ESCAPING MODEL COLLAPSE VIA SYNTHETIC DATA VERIFICATION: NEAR-TERM IMPROVEMENTS AND LONG-TERM CONVERGENCE

Anonymous authors
Paper under double-blind review

ABSTRACT

Synthetic data has been increasingly used to train frontier generative models. However, recent study raises key concerns that iteratively retraining a generative model on its self-generated synthetic data may keep deteriorating model performance, a phenomenon often coined *model collapse*. In this paper, we investigate ways to modify the synthetic retraining process to avoid model collapse, and even possibly help reverse the trend from collapse to improvement. Our key finding is that by injecting information through an external synthetic data verifier, whether a human or a better model, synthetic retraining will not cause model collapse. Specifically, we situate our theoretical analysis in the fundamental linear regression problem, showing that verifier-guided retraining yields early improvements when the verifier is accurate, and in the long run the parameter estimate converges to the verifier's knowledge center. Our theory predicts that the performance of synthetic retraining will have early gains but eventually plateaus or even reverses, unless the verifier is perfectly reliable. Indeed, our experiments on both linear regression as well as Conditional Variational Autoencoder (CVAE) trained on MNIST data also confirm these theoretical insights.

1 INTRODUCTION

The use of synthetic data has gained significant traction due to its ability to reduce data collection costs and enhance privacy protection, with applications in computer vision (Wood et al., 2021), healthcare (Azizi et al., 2021; Santangelo et al., 2025), and finance (Potluru et al., 2023). A growing body of work has demonstrated that training with synthetic data can improve performance, especially when real data are scarce or expensive to obtain (Shrivastava et al., 2017; Doersch & Zisserman, 2019; Liu et al., 2023; Tremblay et al., 2018). However, recent studies caution that recursively training models on synthetic data alone can lead to a degradation of quality, a phenomenon often termed *model collapse* (Shumailov et al., 2024; Dohmatob et al., 2024a;b;c; Alemohammad et al., 2023; Gerstgrasser et al., 2024).

In practice, synthetic data are rarely used in raw form. Instead, practitioners often apply filtering steps to remove low-quality synthetic samples before retraining. For example, in natural language generation, synthetic sentences may be screened using grammar checkers or LLM-as-a-judge pipelines; in computer vision, synthetic images may be filtered using pretrained discriminators or human annotation; in recommendation and preference learning, synthetic feedback is often validated against external heuristics or known user signals (Tu et al., 2024; Iskander et al., 2024; Lupidi et al., 2024; Lampis et al., 2023; Zhang et al., 2024). A common abstraction across these approaches is the use of a *verifier* that evaluates candidate synthetic samples and retains only those passing verification.

While intuitively appealing, it remains unclear whether such verifier-based filtering truly improves model training. Existing studies provide partial insights in specific tasks—such as classification with noisy labels (Feng et al., 2024) or preference-driven data selection (Ferbach et al., 2024)—but a general statistical framework for understanding the impact of verifiers on retraining dynamics is still lacking. In particular, we lack a systematic theory that characterizes both the short-term benefits of verifier filtering and its long-term consequences for iterative retraining.

Our contributions. We develop a statistical framework to analyze retraining on verified synthetic data, focusing on linear regression – a canonical model for principled study of model collapse (Dohmatob et al., 2024a;b; Gerstgrasser et al., 2024) – while also empirically extending insights to real-world generative settings. Our contributions can be summarized as follows:

- *Does verification help?* We show that verifier filtering can indeed improve model training. Our results provide formal conditions under which retraining on verified synthetic data yields performance gains relative to unfiltered retraining.
- *When does it help?* We characterize the regimes in which verification leads to improvement versus degradation, highlighting the role of synthetic sample size, verifier bias, and verifier strength. This provides a concrete answer to *when* verification is beneficial.
- *Why does it help?* We identify the mechanism underlying these improvements: a verifier-induced bias–variance trade-off in the short term, and convergence of the retrained model toward the verifier's knowledge center in the long term. These results reveal distinct asymptotic performance phases depending on verifier quality.
- *Empirical validation.* We validate our theory through both simulations and real-data experiments, including linear regression and conditional variational autoencoder (CVAE) models, showing that our theoretical predictions align with observed training dynamics.

These together offer a comprehensive understanding about the role of external verifiers in synthetic retraining, helping explain *whether*, *when*, and *why* verification can mitigate model collapse.

1.1 Related Work

Understanding and mitigating model collapse. Recent research has shown that relying heavily on synthetic data for training can lead to *model collapse*, a degradation in model quality over successive training iterations. Intuitively, model collapse refers to the phenomenon where repeated retraining on synthetic data produces worse models rather than better ones. A number of recent studies have provided evidence of collapse. For instance, Shumailov et al. (2024) showed that recursively training solely on synthetic data induces distribution shift that leads to collapse. Dohmatob et al. (2024b) demonstrated that even small proportions of synthetic data can harm performance. In linear models, Dohmatob et al. (2024a) analyzed collapse mechanisms explicitly, while Dohmatob et al. (2024c) linked degradation to altered neural scaling laws. To mitigate collapse, some studies propose accumulating data across iterations rather than replacing it entirely, which stabilizes training (Gerstgrasser et al., 2024; Dey & Donoho, 2024). Others, such as Alemohammad et al. (2023), argue that only incorporating fresh data fully avoids collapse.

However, showing that collapse does not occur is not sufficient. Ultimately, the goal of retraining is not merely to avoid deterioration but to *achieve improvement*, since better models are the essential objective in practice. Yet prior work has largely stopped at diagnosing collapse or proposing strategies that stabilize performance, without demonstrating conditions under which retraining can strictly improve models. This gap motivates our focus on verifier-filtered synthetic data, an approach closely aligned with industry practice, where synthetic samples are routinely refined through external feedback mechanisms. By analyzing this setting, we provide a theoretical foundation for when and why retraining can lead to genuine model improvement.

Filtering and selecting synthetic data. A complementary line of work investigates filtering strategies to improve synthetic data quality. Empirical studies have shown that training on filtered synthetic data can mitigate collapse and sometimes even enhance performance (Zhang et al., 2024; Lampis et al., 2023; Haluptzok et al., 2022; Zelikman et al., 2022; Patwa et al., 2024). These results suggest that filtering may offer a pathway toward improvement rather than mere stabilization. Theoretically, Ferbach et al. (2024); Wei & Zhang (2025) interprets curation as a form of implicit preference optimization. In their setting, synthetic data are curated according to a reward function, and the analysis is carried out at the population level: with infinitely many curated samples and repeated retraining, the model distribution converges to the highest-reward level set, so the generator maximizes expected reward but collapses onto those high-reward regions. In contrast, our work takes a parameter-estimation perspective in linear regression: we study how a verifier-guided filtering rule affects the finite-sample MSE of the estimator and explicitly characterize how the estimator depends on the verifier's accuracy, the strictness of the verification rule, and the synthetic sample size. This allows us to describe both when verifier-filtered synthetic retraining improves estimation and

how the long-run behavior depends on these verifier properties and on the amount of synthetic data. Feng et al. (2024) analyzes verifier-based filtering in classification, modeling the verifier by a single error-rate parameter. They identify a sharp phase transition: filtering either achieves perfect accuracy or complete failure, depending on verifier quality. In contrast, our analysis provides a more nuanced characterization. We show that in regression, performance varies smoothly with the verifier's bias and variance, rather than undergoing a sharp threshold effect. Moreover, we provide finite-sample rates that explicitly capture the interplay between real and synthetic data sizes. These distinctions highlight that while empirical work suggests filtering can drive improvement, a comprehensive theoretical understanding of the transition dynamics has been lacking. Our framework aims to fill this gap by rigorously analyzing verifier-filtered retraining under a linear model, thereby offering insights into when filtering not only prevents collapse but also yields strict improvement. Amin et al. (2025) assume a strong, reliable quality function and focus on how an external labeler aids learning under this fixed filtering mechanism. In contrast, we treat the generator as the previous (potentially imperfect) model and study how the verifier's quality itself drives improvement, plateau, or collapse during retraining. Zhu et al. (2024) analyze collapse in synthetic retraining and propose data editing to avoid it, but their theory does not show performance gains. Our framework instead leverages external knowledge to produce higher-quality synthetic data that improves the model, not just prevents collapse.

Comparison with reinforcement learning with verified rewards (RLVR). Our approach share some conceptual similarity with RLVR that assigns a verified reward value for each data sample (Guo et al., 2025; Yu et al., 2025), whereas our problem uses a verifier to pick a subset of data. However, RLVR fundamentally differs from our setting on various aspects. While indeed being a useful approach, RLVR is only applicable in settings where reward signals are clearly verifiable (Guo et al., 2025; Wu & Choi; Yu et al., 2025). In contrast, many important training settings lack reliable reward functions. Even alignment procedures avoid absolute reward assignments and instead rely on pairwise comparison signals (e.g., RLHF (Ouyang et al., 2022)) because scalar rewards are often difficult and noisy to define, for instance, evaluating image quality or open-ended language generation with a single numerical reward is inherently subjective. In contrast, binary accept/reject filtering is much less noisy and widely used in practice. This "generate–verify–retrain" paradigm is also a core primitive in modern LLM training pipelines, including large-scale systems such as DeepSeek-Coder (Guo et al., 2025), and has proven to be stable, scalable, and easy to integrate. These practical advantages, along with the growing research interest in verifier-driven retraining, motivate our theoretical study of this fundamental mechanism.

2 Verifier-Guided Synthetic Retraining in Linear Regression

In this section, we formalize our model of synthetic retraining with verification in the linear regression setting, where the objective is to estimate the coefficient vector $\theta^\star$.

Setup. Consider the linear model

$$y = x^\top \theta^\star + \xi,$$

where $\xi \sim \mathcal{N}(0, \sigma^2)$, $x \in \mathbb{R}^p$, and $\theta^\star \in \mathbb{R}^p$ is the unknown parameter of interest. We evaluate estimators using the mean squared error (MSE), i.e., $\mathbb{E}\|\hat{\theta} - \theta^\star\|^2$.

