof of Theorem 4.3.

Lemma (Cesàro rate for drifting distributions). Let $d(\cdot,\cdot)$ be a metric on distributions that satisfies Assumption A2. Let $\{\mathbb{P}_t\}_{t\geq 1}$ be a sequence of distributions and let $\mathbb{P}_0$ be a target distribution such that for some $q > 0$,

$$d(\mathbb{P}_t, \mathbb{P}_0) \lesssim t^{-q}.$$

Define the Cesàro averages

$$\overline{\mathbb{P}}_t := \frac{1}{t}\sum_{j=1}^{t} \mathbb{P}_j.$$

Then

$$d(\overline{\mathbb{P}}_t, \mathbb{P}_0) \lesssim t^{-\min\{q,1\}},$$

up to a $\log t$ factor in the boundary case $q = 1$.

Proof. By convexity of d in its second argument,

$$d(\overline{\mathbb{P}}_t, \mathbb{P}_0) = d\left(\mathbb{P}_0, \frac{1}{t}\sum_{j=1}^{t}\mathbb{P}_j\right) \leq \frac{1}{t}\sum_{j=1}^{t} d(\mathbb{P}_0, \mathbb{P}_j) \lesssim \frac{1}{t}\sum_{j=1}^{t} j^{-q}.$$

The standard p-series asymptotics yield

$$\frac{1}{t}\sum_{j=1}^{t} j^{-q} \asymp \begin{cases} t^{-q}, & 0 < q < 1, \\ t^{-1}\log t, & q = 1, \\ t^{-1}, & q > 1, \end{cases}$$

which is $t^{-\min\{q,1\}}$ up to a logarithmic factor for $q = 1$. □

We now prove Theorem 4.3.

Proof of Theorem 4.3. At step t, define the real data average

$$\overline{\mathbb{P}}_t^{\text{bias}} := \frac{1}{t}\sum_{j=1}^{t} \mathbb{P}_j^{\text{bias}}.$$

Then

$$\mathbb{Q}_t = \frac{tm_1}{M_t}\,\overline{\mathbb{P}}_t^{\text{bias}} + \frac{m_2}{M_t}\sum_{j=1}^{t-1}\widehat{\mathbb{P}}_j.$$

Using the triangle inequality,

$$d(\widehat{\mathbb{P}}_t, \mathbb{P}_0) \leq d(\widehat{\mathbb{P}}_t, \mathbb{Q}_t) + d(\mathbb{Q}_t, \mathbb{P}_0).$$

Step 1: Learning error and bias term. By Assumption A1,

$$d(\widehat{\mathbb{P}}_t, \mathbb{Q}_t) \lesssim M_t^{-p}.$$

For the bias term, convexity of d in its second argument gives

$$d(\mathbb{Q}_t, \mathbb{P}_0) = d\left(\mathbb{P}_0, \frac{tm_1}{M_t}\,\overline{\mathbb{P}}_t^{\text{bias}} + \frac{m_2}{M_t}\sum_{j=1}^{t-1}\widehat{\mathbb{P}}_j\right)$$
$$\leq \frac{tm_1}{M_t}\,d(\mathbb{P}_0, \overline{\mathbb{P}}_t^{\text{bias}}) + \frac{m_2}{M_t}\sum_{j=1}^{t-1} d(\mathbb{P}_0, \widehat{\mathbb{P}}_j).$$

By the preceding lemma (Section C),

$$d(\overline{\mathbb{P}}_t^{\text{bias}}, \mathbb{P}_0) \lesssim t^{-\min\{q,1\}} \quad (\text{up to } \log t \text{ if } q = 1).$$

Since $M_t \asymp t(m_1 + m_2)$,

$$\frac{tm_1}{M_t} d(\overline{\mathbb{P}}_t^{\text{bias}}, \mathbb{P}_0) \lesssim t^{-\min\{q,1\}}.$$

Define

$$d_t := d(\widehat{\mathbb{P}}_t, \mathbb{P}_0), \qquad S_{t-1} := \sum_{j=1}^{t-1} d_j.$$

We obtain the basic recursion

$$d_t \lesssim M_t^{-p} + t^{-\min\{q,1\}} + \frac{m_2}{M_t}\,S_{t-1}. \tag{6}$$

Step 2: Recursion for the partial sums. Repeating the argument at step $t-1$ gives

$$d_{t-1} \lesssim M_{t-1}^{-p} + (t-1)^{-\min\{q,1\}} + \frac{m_2}{M_{t-1}} S_{t-2}.$$

Hence

$$S_{t-1} = d_{t-1} + S_{t-2} \lesssim M_{t-1}^{-p} + (t-1)^{-\min\{q,1\}} + \left(1 + \frac{m_2}{M_{t-1}}\right) S_{t-2}.$$

Iterating yields

$$S_{t-1} \lesssim \sum_{j=1}^{t-1}\left(M_j^{-p} + j^{-\min\{q,1\}}\right)\left[\prod_{k=j}^{t-2}\left(1 + \frac{m_2}{M_k}\right)\right]. \tag{7}$$

Step 3: Substituting back. Substituting (7) into (6) gives

$$d_t \lesssim M_t^{-p} + t^{-\min\{q,1\}} + \frac{m_2}{M_t}\sum_{j=1}^{t-2}\left(M_j^{-p} + j^{-\min\{q,1\}}\right)\left[\prod_{k=j}^{t-2}\left(1 + \frac{m_2}{M_k}\right)\right]$$
$$\lesssim M_t^{-p} + t^{-\min\{q,1\}} + (1-\alpha)\sum_{j=1}^{t-2}\left(M_j^{-p} + j^{-\min\{q,1\}}\right)\left[\frac{1}{M_t}\prod_{k=j}^{t-2}\left(1 + \frac{m_2}{M_k}\right)\right].$$

Using $M_t \asymp t(m_1 + m_2)$ and

$$\alpha = \frac{m_1}{m_1 + m_2}, \qquad \frac{m_2}{M_t} \asymp \frac{1-\alpha}{t},$$

the remainder of the proof follows the same argument as in the unbiased case:

$$\frac{1}{t}\prod_{k=j}^{t-2}\left(1 + \frac{1-\alpha}{k}\right) \asymp t^{-\alpha} j^{-(1-\alpha)}.$$

In this form, the three effects are additive:

$$\begin{aligned}
d_t &\lesssim M_t^{-p} + t^{-\min\{q,1\}} + (1-\alpha)\, t^{-\alpha} \sum_{j=1}^{t-2} \big(M_j^{-p} + j^{-\min\{q,1\}}\big) j^{-(1-\alpha)} \\
&\lesssim t^{-p} + t^{-\min\{q,1\}} + t^{-\alpha} \sum_{j=1}^{t-2} \left(j^{-1-(p-\alpha)} + j^{-1-(\min\{q,1\}-\alpha)}\right) \\
&\lesssim t^{-p} + t^{-\min\{q,1\}} + \begin{cases} t^{-\alpha}, & p > \alpha, \\ t^{-\alpha}\log t, & p = \alpha, \\ t^{\alpha-p}, & p < \alpha, \end{cases} + \begin{cases} t^{-\alpha}, & \min\{q,1\} > \alpha, \\ t^{-\alpha}\log t, & \min\{q,1\} = \alpha, \\ t^{-\min\{q,1\}}, & \min\{q,1\} < \alpha, \end{cases} \\
&\lesssim t^{-p} + t^{-\min\{q,1\}} + t^{-\min\{p,\alpha\}} + t^{-\min\{q,1\}} \\
&\lesssim t^{-\min\{p,q,\alpha\}},
\end{aligned}$$

up to a log factor. All asymptotic inequalities may be replaced by their equivalents in probability.

□

D Additional Simulation Details

This section provides a detailed description of the pipeline for all simulations, including the justification for the baseline rates of convergence, underlying data-generating process, numerical grids, estimators, metrics, and the recursive procedure used to combine real and synthetic data, as well as additional figures and tables.

D.1 Baseline Rates Used in the Experiments

The theoretical predictions of CRT and BCRT depend on the baseline convergence rate $d(\hat{\mathbb{P}}_n, \mathbb{P}_0) = O_p(n^{-p})$. For all estimators considered in our experiments we take $p = 1/2$, corresponding to the standard root-n rate. For empirical distribution estimation this follows from classical results in W_1 (Fournier and Guillin, 2015) and for MMD under bounded kernels (Gretton et al., 2012). For the KDE experiments, choosing $h_n \asymp n^{-1/2}$ preserves the same rate by a straightforward smoothing argument.

D.2 Additional CRT Simulation details: ECDF and KDE

In the CRT simulations, the target distribution $\mathbb{P}_0$ is a 1-dimensional mixture of two Gaussian components. This smooth distribution serves as the ground truth throughout both experiments. A numerical grid covering the effective support of $\mathbb{P}_0$ is constructed once and used for all deterministic evaluations of distributional discrepancies. The true density and CDF of $\mathbb{P}_0$ are available in closed form on this grid.

Both simulations follow the CRT procedure in the section **Contaminated Recursive Training**. At iteration t, a batch of m_1 real samples drawn from $\mathbb{P}_0$ is added to the accumulated dataset, along with $m_2 = ((1-\alpha)/\alpha)m_1$ synthetic samples drawn from the previous generator $\widehat{\mathbb{P}}_{t-1}$.

In the CRT simulations, we use two estimators: the ECDF as the estimator $\widehat{\mathbb{P}}_t$, and a KDE whose bandwidth is chosen to correspond to a smoothness parameter p; the KDE thus serves as a concrete model with known uncontaminated convergence rate. In both settings, the estimator at iteration t defines a density and distribution function $(\widehat{f}_t, \widehat{F}_t)$ evaluated on the numerical grid.

The primary metric is the W_1 distance,

$$W_1(\widehat{\mathbb{P}}_t, \mathbb{P}_0) \approx \int \left| F_0(x) - \widehat{F}_t(x) \right| dx,$$

computed using the trapezoidal rule on the fixed grid. MMD is evaluated using the same plug-in approach applied to the grid-based density estimates.

To estimate convergence rates, we record the sequence of losses $\{d(\widehat{\mathbb{P}}_t, \mathbb{P}_0)\}$ over iterations and fit a power law of the form

$$\log d(\widehat{\mathbb{P}}_t, \mathbb{P}_0) = a + b \log M_t.$$

After discarding an initial burn-in period, the fitted slope b yields the empirical rate. When $\alpha = p$, CRT theory predicts a logarithmic phase transition; in this case, losses are pre-normalized by $\log t$ before regression. The resulting values of b are reported as the observed convergence rates under recursive contamination. All experimental parameters for model training and sampling are summarized in Table 1.

Finally, we show the estimated densities in each case, for every α in Figure 6 and Figure 8. Additionally, examples of distributional loss curves and fitted slopes are provided for each alpha in Figure 7 and Figure 9.

Parameter	Value	Description
m_1	50	Real samples per iteration
α	$\{0.1, \ldots, 0.9, 1.0\}$	Real-data fraction
T	2000	Total CRT iterations
n_{reps}	100	Number of repetitions
m_{grid}	200	Grid size for deterministic evaluation
$[x_{\min}, x_{\max}]$	Mixture-based	Grid interval for density/CDF evaluation
w_1	0.35	Mixture weight
μ_1, σ_1	$-2.0,\ 0.8$	First Gaussian component
μ_2, σ_2	$1.0,\ 1.3$	Second Gaussian component
h_0	0.5	Base KDE bandwidth

Table 1: CRT Simulation parameters for KDE and ECDF estimators.