Suppose we have access to a verifier that encodes prior knowledge suggesting that the true parameter lies within a certain region. For analytical clarity, we model this knowledge as a spherical constraint:

$$B_r(\theta_c) := \{\, \theta \in \mathbb{R}^p : \|\theta - \theta_c\| \le r \,\},$$

with fixed (but unknown) center θ_c and radius r.

Verifier rule. The verifier does not reveal θ_c or r directly. Instead, it provides binary feedback indicating whether a given (real or synthetic) data point (x_i, y_i) is consistent with the sphere constraint. Specifically, the verifier outputs *Yes* if

$$|y_i - x_i^\top \theta_c| \le r\|x_i\| + \sigma_c, \tag{1}$$

and *No* otherwise. This rule is motivated by the expectation bound

$$\mathbb{E}\big[\,|y_i - x_i^\top \theta_c|\,\big] = \mathbb{E}\big[\,|x_i^\top(\theta^\star - \theta_c) + \xi_i|\,\big] \le r\|x_i\| + \mathbb{E}|\xi_i| = r\|x_i\| + \sqrt{\tfrac{2}{\pi}}\sigma.$$

Since the true σ might be unknown in practice, σ_c serves as an estimate of the true σ.

Motivation. We adopt this binary verifier model for both practical and theoretical reasons: (i) In practice, eliciting simple yes/no feedback is far less noisy and more cost-effective than asking verifiers to directly specify θ_c or r. Indeed, in many applications verifiers may not even know these quantities explicitly. (ii) This design mirrors the success of comparison-based feedback in reinforcement learning from human feedback (RLHF), where binary or relative judgments are easier for humans (or automated raters) to provide than absolute scores. Such binary responses have become a standard tool in preference alignment for large language models, where LLM raters and human evaluators provide pairwise or accept/reject judgments that effectively guide learning at scale. ((Ouyang et al., 2022; Wettig et al., 2024))

Thus, although simple, the binary verifier captures both the practical constraints of real-world feedback and the theoretical tractability needed for analysis, while serving as a natural mechanism to filter synthetic data during retraining.

Synthetic Retraining with Verifier Filtering We begin with a set of real data (X^0, Y^0), where $X^0 \in \mathbb{R}^{n_0 \times p}$ and $Y^0 \in \mathbb{R}^{n_0}$. The initial estimator $\hat{\theta}^0$ is obtained via Ordinary Least Squares (OLS):

$$\hat{\theta}^0 = ({X^0}^\top X^0)^{-1} {X^0}^\top Y^0. \tag{2}$$

We then proceed with iterative synthetic retraining, where each round follows a *generate–verify–retrain* scheme:

- **Generate:** Y^1 is generated by the following formula and X^1 is generated by the design detailed below:
$$Y^1 = X^1 \hat{\theta}^0 + \xi^1, \qquad \xi^1 \sim \mathcal{N}(0, \sigma^2 I).$$
- **Verify:** Each synthetic sample (x_i^1, y_i^1) is passed through the verifier condition equation 1. Only the verified subset is retained, denoted $(X^{1\prime}, Y^{1\prime})$.
- **Retrain:** A new OLS estimator is computed using only the verified data:
$$\hat{\theta}^1 = ({X^{1\prime}}^\top X^{1\prime})^{-1} {X^{1\prime}}^\top Y^{1\prime}. \tag{3}$$

For subsequent iterations $k \geq 1$, we repeat this procedure:

$$\hat{\theta}^k \overset{\text{generate}}{\longrightarrow} (X^{k+1}, Y^{k+1}) \overset{\text{verify}}{\longrightarrow} (X^{k+1\prime}, Y^{k+1\prime}) \overset{\text{retrain}}{\longrightarrow} \hat{\theta}^{k+1}. \tag{4}$$

Because learning proceeds through the conditional $Y^k \mid X^k$, synthetic retraining requires specifying the covariate design X^k; labels Y^k are then generated conditionally via the model under verifier constraints. In principle, one could construct X^k arbitrarily; however, to ensure mathematical clarity and keep the theorem tractable, we adopt a targeted design. We align the synthetic covariates with a fixed orthonormal set $\{v_1, \ldots, v_p\}$ and construct X^k in a block-structured form by repeating each $v_j^\top$ as rows:

$$X^k = (\ \underbrace{v_1, \ldots}_{\text{copies of } v_1},\ \underbrace{v_2, \ldots}_{\text{copies of } v_2},\ \ldots,\ \underbrace{v_p, \ldots}_{\text{copies of } v_p}\ )^\top.$$

After verifier filtering, each orthogonal direction v_j retains exactly n_k samples. This block design diagonalizes the transition operator $\hat{\theta}^k \mapsto \hat{\theta}^{k+1}$. By aligning synthetic samples with fixed orthogonal directions, we remove the rotational variability that arbitrary designs would introduce across iterations and decouple the dynamics along singular directions. In particular, choosing $\{v_j\}$ as the right singular vectors of the real data matrix X^0 yields the cleanest interpretation, making explicit how verifier bias, synthetic sample size, and noise variance interact. This choice clarifies both the short-term bias-variance tradeoff and the long-term convergence behavior, and we will adopt it in the following analysis.

This construction mirrors curating data along approximately orthogonal factors (e.g., topical axes like politics, economics, sports). It is not unique: alternatives (canonical basis, isotropic random directions) can yield similar qualitative conclusions, with potentially different constants or rates.

3 On the Near-Term Improvement under Synthetic Retraining

This section investigates the verifier's role in synthetic retraining: *does it help, when does it help, and why does it help?* We focus on one round and show that verifier-guided retraining can improve

performance under mild assumptions. The key mechanism is a verifier-induced bias–variance trade-off. We first present an error decomposition that isolates this trade-off, then provide a quantitative one-step bound that reveals how synthetic sample size, verifier bias/strength, determine improvement versus degradation. We conclude with design implications that inform the experiments in Section 5.

3.1 Source of Improvement: Bias–Variance Trade-off

To address the question of *when and why* verifier-guided synthetic retraining improves estimation, we analyze the mean squared error (MSE) of the one-step estimator $\hat{\theta}^1$ in estimating the true regression coefficient $\theta^\star$. The MSE admits the following decomposition:

$$\mathbb{E}\|\hat{\theta}^1 - \theta^\star\|^2 = \mathbb{E}_{\hat{\theta}^0}\Big[\operatorname{Tr}\big(\operatorname{Var}(\hat{\theta}^1 \mid \hat{\theta}^0)\big)\Big] + \mathbb{E}_{\hat{\theta}^0}\Big\|\mathbb{E}\Big[\hat{\theta}^1 \mid \hat{\theta}^0\Big] - \theta^\star\Big\|^2. \tag{5}$$

The first term in equation 5 is the **synthetic variance**: it captures additional estimation noise from the randomness in synthetic data generation. This variance decreases at rate $1/n_1$ with the synthetic sample size n_1, but is unaffected by the real sample size n_0. Hence, with abundant synthetic data, this term becomes negligible.

The second term is the **verification error**, which measures the deviation of the conditional mean estimator $\mathbb{E}(\hat{\theta}^1 \mid \hat{\theta}^0)$ from $\theta^\star$. This error depends both on the accuracy of the verifier (i.e., its potential bias) and the quality of the initial estimator $\hat{\theta}^0$, which improves with larger n_0.

To further disentangle the verification error, we decompose it as

$$\mathbb{E}_{\hat{\theta}^0}\Big\|\mathbb{E}\Big[\hat{\theta}^1 \mid \hat{\theta}^0\Big] - \theta^\star\Big\|^2 = \operatorname{Tr}\Big(\operatorname{Var}\Big(\mathbb{E}\Big[\hat{\theta}^1 \mid \hat{\theta}^0\Big]\Big)\Big) + \|\mathbb{E}[\hat{\theta}^1] - \theta^\star\|^2. \tag{6}$$

Here, the first term is the **verification variance**, reflecting variance reduction achieved by discarding inconsistent synthetic samples, while the second is the **verification bias**, capturing systematic deviation introduced by verifier bias.

Putting these together, the full decomposition is

$$\mathbb{E}\|\hat{\theta}^1 - \theta^\star\|^2 = \underbrace{\mathbb{E}_{\hat{\theta}^0}\Big[\operatorname{Tr}\big(\operatorname{Var}(\hat{\theta}^1 \mid \hat{\theta}^0)\big)\Big]}_{\text{Synthetic Variance}} + \underbrace{\operatorname{Tr}\Big(\operatorname{Var}\Big(\mathbb{E}\Big[\hat{\theta}^1 \mid \hat{\theta}^0\Big]\Big)\Big)}_{\text{Verification Variance}} + \underbrace{\|\mathbb{E}[\hat{\theta}^1] - \theta^\star\|^2}_{\text{Verification Bias}}. \tag{7}$$

This decomposition highlights the central trade-off: verifier filtering *reduces variance* but may *introduce bias*. Verified synthetic data leads to improvement precisely when the variance reduction outweighs the bias introduced. In particular, when the verifier is sufficiently accurate and the synthetic sample size n_1 is large, the MSE of $\hat{\theta}^1$ can be strictly smaller than that of the real-data estimator $\hat{\theta}^0$.

3.2 Characterizing Improvement in One-Round Retraining

The next theorem characterizes the MSE of one-step estimator $\hat{\theta}^1$ in 3. In particular, it shows that after one step of verifier-guided synthetic retraining, model can improve given that the bias of the verifier is small.