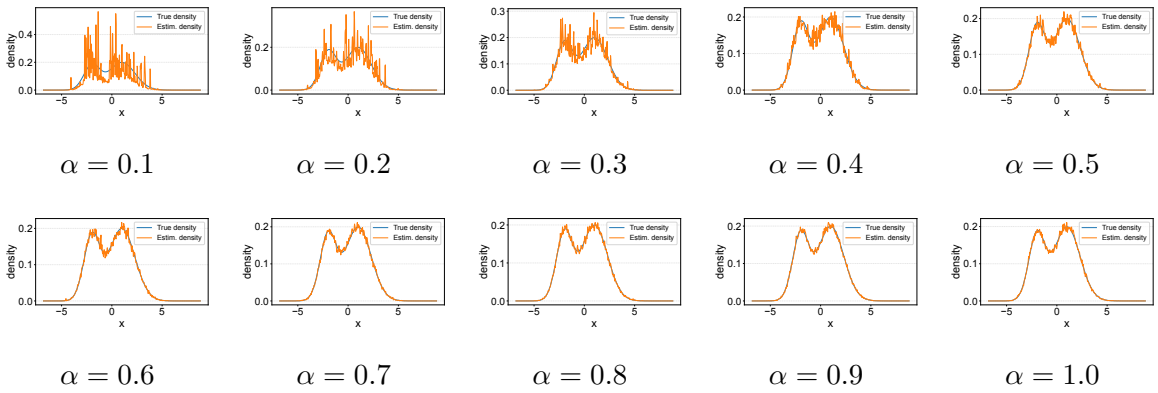
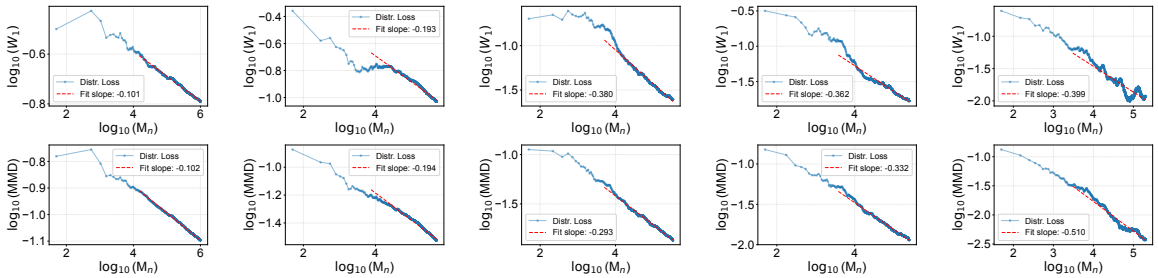
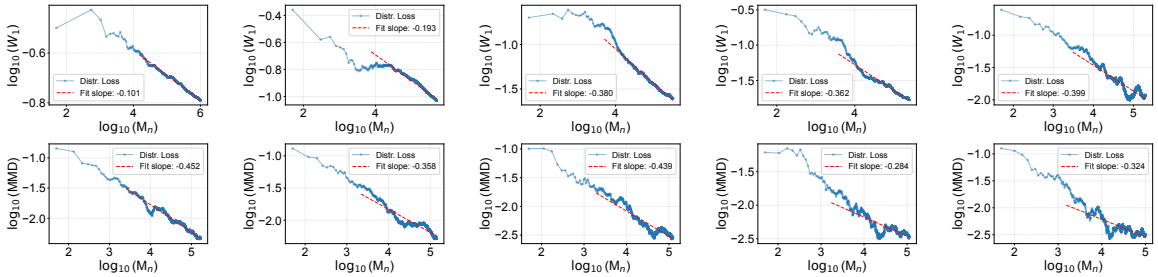


Figure 6: CRT Simulation (ECDF Estimator). Final output distributions $\widehat{\mathbb{P}}_T$ for varying values of α (real-data fraction) where all models are run for the same number of CRT iterations. This fixed compute budget for all models was chosen to minimize the effect of numerical plateaus in the loss when computing optimal transport distances after models have fully converged. Under this fixed compute budget, convergence improves as α increases; however, for all α we observe the predicted rate of convergence.

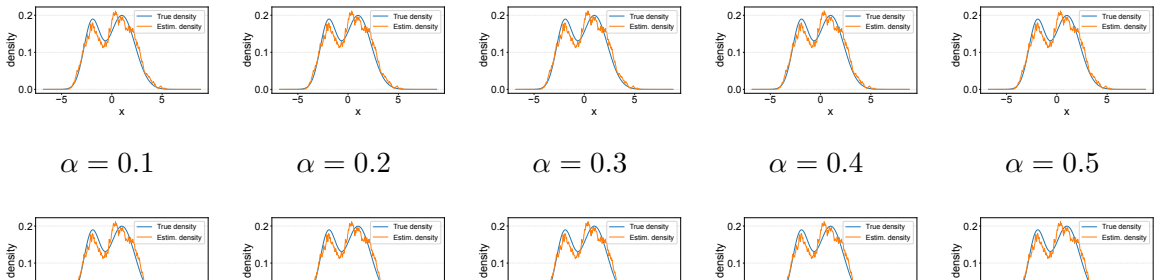


$\alpha = 0.10$ $\alpha = 0.20$ $\alpha = 0.30$ $\alpha = 0.40$ $\alpha = 0.50$



$\alpha = 0.60$ $\alpha = 0.70$ $\alpha = 0.80$ $\alpha = 0.90$ $\alpha = 1.00$

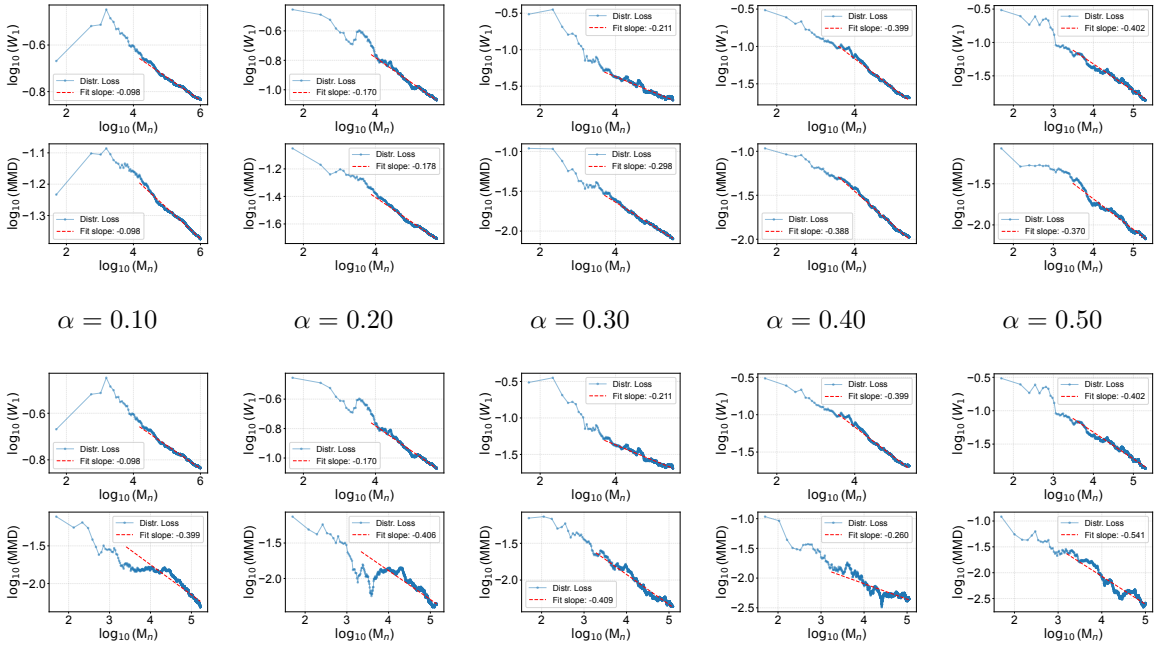
Figure 7: CRT Simulation (ECDF Estimator). W_1 and MMD distributional losses and fitted slopes for varying values of α (real-data fraction) for a single run.