Theorem 3.1. *Suppose each eigenvalue of the design matrix X^0 is $\Omega(\sqrt{n_0})$.*[1] *Then there exist constants $m_{1,j}, m_{3,j} \in \mathbb{R}$ and $m_{2,j} \in (0,1)$ for $j = 1, \ldots, p$, depending only on $r, X^0, \theta^\star, \theta_c$, as well as constants $K, L > 0$ such that:*

$$\left|\frac{1}{\sigma^2}\mathbb{E}||\hat{\theta}^1 - \theta^\star||^2 - \sum_{j=1}^{p}\left(\underbrace{\frac{m_{2,j}}{n_1}}_{\textit{Synthetic Variance}} \underbrace{+ m_{1,j}^2 + \frac{m_{1,j}m_{3,j} + m_{2,j}^2}{\mu_j^2}}_{\textit{Verification Bias+Variance}}\right)\right| < K\left(\frac{1}{n_1 n_0^{1/3}} + \frac{1}{n_0^{3/2}}\right) \tag{8}$$

holds with probability at least $1 - p\exp\left(-Ln_0^{1/3}\right)$, where n_1 denotes the post-verification sample size.

[1] That is, each dimension is well-represented in the original data. This holds easily when, e.g., the feature data is drawn i.i.d. from a full-rank distribution.

Remark 1. *The constants $m_{1,j}, m_{2,j}, m_{3,j}$ (identified explicitly in Appendix B) are moments of a truncated Gaussian distribution induced by the verifier.*

- $m_{1,j}, m_{3,j}$*: capture the directional bias between θ_c and $\theta^\star$ along the j-th singular direction;*
- $m_{2,j}$*: quantifies the variance reduction along that direction, and always satisfies $m_{2,j} < 1$.*

In particular, if $\theta_c = \theta^\star$, then $m_{1,j} = m_{3,j} = 0$ for all $j = 1, \ldots, p$.

Theorem 3.1 reveals that improvement can be achieved after one step of verifier-guided synthetic retraining. For comparison, the MSE of the initial estimator $\hat{\theta}^0$ is

$$\frac{1}{\sigma^2}\,\mathbb{E}\|\hat{\theta}^0 - \theta^\star\|^2 = \sum_{j=1}^{p} \mu_j^{-2}. \tag{9}$$

When the verifier bias is small (so $m_{1,j}, m_{3,j} \approx 0$), the verification bias+variance term

$$m_{1,j}^2 + \tfrac{m_{1,j}m_{3,j} + m_{2,j}^2}{\mu_j^2}$$

is strictly smaller than the real-data variance μ_j^{-2}. Thus, whenever n_1 is sufficiently large, the bound in equation 8 improves upon the baseline equation 9. The gap between them quantifies the additional knowledge injected by the verifier through synthetic retraining.

This result highlights why verifier-guided retraining is practically useful: in regimes where real data are scarce but synthetic data can be generated cheaply, the verifier serves as a mechanism to filter and refine synthetic samples so that they effectively amplify limited real-world evidence. In practice, this suggests that retraining with a moderately accurate verifier can substantially reduce estimation error without requiring more real data, a setting that frequently arises in modern machine learning systems where data collection is costly but simulators or generative models are available.

As we will demonstrate empirically in Section 5, this bias–variance trade-off is not confined to the linear model. It also manifests in complex models such as VAEs, where the benefit of synthetic retraining is most pronounced during the early stages of training on the MNIST dataset.

4 ITERATIVE RETRAINING AS A MARKOV PROCESS, CONTRACTION AND CONVERGENCE TO THE KNOWLEDGE CENTER

In the previous subsection, we showed that one-step verifier-guided retraining can improve estimation accuracy through bias-variance trade-offs. This raises a natural question:

Q: If a single round of verifier-filtered retraining improves performance, can such improvement be sustained over multiple rounds, and what is the eventual outcome?

In this subsection, we address this question. We want to understand the nature of the long-term dynamics of iterative verifier-guided retraining though studying the linear regression model. Prior to presenting our main results, we clarify the terminology frequently employed in the literature on model collapse, focusing on its meaning in our linear regression setting.

- **Model Degradation/Collapse:** $\limsup_{k\to\infty} \mathbb{E}||\hat{\theta}^k - \theta^\star||^2 > \mathbb{E}||\hat{\theta}^1 - \theta^\star||^2$.
- **Model Improvement:** $\limsup_{k\to\infty} \mathbb{E}||\hat{\theta}^k - \theta^\star||^2 < \mathbb{E}||\hat{\theta}^1 - \theta^\star||^2$.

Our key finding is that both behaviors can occur in long-term iterative retraining. The outcome depends critically on three factors: the growth rate of synthetic data, the verifier's bias, and the verifier's strength (i.e., its ability to reduce variance). Over time, iterative retraining injects increasingly more verifier knowledge into the estimator, while the contribution from the original data gradually decays. As a result, the verifier and the generative model family eventually dominate the limit behavior, driving the estimator $\hat{\theta}^k$ toward the verifier's knowledge center θ_c.

This dynamic gives rise to three distinct phases of long-term behavior:

- **(1) Unbiased verifier:** If the verifier is unbiased (i.e., $\theta_c = \theta^\star$), iterative retraining yields continuous improvement and the estimator converges to the true parameter.
- **(2) Mildly biased verifier:** With small bias, iterative retraining can improve performance in the short term by reducing variance, but performance eventually plateaus or deteriorates as verifier bias accumulates.
- **(3) Strongly biased verifier:** With large bias, iterative retraining leads to degradation and may even cause collapse in the limit.

Among these, case (2) is particularly relevant in practice. It highlights a cautionary message: while synthetic retraining can initially boost accuracy, it cannot guarantee sustained improvement unless the verifier is highly reliable. Since ensuring a perfectly unbiased verifier is unrealistic, the influence of the original data will eventually vanish, leaving the verifier (and the chosen generative model family) to dictate the long-term outcome.

Formally, the following theorem characterizes the long-term behavior of the estimator $\hat{\theta}^k$ in linear regression under iterative verifier-guided retraining.

Theorem 4.1. *There exist a synthetic retraining process and some constant* $0 < \rho < 1$ *such that:*

$$\mathbb{E}\|\hat{\theta}^k - \theta_c\|^2 \leq \rho^{2k}\mathbb{E}\|\hat{\theta}^0 - \theta_c\|^2 + p\sigma^2 \sum_{j=0}^{k-1} \frac{\rho^{2(k-j)-1}}{n_j}. \tag{10}$$

In particular, if $\lim_{k\to\infty} n_k = \infty$*, then* $\lim_{k\to\infty} \mathbb{E}\|\hat{\theta}^k - \theta_c\|^2 = 0$.

The proof of Theorem 4.1 is provided in Appendix B, where concentration bounds and supermartingale inequalities are used to establish convergence. Here we focus on the main intuition and highlight the key novelty of our analysis.

The central observation is that the iterative retraining procedure equation 4 induces a *Markov process*: the next state $\hat{\theta}^{k+1}$ depends only on the current state $\hat{\theta}^k$. Formally, the update can be expressed as

$$\hat{\theta}^{k+1} = T(\hat{\theta}^k) + \eta_{k+1}, \tag{11}$$

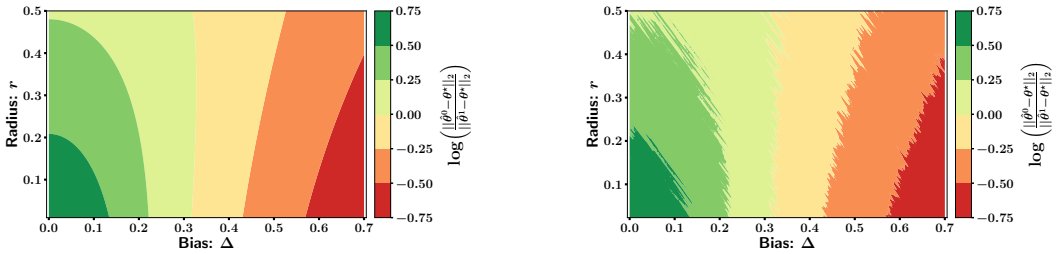
where $T(\cdot)$ is a deterministic mapping determined by verifier filtering, and η_{k+1} is a sub-Gaussian noise term due to the randomness of synthetic samples at iteration $k+1$. Crucially, we show that $T(\cdot)$ is a *contraction mapping*, and that the variance of the noise decays at the rate $\mathrm{Var}(\eta_{k+1}) \asymp 1/n_{k+1}$.

This perspective allows us to view equation 11 as a discretized stochastic differential equation (SDE). As $n_k \to \infty$, the noise term vanishes and the dynamics are dominated by the deterministic contraction $T(\hat{\theta}^k)$, which drives the recursion toward its fixed point—the verifier's knowledge center θ_c. The presence of the verifier is therefore *essential*: it is precisely what transforms the update rule into a contraction, guaranteeing convergence.

By contrast, in prior work on model collapse without a verifier (e.g., Gerstgrasser et al. (2024); Xu et al. (2025)), the update reduces to the identity mapping. In that case, increasing the synthetic sample size can suppress noise accumulation and ensure bounded error (i.e., $\mathbb{E}\|\hat{\theta}^k - \theta^\star\|^2 < \infty$), but there is no contraction and hence no convergence or sustained improvement. The critical difference between $T(\cdot)$ and the identity is exactly the knowledge extracted from the verifier through synthetic data. Our analysis is the first to formally show that the verifier fundamentally alters the long-term dynamics: it continuously injects knowledge, iteration by iteration, so that the estimator moves closer to θ_c over time.

This contribution also clarifies a common misconception: even with a perfect verifier ($\theta_c = \theta^\star$) and infinitely many synthetic samples in one iteration, convergence cannot occur in a single step. As shown in Theorem 3.1, while infinite samples remove the synthetic variance term, the verification bias+variance term persists. Thus, convergence requires the *iterative* action of the verifier, which gradually aligns the estimator with the truth.

We observe the same phenomenon in our CVAE experiments on MNIST (Section 5). During early iterations, enlarging the synthetic sample size substantially improves the model; however, beyond a threshold, further increases bring diminishing returns.



(a) Loss Landscape Predicted by Theory

(b) Empirical Loss Landscape from Simulations

Figure 1: Error changes of the one-step retraining estimator $\hat{\theta}^1$ versus estimator $\hat{\theta}^0$ only using original real data, measured by $\log(\frac{||\hat{\theta}^0-\theta^\star||}{||\hat{\theta}^1-\theta^\star||})$: theory's prediction (left) and empirical comparisons (right).