$\alpha = 0.1$ $\alpha = 0.2$ $\alpha = 0.3$ $\alpha = 0.4$ $\alpha = 0.5$

$\alpha = 0.6$ $\alpha = 0.7$ $\alpha = 0.8$ $\alpha = 0.9$ $\alpha = 1.0$

Figure 8: CRT Simulation (KDE Estimator). Final output distributions $\widehat{\mathbb{P}}_T$ for varying values of α (real-data fraction) where all models are run for the same number of CRT iterations. This fixed compute budget for all models was chosen to minimize the effect of numerical plateaus in the loss when computing optimal transport distances after models have fully converged. Under this fixed compute budget, convergence improves as α increases; however, for all α we observe the predicted rate of convergence.



$\alpha = 0.10$ $\alpha = 0.20$ $\alpha = 0.30$ $\alpha = 0.40$ $\alpha = 0.50$

$\alpha = 0.60$ $\alpha = 0.70$ $\alpha = 0.80$ $\alpha = 0.90$ $\alpha = 1.00$

Figure 9: CRT Simulation (KDE Estimator). W_1 and MMD distributional losses and fitted slopes for varying values of α (real-data fraction) for a single run.

D.3 Additional CRT Simulation Details: WGAN

In this simulation, the target distribution $\mathbb{P}_0$ is the same as in the previous experiment. The generator $\widehat{\mathbb{P}}_t$ is implemented as a fully connected feedforward network. A latent vector $z \in \text{Unif}(0,1)$ is mapped through three 64-neuron hidden layers, LeakyReLU activations, and a final linear layer producing points in $\mathbb{R}^1$.

At iteration t, a batch of m_1 new samples from $\mathbb{P}_0$ is appended to the dataset, together with $m_2 = ((1-\alpha)/\alpha)m_1$ synthetic samples generated from the previous distribution $\widehat{\mathbb{P}}_{t-1}$. At each CRT iteration, the generator is completely re-initialized, and is then trained for k epochs using minibatch stochastic gradient descent on the current accumulated dataset of real and synthetic samples. Optimization uses Adam with a fixed learning rate and weight decay.

The training loss is a differentiable optimal-transport discrepancy. We employ an empirical W_1 loss computed via quantile (inverse CDF) matching from the Python OT package (Flamary et al., 2021). At each iteration, convergence is assessed by computing an empirical W_1 distance between $\widehat{\mathbb{P}}_t$ and $\mathbb{P}_0$, using fresh evaluation batches of synthetic samples against a fixed large sample of the target distribution. This produces a sequence $W_1(\widehat{\mathbb{P}}_t, \mathbb{P}_0)$ indexed by M_t.

To estimate convergence rates, we fit a power law of the form

$$\log W_1(\widehat{\mathbb{P}}_t, \mathbb{P}_0) = a + b \log M_t,$$

after discarding an initial burn-in period. When $\alpha = p$, CRT theory predicts a logarithmic phase transition; in this regime, the losses are normalized by $\log(t)$ prior to regression. The fitted slope b is reported as the observed convergence rate of the recursive neural estimator. Examples of final densities and loss curves are shown in Figure 10, 11. All experimental parameters for model training and sampling are summarized in Table 2.

Parameter	Value	Description
$m_1 + m_2$	500	Total samples per iteration
α	$\{0.1, 0.2, \ldots, 1.0\}$	real-data fraction
T	500	Total CRT (outer training loop) iterations
k	25	Training epochs per CRT iteration (inner training loop)
Latent dimension	1	Dimension of z
Latent distribution	Uniform(0,1)	Input to generator
Network width	64	Hidden layer width
Network layers	3	Number of hidden layers
Activation	LeakyReLU(0.02)	Nonlinearity
Optimizer	Adam	Generator optimization
Learning rate	2×10^{-4}	Step size
Weight decay	1×10^{-3}	Optimizer weight decay
Batch size	1024	Training minibatch size
Loss type	Quantile W_1	W_1 loss using POT package
Evaluation Samples	200000	Number of samples (real, synthetic) for evaluation W_1

Table 2: CRT Simulation parameters for WGAN estimator.

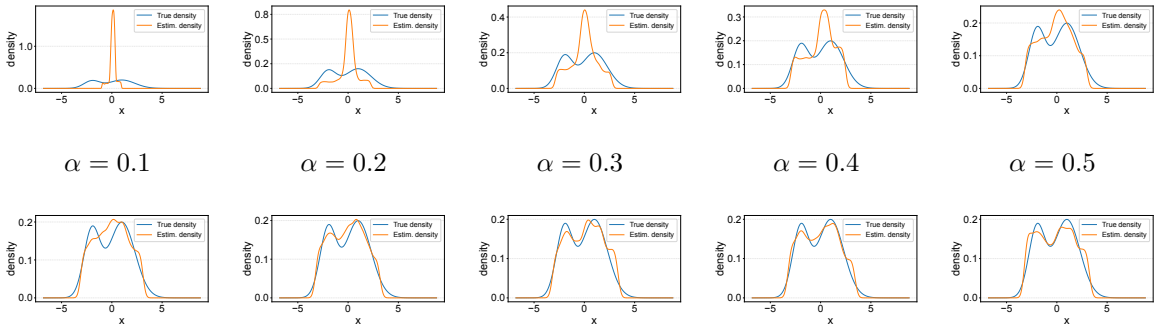


Figure 10: CRT Simulation (WGAN Estimator). Final output distributions $\widehat{\mathbb{P}}_T$ for varying values of α (real-data fraction) visualized using KDE, where all models are run for the same number of CRT iterations. This fixed compute budget for all models was chosen to minimize the effect of numerical plateaus in the loss when computing optimal transport distances. Under this fixed compute budget, models at greater α are shown to converge more easily, however, for all α we observe the predicted rate of convergence.