5 Experiments

In this section, we evaluate our method in two settings: *linear regression simulation*, which mirrors the theoretical assumptions, and a *Conditional Variational Autoencoder (CVAE) on MNIST*, which demonstrates practical behavior under iterative retraining and filtering. Appendix D.6 further includes a large-scale news summarization experiment using `SmolLM2-135M`(Allal et al., 2025). In all cases, the results closely align with our theoretical predictions.

5.1 Simulation: Linear Regression

Setting. We consider the linear model $y = x^\top \theta^\star + \xi$, with $\xi \sim \mathcal{N}(0,1)$, $\theta^\star \in \mathbb{R}^p$, and $x \in \mathbb{R}^p$. An initial OLS estimator is fitted on a small real dataset (X^0, Y^0), after which we conduct K iterative rounds of synthetic top-up aligned with the right singular vectors of X^0.

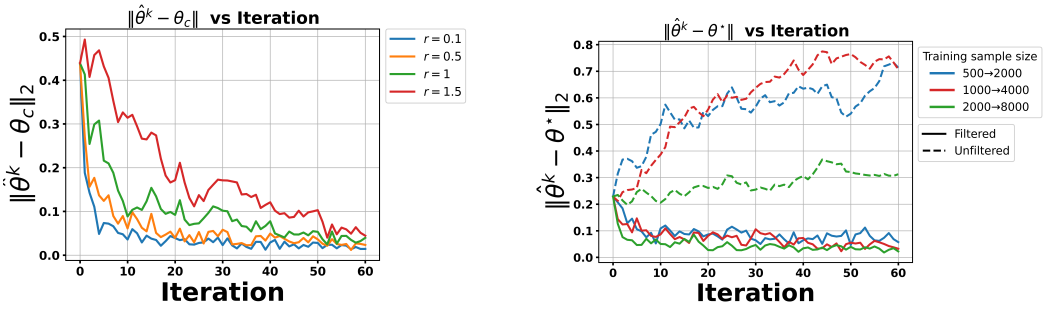
One-step Synthetic Retraining. Figure 1a shows the error reduction predicted by Theorem 3.1 align closely with the empirical results in Figure 1b, validating the sharpness of our theoretical bounds in Theorem 3.1. In this experiment, we set $\theta^\star = \mathbf{1}_8$ and define the verifier's belief center as $\theta_c = \theta^\star + \Delta \cdot \mathbf{1}$, where Δ controls the verifier's bias relative to the truth. The verification radius r determines how strictly synthetic samples are filtered: smaller r enforces tighter acceptance around θ_c, while larger r admits looser acceptance. Using 100 real samples and 200 verified synthetic samples per singular direction, we find that verifier-guided retraining outperforms the real-only baseline when verifier bias is small (green region), whereas excessive bias leads to degradation (red region). This experiment empirically confirms the short-term bias–variance trade-off formalized in Theorem 3.1.

Iterative Synthetic Retraining. Similarly, Figure 2a confirms Theorem 4.1, because it shows that under a biased verifier, the retraining estimator converges to the verifier's 'knowledge center" θ_c. In this experiment, the sample size increases linearly from 100 to 5500 over 60 rounds, with $\theta^\star = \mathbf{1}_8$ and $\theta_c = \theta^\star + 0.1 \cdot \mathbf{1}$. The results also show that convergence is faster with a smaller verification radius. In Figure 2b, we repeat the experiment with an unbiased verifier ($\theta_c = \theta^\star = \mathbf{1}_8$). In this case, verifier-guided retraining achieves consistently lower error than retraining without verification. These findings provide empirical support for our long-term analysis in Theorem 4.1, demonstrating how the contraction effect of the verifier yields convergence in practice. Appendix D.1 reports results using randomly designed synthetic data, and Appendix D.2 provides experiments with different verifier shapes. These results show that our conclusions are robust to the synthetic data design and the verifier shape.

5.2 Conditional Variational Autoencoders (CVAE) on MNIST

We also conduct experiments on real-world image generation to demonstrate the applicability of our theory beyond linear regression.

Setting. To make the bias–variance trade-off and verifier-injection effects clearly observable, we initialize the CVAE with only 500 real MNIST images, creating a challenging low-resource scenario.



(a) Verifier with bias

(b) Verifier without bias

Figure 2: Iterative synthetic retraining with and without bias.

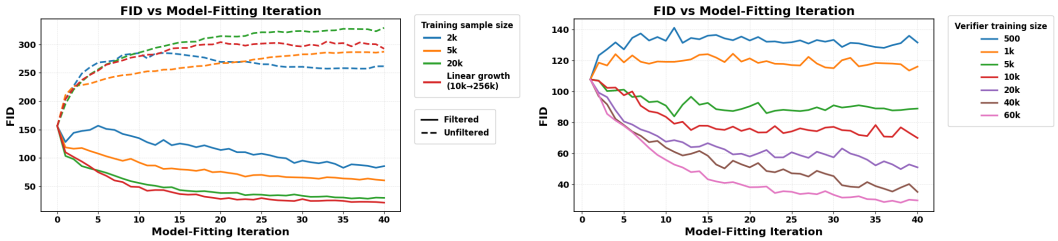
We use a small initial real-data sample because this is the regime where our theory predicts the strongest benefits from verifier-guided retraining: an undertrained generator can leverage the verifier's external information. When initialized on the full 60k MNIST images, the generator leaves little room for such gains, and the verifier primarily prevents collapse. Additional results for varying initial sample sizes are presented in Appendix D.5. A discriminator, trained on varying amounts of real data together with an equal number of synthetic samples, serves as the verifier. It assigns each synthetic sample a probability of being real, and we retain the top 10% per digit. This 10% threshold is motivated by a one-step synthetic retraining study: across synthetic sizes and filtering thresholds, retaining the top 10% yielded the best balance between quality and diversity. Overly strict filtering produces high-quality but low-diversity samples, while overly loose filtering yields diverse but lower-quality samples.

The number of retained samples n_1 follows two schedules: (i) a fixed sample size, or (ii) a linear growth schedule. We then retrain the CVAE on the retained synthetic data and repeat this procedure until performance stabilizes. Empirically, beyond 40 iterations the Fréchet Inception Distance (FID) no longer improves, so we report results up to 40 rounds as a conservative steady-state horizon. Generative quality is measured by the FID between generated data and real data. The MNIST-specific FID is reported in Appendix D.3, and the results are consistent with those presented in this section. For more details on model architecture, training, and evaluation, see Appendix C.

Results. Because our verifier—implemented as a discriminator—provides feedback biased toward perceptual realism rather than likelihood calibration, we report **FID** as the primary metric in the main text and defer likelihood-based reconstruction metrics (ELBO) to Appendix C. Figure 3a reports FID across retraining iterations. With a strong verifier (trained on the full real dataset and an equal amount of synthetic data), we observe rapid FID improvement within the first 15 rounds, even under small fixed-size schedules (green (20K) and orange (5K) curves). Afterward, the improvement slows and eventually plateaus. In contrast, synthetic retraining without a verifier leads to severe degradation. This behavior closely mirrors our theory: (i) early gains arise from the short-term bias-variance trade-off (Theorem 3.1), and (ii) long-term stability is predicted by the contraction effect of verifier filtering (Theorem 4.1).

Figure 4 provides qualitative evidence. Compared to the baseline CVAE trained on 500 real samples, the model retrained for 40 rounds with verified synthetic data produces significantly sharper and more realistic images. By contrast, the model retrained without verification deteriorates after 40 rounds, consistent with model collapse. The choice of 40 rounds corresponds to the point at which loss and FID stabilize, so further retraining brings no additional benefit. Additional qualitative results are provided in Appendix D.4.

The plateau highlights verifier limitations: because the verifier is relatively simple, it may overemphasize certain styles or patterns in synthetic data that are easier to distinguish from real data, thereby introducing bias. For reference, a CVAE trained on all 60K real samples achieves an FID of 17.56 and reconstruction error of 71.52, while the best synthetic model (red curve) after 40 verified retraining iterations reaches 21.17 and 91.21, respectively.



(a) Filtering and training sample size.

(b) Verifier quality.

Figure 3: FID results across retraining rounds. (a) Effect of filtering and retained sample size. (b) Effect of verifier quality, varied by training data size. Together, the plots highlight how both sample selection and verifier strength shape generative performance.

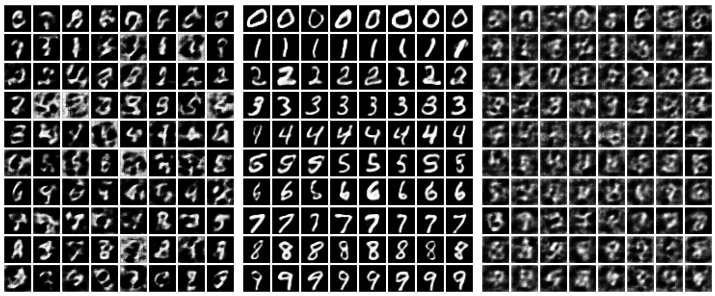


Figure 4: Samples generated by the CVAE at different stages: **Left:** model trained in the first round on 500 real images, **Middle:** model after 40 rounds with filtering under a linear sample growth schedule (10k→256k), **Right:** model after 40 rounds without filtering under the same linear schedule.

Finally, Figure 3b examines how verifier quality affects retraining. Here, the CVAE is trained with 20K synthetic samples per round. As expected, stronger verifiers (trained on more real data) yield larger FID improvements, whereas weaker verifiers cause the FID curve to plateau early and can even degrade performance.

We also report test ELBO in Appendix C. Although ELBO is harder to improve than FID under our current verifier design, the same theoretical predictions persist: (i) the verifier prevents collapse, (ii) early gains reflect the bias–variance trade-off, and (iii) performance eventually plateaus and can even reverse after $\sim$10 iterations.