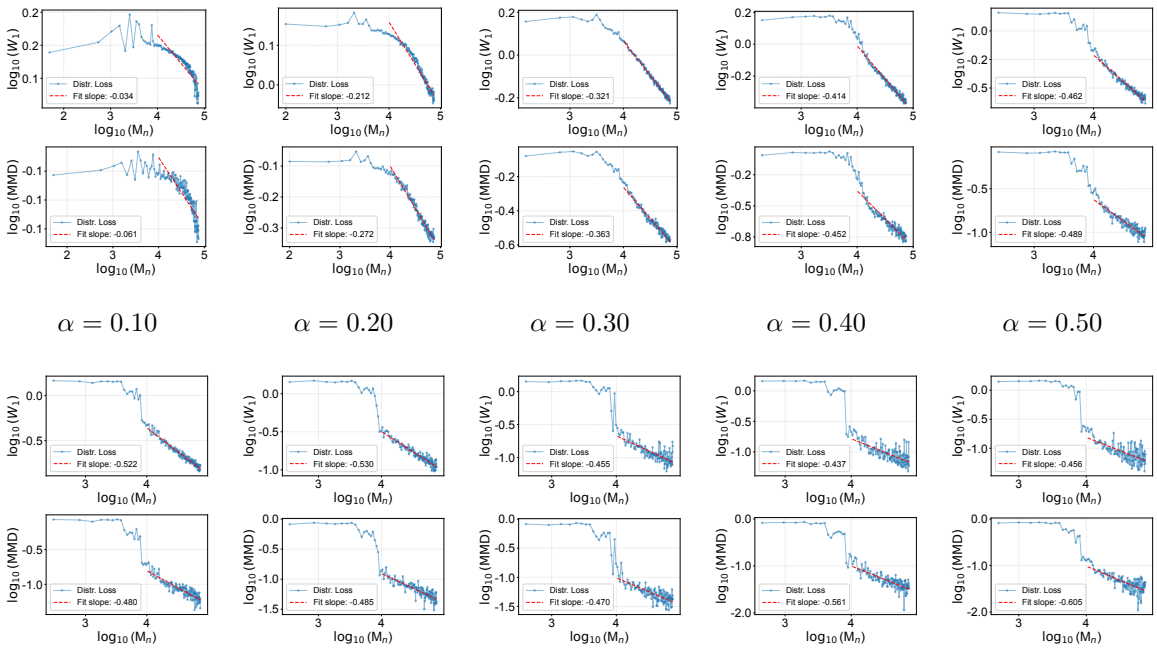


Figure 11: CRT Simulation (WGAN Estimator). W_1 and MMD distributional losses and fitted slopes for varying values of α (real-data fraction) for a single run.

D.4 Additional BCRT Simulation Details

In the BCRT setting, we use the same one-dimensional Gaussian mixture target distribution $\mathbb{P}_0$ as in the CRT simulations, with identical mixture weights and component parameters. The numerical grid used to evaluate distributional discrepancies, as well as the closed-form density and CDF of $\mathbb{P}_0$ on this grid, are constructed exactly as described in the previous subsection and reused throughout.

We follow the BCRT procedure of **Biased Contaminated Recursive Training**. At iteration t, a batch of m_1 real samples is added to the accumulated dataset, together with $m_2 = ((1-\alpha_{\text{real}})/\alpha_{\text{real}})m_1$ synthetic samples drawn from the previous distribution $\widehat{\mathbb{P}}_{t-1}$. Synthetic samples are generated by resampling from the accumulated dataset and adding Gaussian noise with variance equal to the current KDE bandwidth. Unlike in the CRT simulations, the real data stream here is biased. At iteration t, real samples are drawn from a contaminated distribution

$$\mathbb{P}_t^{\text{bias}} = (1-\text{bias}_t)\mathbb{P}_0 + \text{bias}_t\,\mathcal{N}(\mu_3, \sigma_3),$$

where $(\mu_3, \sigma_3) = (3.0, 1.0)$. The contamination level decays polynomially as

$$\text{bias}_t = 0.2\,(t+5)^{-q},$$

with decay rate parameter $q > 0$.

The estimator $\widehat{\mathbb{P}}_t$ is a KDE, as in the CRT case. The bandwidth follows a deterministic decay schedule

$$h_t = h_0\,t^{-p},$$

with base bandwidth $h_0 = 2$ and smoothness parameter $p = 0.5$. The estimator defines a density-CDF pair $(\widehat{f}_t, \widehat{F}_t)$ evaluated on the fixed grid, using trapezoidal integration and renormalization as in previous simulations.

Performance is evaluated using the same deterministic plug-in metrics as in Simulations 1 and 2: the W_1 distance,

$$W_1(\widehat{\mathbb{P}}_t, \mathbb{P}_0) \approx \int \left|F_0(x) - \widehat{F}_t(x)\right| dx,$$

and squared MMD, with the MMD computed using a Gaussian kernel on the evaluation grid.

To estimate convergence rates, we record the sequence of losses $\{d(\widehat{\mathbb{P}}_t, \mathbb{P}_0)\}$ over iterations and fit a power law of the form

$$\log d(\widehat{\mathbb{P}}_t, \mathbb{P}_0) = a + b \log M_t,$$

where M_t denotes the total number of accumulated samples. After discarding an initial burn-in period, the fitted slope b yields the estimated empirical convergence rate. The theory predicts that the effective rate is governed by $\min(p, \alpha, q)$; empirical estimates are compared against this benchmark in the main text. Simulation parameters specific to this

section are summarized below. Additionally, examples of final fitted models are visualized in Figure 12 and Figure 14, and examples of training curves in Figure 13 and Figure 15. All experimental parameters for model training and sampling are summarized in Table 3 and Table 4.

Parameter	**Value**	**Description**
m_1	25	Real samples per iteration
α	$\{0.25, 0.5, 0.75\}$	real-data fraction
q	$\{0.25, 0.5, 0.75\}$	Bias Decay Rate
T	3000	Total BCRT iterations
n_{reps}	100	Number of repetitions
m_{grid}	200	Grid size for deterministic evaluation
$[x_{\min}, x_{\max}]$	Mixture-based	Grid interval for density/CDF evaluation
w_1	0.35	Mixture weight
μ_1, σ_1	$-2.0,\ 0.8$	First Gaussian component
μ_2, σ_2	$1.0,\ 1.3$	Second Gaussian component
μ_3, σ_3	$3.0,\ 1.0$	Bias Gaussian component
h_0	2.0	Base KDE bandwidth

Table 3: BCRT Simulation (ECDF Estimator) experimental parameters.