6 Discussion

Our study provides a theoretical and empirical characterization of verifier-guided synthetic retraining. We show that the process yields *short-term gains* by reducing variance through verifier filtering, but in the *long run* the estimator converges to the verifier's knowledge center. This explains both the promise and the risk of such methods: a high-quality verifier can inject reliable external knowledge, while a biased verifier inevitably steers the model away from the truth. Viewed through the lens of *information elicitation*, our framework formalizes how external signals are incorporated recursively into training and why the outcome reflects the verifier's information.

At the same time, our framework has limitations. We have focused on linear regression as the analytical testbed, and although extensions to generative models such as VAEs validate the theory qualitatively, further generalization is needed. Future work includes developing sharper bounds for nonlinear models, exploring alternative synthetic design strategies beyond block orthogonalization, and studying verifier dynamics in large-scale language and vision models.

REFERENCES

Sina Alemohammad, Josue Casco-Rodriguez, Lorenzo Luzi, Ahmed Imtiaz Humayun, Hossein Babaei, Daniel LeJeune, Ali Siahkoohi, and Richard G Baraniuk. Self-consuming generative models go mad. *arXiv preprint arXiv:2307.01850*, 4:14, 2023.

Loubna Ben Allal, Anton Lozhkov, Elie Bakouch, Gabriel Martín Blázquez, Guilherme Penedo, Lewis Tunstall, Andrés Marafioti, Hynek Kydlíček, Agustín Piqueres Lajarín, Vaibhav Srivastav, Joshua Lochner, Caleb Fahlgren, Xuan-Son Nguyen, Clémentine Fourrier, Ben Burtenshaw, Hugo Larcher, Haojun Zhao, Cyril Zakka, Mathieu Morlon, Colin Raffel, Leandro von Werra, and Thomas Wolf. Smollm2: When smol goes big – data-centric training of a small language model, 2025. URL https://arxiv.org/abs/2502.02737.

Kareem Amin, Sara Babakniya, Alex Bie, Weiwei Kong, Umar Syed, and Sergei Vassilvitskii. Escaping collapse: The strength of weak data for large language model training. *arXiv preprint arXiv:2502.08924*, 2025.

Zahra Azizi, Chaoyi Zheng, Lucy Mosquera, Louise Pilote, and Khaled El Emam. Can synthetic data be a proxy for real clinical trial data? a validation study. *BMJ open*, 11(4):e043497, 2021.

Derrick Adrian Chan and Siphesihle Philezwini Sithungu. Evaluating the suitability of inception score and fréchet inception distance as metrics for quality and diversity in image generation. In *Proceedings of the 2024 7th International Conference on Computational Intelligence and Intelligent Systems*, pp. 79–85, 2024.

Bin Dai and David Wipf. Diagnosing and enhancing vae models. In *International Conference on Learning Representations*, 2019.

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

Carl Doersch and Andrew Zisserman. Sim2real transfer learning for 3d human pose estimation: motion to the rescue. *Advances in Neural Information Processing Systems*, 32, 2019.

Elvis Dohmatob, Yunzhen Feng, and Julia Kempe. Model collapse demystified: The case of regression. *arXiv preprint arXiv:2402.07712*, 2024a.

Elvis Dohmatob, Yunzhen Feng, Arjun Subramonian, and Julia Kempe. Strong model collapse. *arXiv preprint arXiv:2410.04840*, 2024b.

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*, 2024c.

Yunzhen Feng, Elvis Dohmatob, Pu Yang, Francois Charton, and Julia Kempe. Beyond model collapse: Scaling up with synthesized data requires reinforcement. In *ICML 2024 Workshop on Theoretical Foundations of Foundation Models*, 2024.

Damien Ferbach, Quentin Bertrand, Avishek Joey Bose, and Gauthier Gidel. Self-consuming generative models with curated data provably optimize human preferences. *arXiv preprint arXiv:2407.09499*, 2024.

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.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638, 2025.

Patrick Haluptzok, Matthew Bowers, and Adam Tauman Kalai. Language models can teach themselves to program better. *arXiv preprint arXiv:2207.14502*, 2022.

Shadi Iskander, Nachshon Cohen, Zohar Karnin, Ori Shapira, and Sofia Tolmach. Quality matters: Evaluating synthetic data for tool-using llms. *arXiv preprint arXiv:2409.16341*, 2024.

Andrea Lampis, Eugenio Lomurno, and Matteo Matteucci. Bridging the gap: Enhancing the utility of synthetic data via post-processing techniques. *arXiv preprint arXiv:2305.10118*, 2023.

Mikhail Leontev, Alexander Mikheev, Kirill Sviatov, and Sergey Sukhov. Quality metrics of variational autoencoders. In *2020 International Conference on Information Technology and Nanotechnology (ITNT)*, pp. 1–5. IEEE, 2020.

Zhaoshan Liu, Qiujie Lv, Yifan Li, Ziduo Yang, and Lei Shen. Medaugment: Universal automatic data augmentation plug-in for medical image analysis. *arXiv preprint arXiv:2306.17466*, 2023.

Alisia Lupidi, Carlos Gemmell, Nicola Cancedda, Jane Dwivedi-Yu, Jason Weston, Jakob Foerster, Roberta Raileanu, and Maria Lomeli. Source2synth: Synthetic data generation and curation grounded in real data sources. *arXiv preprint arXiv:2409.08239*, 2024.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. Don't give me the details, just the summary! Topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Brussels, Belgium, 2018.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

Parth Patwa, Simone Filice, Zhiyu Chen, Giuseppe Castellucci, Oleg Rokhlenko, and Shervin Malmasi. Enhancing low-resource llms classification with peft and synthetic data. *arXiv preprint arXiv:2404.02422*, 2024.

Vamsi K Potluru, Daniel Borrajo, Andrea Coletta, Niccolò Dalmasso, Yousef El-Laham, Elizabeth Fons, Mohsen Ghassemi, Sriram Gopalakrishnan, Vikesh Gosai, Eleonora Kreačić, et al. Synthetic data applications in finance. *arXiv preprint arXiv:2401.00081*, 2023.

Gabriele Santangelo, Giovanna Nicora, Riccardo Bellazzi, and Arianna Dagliati. How good is your synthetic data? synthro, a dashboard to evaluate and benchmark synthetic tabular data. *BMC Medical Informatics and Decision Making*, 25(1):89, 2025.

Ashish Shrivastava, Tomas Pfister, Oncel Tuzel, Joshua Susskind, Wenda Wang, and Russell Webb. Learning from simulated and unsupervised images through adversarial training. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2107–2116, 2017.

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.

Jonathan Tremblay, Aayush Prakash, David Acuna, Mark Brophy, Varun Jampani, Cem Anil, Thang To, Eric Cameracci, Shaad Boochoon, and Stan Birchfield. Training deep networks with synthetic data: Bridging the reality gap by domain randomization. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pp. 969–977, 2018.

Zeao Tu, Xiangdi Meng, Yu He, Zihan Yao, Tianyu Qi, Jun Liu, and Ming Li. Resofilter: Rine-grained synthetic data filtering for large language models through data-parameter resonance analysis. *arXiv preprint arXiv:2412.14809*, 2024.

Xiukun Wei and Xueru Zhang. Self-consuming generative models with adversarially curated data. *arXiv preprint arXiv:2505.09768*, 2025.

Alexander Wettig, Aatmik Gupta, Saumya Malik, and Danqi Chen. Qurating: Selecting high-quality data for training language models. *arXiv preprint arXiv:2402.09739*, 2024.

Erroll Wood, Tadas Baltrušaitis, Charlie Hewitt, Sebastian Dziadzio, Thomas J Cashman, and Jamie Shotton. Fake it till you make it: face analysis in the wild using synthetic data alone. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 3681–3691, 2021.

Fang Wu and Yejin Choi. The invisible leash: Why rlvr may not escape its origin. In *2nd AI for Math Workshop@ ICML 2025*.

Shirong Xu, Hengzhi He, and Guang Cheng. A probabilistic perspective on model collapse. *arXiv preprint arXiv:2505.13947*, 2025.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. Star: Bootstrapping reasoning with reasoning. *Advances in Neural Information Processing Systems*, 35:15476–15488, 2022.

Jinghui Zhang, Dandan Qiao, Mochen Yang, and Qiang Wei. Regurgitative training: The value of real data in training large language models. *arXiv preprint arXiv:2407.12835*, 2024.

Xuekai Zhu, Daixuan Cheng, Hengli Li, Kaiyan Zhang, Ermo Hua, Xingtai Lv, Ning Ding, Zhouhan Lin, Zilong Zheng, and Bowen Zhou. How to synthesize text data without model collapse? *arXiv preprint arXiv:2412.14689*, 2024.

APPENDIX OVERVIEW

This appendix contains: Appendix A (1-D Gaussian toolkit), Appendix B (reduction and full proof for linear regression), Appendix C (additional details on CVAE experiments), Appendix D (additional simulations and experiments), Appendix E (use of large language models)

A ONE-DIMENSIONAL GAUSSIAN TOOLKIT

In this section, we provide a toolkit for analyzing the one-dimensional Gaussian mean estimation problem with verifier-filtered synthetic data. This toolkit serves as the foundation for our analysis of the linear regression models. We will establish several key lemmas and theorems that characterize the MSE of the mean estimator under the one-dimensional Gaussian model. These results will be instrumental in proving Theorem 3.1 and Theorem 4.1 in Appendix B.

A.1 SETUP AND NOTATIONS

We consider the one-dimensional mean estimation problem where the real data $X_1^0, \ldots, X_{n_0}^0$ are independently and identically distributed (i.i.d.) from a Gaussian distribution:

$$X_1^0, \ldots, X_{n_0}^0 \overset{\text{i.i.d.}}{\sim} \mathcal{N}(\mu, \sigma^2),$$

with known variance σ^2.

In our setting, a verifier exists and encodes external knowledge that the true mean lies in an interval $[a, b]$ (i.e.$\mu \in [a, b]$). Therefore, $\bar{X}^0 = \frac{X_1 + \cdots + X_{n_0}}{n_0}$ is the empirical mean of real data, which minimizes MSE if *no extra* information is supplied. We are interested in whether data verification could effectively inject new information and improve over $\bar{X}^0$. Consider the following synthetic data generation and filtering procedure:

- Generate n_1 synthetic data $X_1^1, \ldots, X_{n_1}^1 \overset{\text{i.i.d.}}{\sim} \mathcal{N}(\bar{X}^0, \sigma^2)$.
- Retain $X_i^0 \in [a, b]$ as $X_1'^1, \ldots, X_{n_1'}'^1$, and estimate μ using $\bar{X}^1 = \frac{1}{n_1'} \sum_{i=1}^{n_1'} X_i'^1$.

We will compare the estimator $\bar{X}^1$ with $\bar{X}^0$ and formally characterize when data verification enhances or degrades model performance - i.e., when $\mathbb{E}(\bar{X}^1 - \mu)^2 < \mathbb{E}(\bar{X}^0 - \mu)^2$ or not. Our key finding is that $\bar{X}^1$ introduces the core bias-variance trade-off that underpins model improvement or degradation. We will characterize the MSE of $\bar{X}^1$ which reveals how key quantities such as the real and synthetic sample size, the verifier's bias and variance will decide performance of the filtering strategy. These insights provide intuition for extending verifier-guided re-training to more complex settings.

We first review some notation and key results for the truncated normal distribution, which will be used in the subsequent sections. Consider a one-dimensional normal distribution $X \sim \mathcal{N}(\mu, \sigma^2)$ and let X' be its truncated version restricted to the interval $[a, b]$. The distribution of X' is the called the *truncated normal distribution*, denoted as $X' \sim \mathcal{N}(x|\mu, \sigma^2) \cdot \mathbb{1}_{\{a<x<b\}}$. The mean and variance of the truncated normal distribution X' are given analytically:

$$\begin{aligned}
\mathbb{E}[X'|\mu] &= \mu - \sigma \frac{\phi(\frac{b-\mu}{\sigma}) - \phi(\frac{a-\mu}{\sigma})}{\Phi(\frac{b-\mu}{\sigma}) - \Phi(\frac{a-\mu}{\sigma})} := \mu + \sigma m_1(\frac{a-\mu}{\sigma}, \frac{b-\mu}{\sigma}) \\
\mathrm{Var}(X'|\mu) &= \sigma^2 \left[1 - \frac{\frac{b-\mu}{\sigma}\phi(\frac{b-\mu}{\sigma}) - \frac{a-\mu}{\sigma}\phi(\frac{a-\mu}{\sigma})}{\Phi(\frac{b-\mu}{\sigma}) - \Phi(\frac{a-\mu}{\sigma})} - \left(\frac{\phi(\frac{b-\mu}{\sigma}) - \phi(\frac{a-\mu}{\sigma})}{\Phi(\frac{b-\mu}{\sigma}) - \Phi(\frac{a-\mu}{\sigma})} \right)^2 \right] \\
&:= \sigma^2 m_2(\frac{a-\mu}{\sigma}, \frac{b-\mu}{\sigma}) \qquad (12)
\end{aligned}$$

where $\phi(x)$ and $\Phi(x)$ denote the standard normal density and cumulative distribution functions, respectively. Standardizing X via $Z := \frac{X-\mu}{\sigma}$ and setting

$$\alpha = \frac{a-\mu}{\sigma}, \quad \beta = \frac{b-\mu}{\sigma}, \qquad (13)$$

the expression in equation 12 become:

$$\begin{aligned}\mathbb{E}[Z'] &= m_1(\alpha,\beta)\\ \mathrm{Var}(Z') &= m_2(\alpha,\beta)\end{aligned} \tag{14}$$

where $Z' \sim \mathcal{N}(x|0,1)\cdot\mathbb{1}_{\{\alpha<x<\beta\}}$ is the standardized truncated normal distribution. For convenience, we write $\mathcal{N}_{trunc}(\alpha,\beta) := \mathcal{N}(x|0,1)\cdot\mathbb{1}_{\{\alpha<x<\beta\}}$. Thus, m_1 and m_2 correspond to the first and second central moments of the standardized truncated normal distribution. In addition, we also define the third central moment of the standardized truncated normal distribution:

$$\begin{aligned}m_3(\alpha,\beta) &:= \mathbb{E}(Z'-\mathbb{E}Z')^3\\ &= -\frac{(\beta^2-1)\phi(\beta)-(\alpha^2-1)\phi(\alpha)}{(\Phi(\beta)-\Phi(\alpha))} - \frac{3(\phi(\beta)-\phi(\alpha))(\beta\phi(\beta)-\alpha\phi(\alpha))}{(\Phi(\beta)-\Phi(\alpha))^2}\\ &\quad - \frac{2(\phi(\beta)-\phi(\alpha))^3}{(\Phi(\beta)-\Phi(\alpha))^3}.\end{aligned} \tag{15}$$

In particular, $0 < m_2(\alpha,\beta) < 1$ for any $\alpha < \beta$ and $m_1(\alpha,\beta) = m_3(\alpha,\beta) = 0$ if $\alpha+\beta = 0$.

A.2 Characterization of $\mathbb{E}(\bar{X}^1-\mu)^2$, Bias-Variance Trade-off, and Model Improvement

Theorem A.1. *Assume that $n_1 > n_0 \geq 100$. Then there exists constant K, depending only on α and β, such that*

$$\begin{aligned}&\left|\frac{1}{\sigma^2}\mathbb{E}(\bar{X}^1-\mu)^2 - \underbrace{\frac{m_2(\alpha,\beta)}{n_1}}_{\textit{Synthetic Variance}} - \underbrace{\left(m_1^2(\alpha,\beta)+\frac{m_2^2(\alpha,\beta)+m_3(\alpha,\beta)m_1(\alpha,\beta)}{n_0}\right)}_{\textit{Verification Bias+Variance}}\right|\\ &< K\left(\frac{1}{n_1 n_0^{1/3}}+\frac{1}{n_0^{3/2}}\right)\end{aligned} \tag{16}$$

holds with probability at least $1-\exp\left(-\frac{1}{2}n_0^{1/3}\right)$.

Proof of Theorem A.1. It will be convenient to reparameterize the sample mean estimators by centering them around the true mean. Specifically, we define the residuals:

$$\epsilon_1 := \frac{\bar{X}^0-\mu}{\sigma}, \quad \epsilon_1 \sim \mathcal{N}(0,\frac{1}{n_0}). \tag{17}$$

Note that $\bar{X}^1$ is the mean of n_1 i.i.d. samples from the truncated normal distribution $\mathcal{N}(x|\bar{X}^0,\sigma^2)\cdot\mathbb{1}_{\{a<x<b\}}$. The MSE of $\bar{X}^1$ can be decomposed as follows:

$$\begin{aligned}\mathbb{E}[(\bar{X}^1-\mu)^2] &= \mathbb{E}_{\bar{X}^0}\,\mathbb{E}_{\bar{X}^1|\bar{X}^0}\left[(\bar{X}^1-\mu)^2\right]\\ &= \mathbb{E}_{\bar{X}^0}\left[\mathrm{Var}(\bar{X}^1\mid,\bar{X}^0)+\left(\mathbb{E}[\bar{X}^1\mid\bar{X}^0]-\mu\right)^2\right]\\ &= \sigma^2\mathbb{E}_{\bar{X}^0}\left[\frac{m_2(\alpha-\epsilon_1,\beta-\epsilon_1)}{n_1}\right]+\mathbb{E}_{\bar{X}^0}\left[(\bar{X}^0-\mu-\sigma m_1(\alpha-\epsilon_1,\beta-\epsilon_1))^2\right]\\ &= \frac{\sigma^2}{n_1}\mathbb{E}_{\epsilon_1}\left[m_2(\alpha-\epsilon_1,\beta-\epsilon_1)\right]+\sigma^2\,\mathbb{E}_{\epsilon_1}\left[\left(m_1(\alpha-\epsilon_1,\beta-\epsilon_1)+\epsilon_1\right)^2\right]\end{aligned} \tag{18}$$

For the first term in 18, we consider the event $E_1 := \left\{|\epsilon_1| < n_0^{-1/3}\right\}$, the function $m_2(\cdot,\cdot)$ is Lipschitz continuous in a neighborhood of (α,β), so we have

$$|m_2(\alpha-\epsilon_1,\beta-\epsilon_1)-m_2(\alpha,\beta)| = |\epsilon_1|\cdot\left|m_2^{(1)}(\alpha-\xi,\beta-\xi)\right| < \frac{M_1}{n_0^{1/3}}, \tag{19}$$

for some $\xi \in (0, \epsilon_1)$, where we define

$$M_1 := \sup_{|\xi| < \frac{1}{100^{\frac{1}{3}}}} \left| m_2^{(1)}(\alpha - \xi, \beta - \xi) \right|,$$

and M_1 is a constant independent of n_0 as long as $n_0 \geq 100$. Event E_1 hold with high probability:

$$\mathbb{P}\left(|\epsilon_1| < n_0^{-1/3}\right) > 1 - \frac{\exp\left(-\frac{n_0^{1/3}}{2}\right)}{\sqrt{\pi/2} \cdot n_0^{1/6}} > 1 - \frac{\exp\left(-\frac{n_0^{1/3}}{2}\right)}{\sqrt{\pi/2} \cdot 100^{1/6}} > 1 - \exp\left(-\frac{n_0^{1/3}}{2}\right).$$