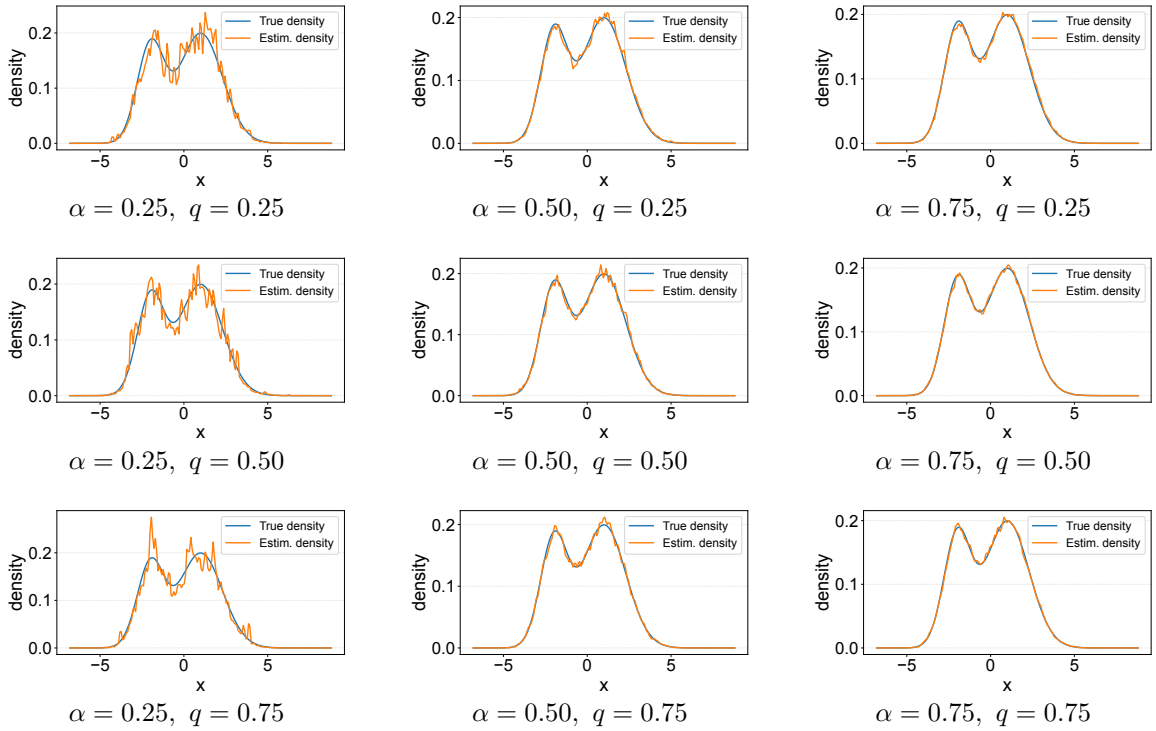
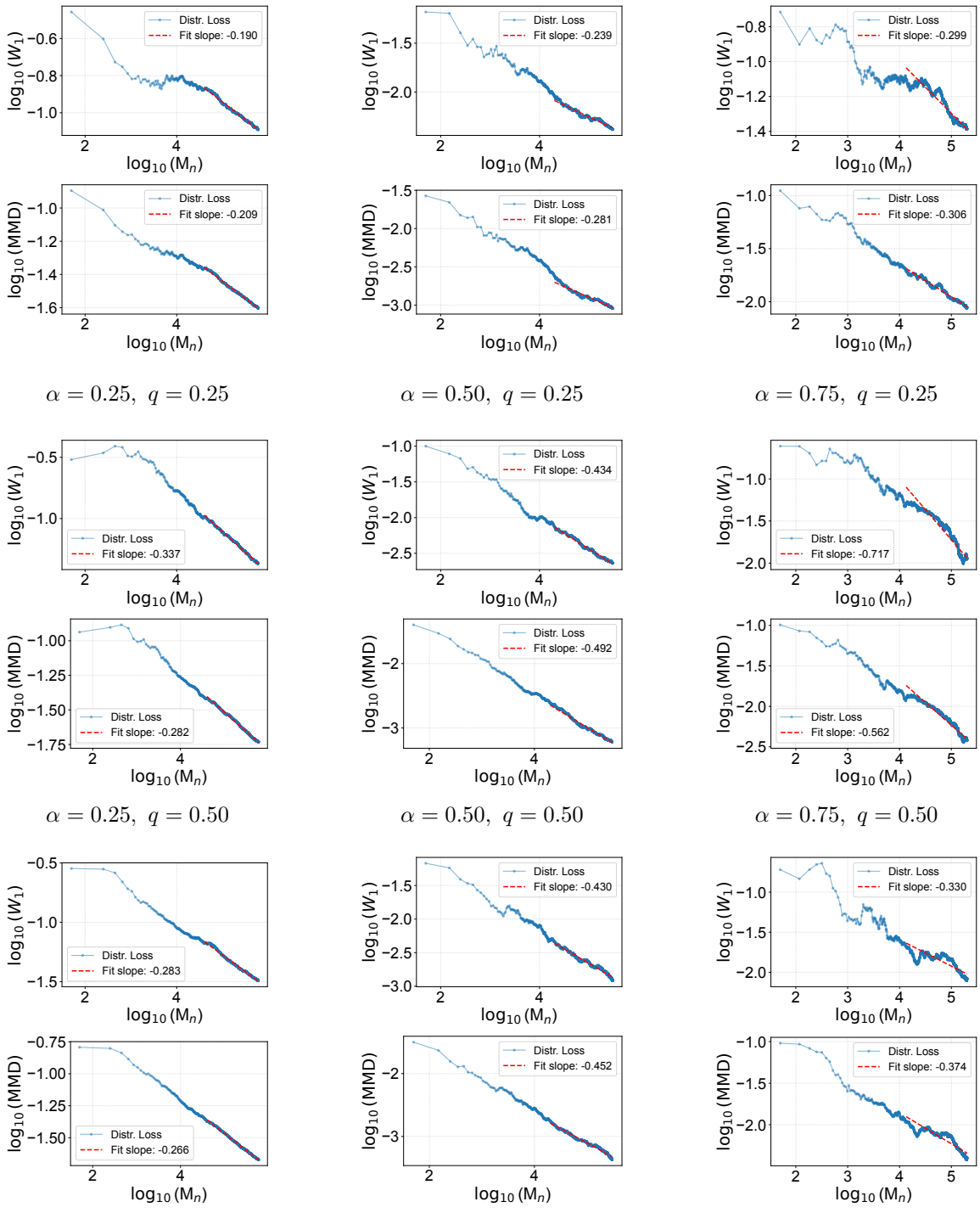


Figure 12: BCRT Simulation (ECDF Estimator). Final output distributions across combinations of real-data fraction $\alpha \in \{0.25, 0.5, 0.75\}$ and bias decay rate $q \in \{0.25, 0.5, 0.75\}$.



$\alpha = 0.25,\ q = 0.25$ $\alpha = 0.50,\ q = 0.25$ $\alpha = 0.75,\ q = 0.25$

$\alpha = 0.25,\ q = 0.50$ $\alpha = 0.50,\ q = 0.50$ $\alpha = 0.75,\ q = 0.50$

$\alpha = 0.25,\ q = 0.75$ $\alpha = 0.50,\ q = 0.75$ $\alpha = 0.75,\ q = 0.75$

Figure 13: BCRT Simulation (ECDF Estimator). W_1 and MMD distributional losses and fitted slopes for varying values of α and q for a single run. Slopes are fitted after dropping the first 200 out of 3000 iterations.

Parameter	Value	Description
m_1	50	Real samples per iteration
α	$\{0.25, 0.5, 0.75\}$	real-data fraction
q	$\{0.25, 0.5, 0.75\}$	Bias Decay Rate
T	3000	Total BCRT iterations
n_{reps}	100	Number of repetitions
m_{grid}	200	Grid size for deterministic evaluation
$[x_{\min}, x_{\max}]$	Mixture-based	Grid interval for density/CDF evaluation
w_1	0.35	Mixture weight
μ_1, σ_1	$-2.0,\ 0.8$	First Gaussian component
μ_2, σ_2	$1.0,\ 1.3$	Second Gaussian component
μ_3, σ_3	$3.0,\ 1.0$	Bias Gaussian component
h_0	2.0	Base KDE bandwidth

Table 4: BCRT Simulation (KDE Estimator) experimental parameters

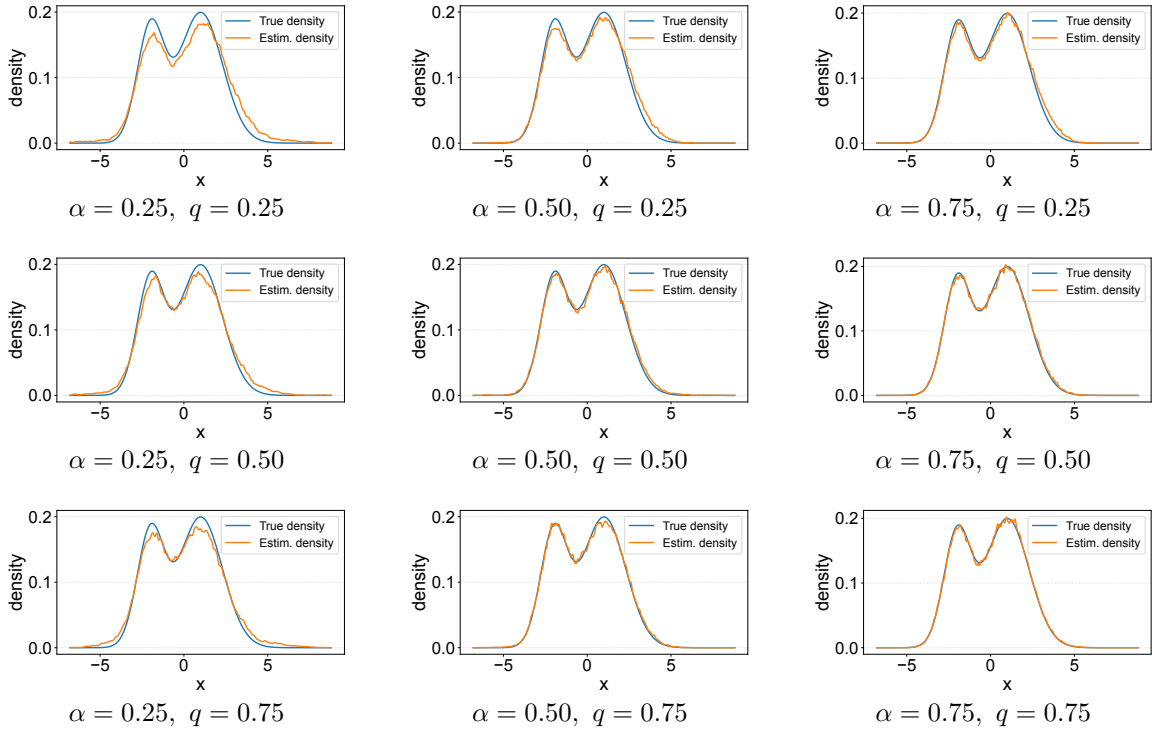
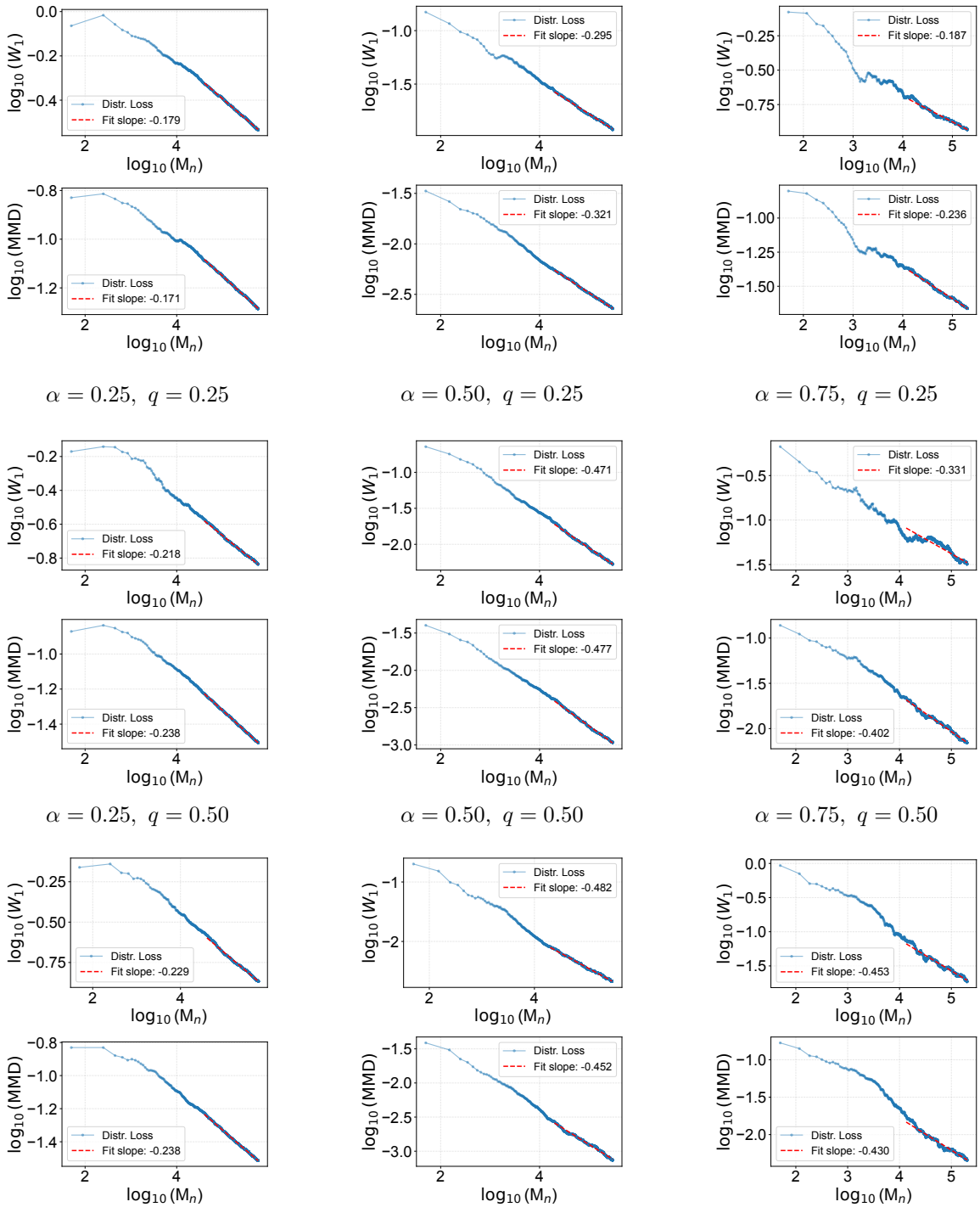


Figure 14: BCRT Simulation (KDE Estimator). Final output distributions across combinations of real-data fraction $\alpha \in \{0.25, 0.5, 0.75\}$ and bias decay rate $q \in \{0.25, 0.5, 0.75\}$.