Then we consider then second term in 18. The Taylor expansion of the function

$$m_1(\epsilon_1) := m_1(\alpha - \epsilon_1, \beta - \epsilon_1)$$

up to the third-order terms is:

$$m_1(\epsilon_1) = m_1(\alpha, \beta) - [1 - m_2(\alpha, \beta)]\,\epsilon_1 + \frac{1}{2} m_3(\alpha, \beta)\epsilon_1^2 + \frac{1}{6} m_1^{(3)}(\xi)\epsilon_1^3, \quad \text{for some } \xi \in (0, \epsilon_1), \tag{20}$$

where $m_1^{(3)}(\xi)$ denotes the third derivative of m_1 evaluated at some point between 0 and ϵ_1. Then we can get

$$\begin{aligned}
\mathbb{E}_{\epsilon_1}\left[(m_1(\alpha - \epsilon_1, \beta - \epsilon_1) + \epsilon_1)^2\right] &= \mathbb{E}\left(m_1(\alpha, \beta) + m_2(\alpha, \beta)\epsilon_1 + \frac{1}{2} m_3(\alpha, \beta)\epsilon_1^2 + \frac{1}{6} m_1^{(3)}(\xi)\epsilon_1^3\right)^2 \\
&= m_1^2(\alpha, \beta) + \frac{m_2^2(\alpha, \beta) + m_1(\alpha, \beta) m_3(\alpha, \beta)}{n_0} + \frac{3 m_3^2(\alpha, \beta)}{4 n_0^2} \\
&\quad + \mathbb{E}\left(m_1(\alpha, \beta) + m_2(\alpha, \beta)\epsilon_1 + \frac{1}{2} m_3(\alpha, \beta)\epsilon_1^2\right) \frac{m_1^{(3)}(\xi)}{3} \epsilon_1^3 \\
&\quad + \mathbb{E}\left(\frac{{m_1^{(3)}}^2(\xi)}{36} \epsilon_1^6\right).
\end{aligned} \tag{21}$$

First, using the fact that there exists constant M that only depends on α and β, such that $|m_1^{(3)}(x)| < M$ for any x, we have:

$$\begin{aligned}
&\left| \mathbb{E}\left[\left(m_1(\alpha, \beta) + m_2(\alpha, \beta)\epsilon_1 + \frac{1}{2} m_3(\alpha, \beta)\epsilon_1^2\right) \frac{m_1^{(3)}(\xi)}{3} \epsilon_1^3\right] \right| \\
&\leq \mathbb{E}\left[\left(|m_1(\alpha, \beta)| + m_2(\alpha, \beta)|\epsilon_1| + \frac{1}{2}|m_3(\alpha, \beta)|\epsilon_1^2\right) \cdot \frac{M}{3}|\epsilon_1|^3\right] \\
&= \mathbb{E}\left[\frac{M}{3}|m_1(\alpha, \beta)||\epsilon_1|^3 + \frac{M}{3} m_2(\alpha, \beta)|\epsilon_1|^4 + \frac{M}{6}|m_3(\alpha, \beta)||\epsilon_1|^5\right] \\
&\leq \frac{K_1}{n_0^{3/2}}.
\end{aligned}$$

for some constant K_1 depending only on α and β.

Secondly, the last term in equation 21 is bounded by:

$$\mathbb{E}\left[\frac{{m_1^{(3)}}^2(\xi)}{36} \epsilon_1^6\right] \leq \frac{M^2}{36} \mathbb{E}[\epsilon_1^6] = \frac{5 M^2 \sigma^6}{12 n_0^3} \leq \frac{K_2}{n_0^3},$$

for some constant K_2.

So the second term in 18 is bounded by

$$\left| \mathbb{E}_{\epsilon_1}\left[(m_1(\alpha - \epsilon_1, \beta - \epsilon_1) + \epsilon_1)^2\right] - m_1^2(\alpha, \beta) - \frac{m_2^2(\alpha, \beta) + m_1(\alpha, \beta) m_3(\alpha, \beta)}{n_0} \right| < \frac{K}{n_0^{3/2}} \tag{22}$$

for some constant K.

Combining 18, 19, and 22 completes the proof.

□

A.3 Iterative Retraining and Long-Term Dynamics in One-Dimensional Gaussian Mean Estimation

Now consider the verifier-guided synthetic retraining in the Gaussian mean estimation setting. The iterative retraining process can be described by the following algorithm.

Algorithm 1 Iterative Verifier-Guided Retraining for Gaussian Mean Estimation

1: **Input:** Initial estimate $\bar{X}^0$ from real data
2: **for** $k = 0, 1, 2, \dots$ **do**
3: Draw $\xi_i \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma^2)$ and construct synthetic samples $X_i^k = \bar{X}^k + \xi_i$.
4: Retain points with $a < X_i^k < b$, yielding n_k verified samples $\{X_i'^k : i = 1, 2, \dots n_k\}$.
5: $\bar{X}^{k+1} \leftarrow \frac{1}{n_k} \sum_{i=1}^{n_k} X_i'^k$.
6: **end for**

Algorithm 1 defines a Markov process $\{\bar{X}^0, \bar{X}^1, \dots \bar{X}^k, \dots\}$, where the conditional distribution $p(\bar{X}^{k+1}|\bar{X}^k)$ is given by

$$p(\bar{X}^{k+1}|\bar{X}^k) : \bar{X}^{k+1} = \bar{X}^k + \sigma \frac{\sum_{i=1}^{n_k} \xi_i'^{k+1}}{n_k}, \qquad \xi_i'^{k+1} \text{ i.i.d } \sim \mathcal{N}_{trunc}(\frac{a - \bar{X}^k}{\sigma}, \frac{b - \bar{X}^k}{\sigma}) \tag{23}$$

The following theorem summarizes these findings:

Theorem A.2. *Let $\bar{X}^k$ be the Markov process determined by equation 23 with initial condition*

$$\bar{X}^0 \sim \mathcal{N}(0, \frac{\sigma^2}{n_0}),$$

and assume n_k is non-decreasing in k. Then the following statements hold:

- *If $|a|, |b| < \infty$, there exists a constant $0 < \rho < 1$ such that,*

$$\mathbb{E}\left(\bar{X}^k - \frac{a+b}{2}\right)^2 \leq \rho^{2k} \mathbb{E}(\bar{X}^0 - \frac{a+b}{2})^2 + \sum_{j=0}^{k-1} \frac{\rho^{2(k-j)-1}}{n_j}.$$

  Moreover, if $\lim_{k\to\infty} n_k = \infty$, $\lim_{k\to\infty} \mathbb{E}|\bar{X}^k - \frac{a+b}{2}|^2 = 0$.

- *If $-\infty = a < b < \infty$, then $\liminf_{k\to\infty} \bar{X}^k = -\infty$. If $-\infty < a < b = \infty$, then $\limsup_{k\to\infty} \bar{X}^k = \infty$.*

Proof of Theorem A.2. Define

$$\epsilon_k = \frac{\bar{X}^k - \mu}{\sigma}, \tag{24}$$

which represents the standardized error of the estimator $\bar{X}^k$. It is easy to see that $\epsilon_k \in [\alpha, \beta] \Leftrightarrow \bar{X}^k \in [a, b]$, where α, β are defined in equation 13. Therefore, it suffices to consider the standardized process $\{\epsilon_k, k = 0, 1, 2, \dots\}$. equation 23 can be standardized as:

$$\epsilon_{k+1} = \epsilon_k + \frac{\sum_{i=1}^{n_k} \xi_i'^{k+1}}{n_k}, \qquad \xi_i'^{k+1} \sim \mathcal{N}_{\text{trunc}}\left(\alpha - \epsilon_k, \beta - \epsilon_k\right), \tag{25}$$

For convenience, we shift the noise terms $\xi_i'^{k+1}$ in equation 25 to have mean zero. Therefore, we introduce

$$T_{\alpha,\beta}(x) := x + \mathbb{E}\big[Z \;\big|\; \alpha - x \leq Z \leq \beta - x\big], \qquad v_{\alpha,\beta}(x) := \text{Var}\big(Z \;\big|\; \alpha - x \leq Z \leq \beta - x\big). \tag{26}$$

where $Z \sim \mathcal{N}(0,1)$.

Therefore, equation 25 can be rewritten as

$$\epsilon_{k+1} = T_{\alpha,\beta}(\epsilon_k) + \eta_{k+1} \tag{27}$$

where $\eta_{k+1} = \frac{1}{n_k}\sum_{i=1}^{n_k}\left(\xi_i'^{k+1} - \mathbb{E}\xi_i'^{k+1}\right)$ is the average of independent mean zero noise in equation 25. In particular, we have

$$\mathbb{E}[\eta_{k+1} \mid \mathcal{F}_k] = 0, \qquad \mathrm{Var}(\eta_{k+1} \mid \mathcal{F}_k) = \frac{v_{\alpha,\beta}(\epsilon_k)}{n_k}.$$

where $\mathcal{F}_k := \sigma(\epsilon_0, \eta_1, \ldots, \eta_k)$ and n_k is the (post-filtering) batch size at round k.

It is easy to see that

$$\begin{aligned} T_{\alpha,\beta}(x) &= x + m_1(\alpha - x, \beta - x), \\ v_{\alpha,\beta}(x) &= m_2(\alpha - x, \beta - x), \\ T'_{\alpha,\beta}(x) &= v_{\alpha,\beta}(x). \end{aligned}$$

We first consider $|a|, |b| < \infty$. In this case, we first show that the derterministic part $T_{\alpha,\beta}(x)$ in equation 27 is a global contraction. Since $-\infty < \alpha < \beta < \infty$, we have

$$\sup_{x\in\mathbb{R}} T'_{\alpha,\beta}(x) = \sup_{x\in\mathbb{R}} \mathrm{Var}\big(Z \mid \alpha - x \le Z \le \beta - x\big) = \mathrm{Var}\big(Z \mid |Z| < |\frac{\alpha+\beta}{2}|\big) := \rho < 1.$$

Therefore, $T_{\alpha,\beta}(x)$ is a global contraction. By the contractive mapping theorem that $T_{\alpha,\beta}(x)$ has a unique fixed point x^*, which solves $x^* = T_{\alpha,\beta}(x^*)$. It is easy to see that

$$x^* = T_{\alpha,\beta}(x^*) \implies x^* = x^* + \mathbb{E}(Z\big|\alpha - x^* \le Z \le \beta - x^*) \implies x^* = \frac{\alpha+\beta}{2}. \tag{28}$$