$\alpha = 0.25,\ q = 0.25$ $\alpha = 0.50,\ q = 0.25$ $\alpha = 0.75,\ q = 0.25$

$\alpha = 0.25,\ q = 0.50$ $\alpha = 0.50,\ q = 0.50$ $\alpha = 0.75,\ q = 0.50$

$\alpha = 0.25,\ q = 0.75$ $\alpha = 0.50,\ q = 0.75$ $\alpha = 0.75,\ q = 0.75$

Figure 15: BCRT Simulation (KDE Estimator). W_1 and MMD distributional losses and fitted slopes for varying values of α and q for a single run. Slopes are fitted after dropping the first 200 out of 3000 iterations.

E Additional MNIST Experiment Details

In this section, we present experimental parameters of the DDPM diffusion model trained on MNIST under CRT. We use data from the 60,000 MNIST training set where each training sample is used by the optimizer an equal number of epochs, as per the setup in Theorem 3.4. Each iteration builds on the previous model's generator, such that every new sample is trained for a fixed number of epochs and all data points are used by the optimizer for an equal number of training steps. To ensure uniform representation, the newly introduced real data points at each step are explicitly sampled to maintain a balanced distribution across all ten digit classes. A new sample is produced at each CRT iteration to visualize the output of each iteration's generator.

Parameter	**Value**	**Description**
α	$\{0.0, 0.25, 0.50, 0.75, 1.00\}$	real-data fraction (Sweep default)
Samples per step	300	Total combined batch size per iteration
m_1	$\alpha \times 300$	Real samples per iteration
T	200	Total CRT iterations
T_{diff}	400	Number of diffusion steps
Epochs	100	Epochs of training per data iteration
Batch Size	100	Training batch size
Schedule	Cosine	Optimizer schedule
β_{start}	1×10^{-4}	Noise variance schedule
β_{end}	2×10^{-2}	Noise variance schedule
Time dimension	128	Embedding dimension for time conditioning
Base channels	64	Base channel multiplier for the UNet

Table 5: Experimental parameters for MNIST experiment.

We present the details of the network architecture and training objective. We parameterize the diffusion noise predictor using a lightweight UNet-style convolutional network with residual blocks, attention mechanisms, and time conditioning. The timestep t is embedded via a sinusoidal encoding and passed through a small MLP. To improve training stability, the mean squared error loss for epsilon prediction is scaled using Min-SNR weighting ($\gamma = 5.0$). All experimental parameters for model training are summarized in Table 5.

Architecturally, the model follows a two-level UNet structure. The encoder consists of an initial convolution followed by residual blocks, downsampling from 28x28 to 14x14, where a scaled dot-product attention block is applied. A second downsampling reduces the spatial resolution to 7x7 while increasing the channel width from 64 to 128. The bottleneck consists of residual blocks interleaved with a secondary attention block at the 7x7 resolution. The decoder mirrors the encoder with two stages of upsampling, skip connections, and a final attention block at the 14x14 resolution. The final convolution maps features back to the single output channel predicting the noise.

F Additional LLM Experiment Details

In this section we provide additional details on the contaminated recursive training experiment using LLMs. The dataset we use is the WikiText-103 corpus, consisting of high-quality Wikipedia articles, pre-partitioned into training and test splits. We use the standard release training split for fine-tuning, and the standard release test split for independent evaluation of the text diversity metrics. The training set consists of approximately 103 million tokens, while the test set contains approximately 0.25 million tokens. Both sets of data are further divided into disjoint blocks of 1,024 tokens for training and evaluation.

The model used for all experiments is the 124M parameter version of GPT-2 from OpenAI with its standard tokenizer, initialized from publicly available pretrained weights and fine-tuned within the recursive training framework described above.

F.1 Training

Fine-tuning was performed using the standard autoregressive objective, in which the model is optimized to predict each token given its preceding context. For a sequence of tokens $(x_1, \ldots, x_L)$, the objective is

$$\mathcal{L} = -\sum_{i=1}^{L} \log p_\theta(x_i \mid x_{<i}),$$

where θ denotes the model parameters.

Training Parameter	**Value**	**Description**
Block size	1024	Tokens per training example
m_{total}	1024	Blocks per iteration dataset D_t
α	$\{0.0, 0.10, 1.0\}$	real-data fraction
T	50	Number of recursive iterations
Epochs	1	Epochs per iteration
Learning rate	2×10^{-5}	AdamW optimizer
Batch size	1	Micro-batch size
Gradient accumulation.	32	Gradient accumulation steps
Schedule	Linear warmup + decay	Learning rate schedule
Sampling Parameter		
Generation p	0.95	Nucleus sampling parameter
Temperature	1.0	Sampling temperature
Repetition penalty	1.05	Decoding penalty

Table 6: Experimental parameters for recursive LLM training experiment. Note that the effective batch size for the experiment is the micro-batch size multiplied by the number of gradient accumulations steps.

Optimization was performed using the AdamW optimizer with a learning rate of 2×10^{-5}, weight decay, and linear learning rate warmup followed by decay. Gradient accumulation was used to achieve a larger effective batch size under memory constraints. Mixed-precision (FP16) training was employed when supported by the hardware.

At each iteration, the model was trained for a fixed number of full passes (epochs) over the constructed dataset D_t at outer training iteration t, such that no training token was re-used between iterations. At each t, D_t consists of a mixture of real data blocks and synthetic data blocks generated by the generator trained at $t-1$, where the fraction of real data blocks is defined by α. Model parameters were updated between iterations without reinitialization unless otherwise specified, as in previous experiments, to ensure all data points are utilized by the optimizer an equal number of times as per Theorem 3.4. All experimental parameters for model training and sampling are summarized in Table 6.

F.2 Evaluation

All evaluations are performed on the pre-partitioned test set, which was not directly used for any training or fine-tuning. The primary evaluation metrics used were 1) model perplexity, reflecting the model's predictive accuracy, 2) average per-token predictive entropy, measuring the model output uncertainty, and 3) distinct$-n$ for $n = 1, 2, 3$ to measure output diversity. Perplexity and entropy were evaluated over the independent test set, while distinct$-n$ were evaluated over a new sample of the trained models.