By the mean-value theorem,

$$|T_{\alpha,\beta}(\epsilon_k) - \frac{\alpha+\beta}{2}| \le \rho\,|\epsilon_k - \frac{\alpha+\beta}{2}|.$$

Let $V_k := (\epsilon_k - \frac{\alpha+\beta}{2})^2$, we have

$$\mathbb{E}\big[V_{k+1} \mid \epsilon_k\big] = (T_{\alpha,\beta}(\epsilon_k) - \frac{\alpha+\beta}{2})^2 + \frac{v_{\alpha,\beta}(\epsilon_k)}{n_k} \le \rho^2(\epsilon_k - \frac{\alpha+\beta}{2})^2 + \frac{\rho}{n_k}.$$

Taking expectations yields

$$\mathbb{E}V_{k+1} \le \rho^2\,\mathbb{E}V_k + \frac{\rho}{n_k}. \tag{29}$$

Unrolling equation 29,

$$\mathbb{E}V_k \le \rho^{2k}\mathbb{E}V_0 + \rho\sum_{j=0}^{k-1}\frac{\rho^{2(k-1-j)}}{n_j}. \tag{30}$$

It is easy to see that

$$\mathbb{E}V_k \le \rho^{2k}\mathbb{E}V_0 + \rho\sum_{j=0}^{k-1}\frac{\rho^{2(k-1-j)}}{n_0} < \rho^{2k}\mathbb{E}V_0 + \frac{\rho}{n_0(1-\rho^2)}.$$

Therefore, by the Cauchy-Schwarz inequality, $\lim_{k\to\infty}\mathbb{E}\epsilon_k^2 < \infty$ easily follows. Moreover, when $n_k \to \infty$, let $g_i := \rho^{2i}$ and $a_j := 1/n_j \to 0$. A standard ℓ^1-convolution argument shows $(g * a)_k := \sum_{j=0}^{k-1} g_{k-1-j}a_j = \sum_{j=0}^{k-1}\frac{\rho^{2(k-1-j)}}{n_j} \to 0$. Therefore $\lim_{k\to\infty}\mathbb{E}V_k = \lim_{k\to\infty}\mathbb{E}(\epsilon_k - \frac{\alpha+\beta}{2})^2 = 0$.

Now we consider the case $-\infty = a < b < \infty$ (equivalently $-\infty = \alpha < \beta < \infty$). We will show that $\liminf_{k\to\infty} \epsilon_k = -\infty$ a.s..

Let $t_k := \beta - \epsilon_k$ and the recursion equation 27 can be rewritten for t_k:

$$t_{k+1} = t_k + \lambda(t_k) - \eta_{k+1},$$

where $\lambda(t_k) = -\mathbb{E}(Z|Z < \beta - \epsilon_k) = \mathbb{E}[Z \mid Z \geq -t_k]$.

Consider the hitting time $\tau_M := \inf\{k : t_k \geq M\}$ for any $M > 0$. Fix $M > 0$ and define

$$m(M) := \min_{t \leq M} \lambda(t) = \mathbb{E}[Z \mid Z \geq -M] > 0,$$

which is strictly positive the fact that $\lambda(t) > 0$ and $\lambda(t)$ is a decreasing function. On the event $\{\tau_M > K\}$ we have $t_j < M$ for $j = 0, \ldots, K-1$, hence $\lambda(t_j) \geq m(M)$. Summing the recursion yields

$$t_K = t_0 + \sum_{j=0}^{K-1} \lambda(t_j) - \sum_{j=0}^{K-1} \eta_{j+1} \geq t_0 + K\, m(M) - S_K,$$

where $S_K := \sum_{j=0}^{K-1} \eta_{j+1}$ and $t_0 = \beta - \epsilon_0$ is $\mathcal{F}_0$-measurable (hence random). Therefore,

$$\{\tau_M > K\} \subseteq \Big\{S_K \geq t_0 + K\, m(M) - M\Big\}. \tag{31}$$

Define the (random) burn-in index

$$K_0 := \Big\lceil \frac{2(M - t_0)}{m(M)} \Big\rceil.$$

Then for all $K \geq K_0$,

$$t_0 + K\, m(M) - M \geq \frac{m(M)}{2} K,$$

and equation 31 gives, conditionally on $\mathcal{F}_0$,

$$\{\tau_M > K\} \subseteq \Big\{S_K \geq \frac{m(M)}{2} K\Big\}, \qquad \text{for all } K \geq K_0. \tag{32}$$

Next, we will show that S_K is a sub-exponential random variable in event $\{\tau_M > K\}$. Since $S_K = \sum_{j=0}^{K-1} \eta_{j+1} = \sum_{j=0}^{K-1} \frac{1}{n_j} \sum_{i=1}^{n_j} \left(\xi_i'^{j+1} - \mathbb{E}\xi_i'^{j+1}\right)$, we will first show that $\xi_i'^{j+1} - \mathbb{E}\xi_i'^{j+1}$ is sub-exponential.

Since $\xi_i'^{j+1} \sim \mathcal{N}_{\text{trunc}}(-\infty, \beta - \epsilon_j) = \mathcal{N}_{\text{trunc}}(-\infty, t_j)$, on the event $\{\tau_M > K\}$ we have

$$\xi_i'^{j+1} - \mathbb{E}\xi_i'^{j+1} < t_j - \mathbb{E}[Z \mid Z < t_j] \leq M - \mathbb{E}[Z \mid Z < M] := b(M) < \infty.$$

The above inequality follows from the fact that $t - \mathbb{E}[Z \mid Z < t]$ is an increasing function of t and $t_j < M$ for $j = 0, \ldots, K-1$ on the event $\{\tau_M > K\}$. In addition, $\mathrm{Var}(\xi_i'^{j+1}) = \mathrm{Var}(Z|Z < t_j) \leq 1$. Therefore, $\xi_i'^{j+1} - \mathbb{E}\xi_i'^{j+1}$ is mean zero, bounded above by $b(M)$ with $\mathrm{Var}\left(\xi_i'^{j+1} - \mathbb{E}\xi_i'^{j+1}\right) < 1$. By Bennet/Bernstein MGF inequality, we have

$$\log \mathbb{E} e^{\lambda(\xi_i'^{j+1} - \mathbb{E}\xi_i'^{j+1})} \leq \frac{\lambda^2}{2(1 - b(M)\lambda/3)},$$

for $0 < \lambda < \frac{3}{b(M)}$. This shows that $\xi_i'^{j+1} - \mathbb{E}\xi_i'^{j+1}$ is sub-exponential with parameters $SE(1, 2b(M)/3)$. By standard properties of sub-exponential random variables, $\eta_{j+1} = \frac{1}{n_j} \sum_{i=1}^{n_j} \left(\xi_i'^{j+1} - \mathbb{E}\xi_i'^{j+1}\right)$ is $SE(1/n_j, 2b(M)/(3n_j))$ and $S_K = \sum_{j=0}^{K-1} \eta_{j+1}$ is $SE(\sum_{j=0}^{K-1} 1/n_j, 2b(M)/(3n_1))$ since n_j is non-decreasing. Therefore, for any $t > 0$ we have tail bound

$$\mathbb{P}\left(S_K \geq t\right) \leq \exp\left(-\frac{1}{2}\min\{\frac{t^2}{\sum_{j=0}^{K-1} 1/n_j}, \frac{n_1 t}{2b(M)}\}\right) \leq \exp\left(-\frac{1}{2}\min\{\frac{n_1 t^2}{K}, \frac{n_1 t}{2b(M)}\}\right). \tag{33}$$

Use the tail bound equation 33 in equation 32, we have

$$\mathbb{P}\left(\tau_M > K \mid \mathcal{F}_0\right) \leq \mathbb{P}\left(S_K \geq \frac{m(M)}{2}K\right) \leq \exp\Big(-c(M)n_1K\Big) \tag{34}$$

for all $K \geq K_0$ with $c(M) = \min\left\{\frac{m(M)^2}{8}, \frac{n(M)}{8b(M)}\right\}$.

$$\begin{aligned}\mathbb{P}\left(\tau_M > K\right) &= \mathbb{E}\left[\mathbb{P}\left(\tau_M > K \mid \mathcal{F}_0\right)\right] \\ &\leq \mathbb{E}\left[\exp\Big(-c(M)n_1K\Big)\mathbb{1}_{\{K>K_0\}}\right] + \mathbb{P}\left(K \leq K_0\right)\end{aligned} \tag{35}$$

Let $K \to \infty$ in equation 35, we get $\mathbb{P}(\tau_M < \infty) = 1$. Since M is arbitrary, this implies $\liminf_{k\to\infty} \epsilon_k = -\infty$ a.s..

The case $-\infty < a < b = \infty$ can be proved in the same way, therefore is omitted.

□

B PROOFS OF ALL THEOREMS IN SECTION 2

Because of our special synthetic data design, the OLS estimator is equivalent to learning each coordinate of θ along the orthogonal directions $\{v_j\}$ separately. We can therefore rewrite the retraining procedure as follows:

Algorithm 2 Iterative Verifier-Guided Retraining in Linear Regression

Input: Real data (X^0, Y^0)
Compute initial estimator $\hat{\theta}^0 = ({X^0}^\top X^0)^{-1} {X^0}^\top Y^0$
Let $X^0 = U\Sigma V^\top$ be the SVD of X^0, with right singular vectors $V = (v_1, \dots, v_p)$
for $k = 0, 1, 2, \dots$ **do**
  for $j = 1, \dots, p$ **do**
    Construct synthetic design matrix $X^{k+1,j}$ with all rows equal to $v_j^\top$
    Generate synthetic responses $Y^{k+1,j} = X^{k+1,j}\hat{\theta}^k + \sigma\xi^{k+1,j}$, where $\xi^{k+1,j} \sim \mathcal{N}(0, I)$
    Apply verifier to each $(x_i^{k+1,j}, y_i^{k+1,j})$ and retain valid samples satisfying

$$|y_i^{k+1,j} - (x_i^{k+1,j})^\top \theta_c| \le r\|x_i^{k+1,j}\| + \sigma_c, \tag{36}$$

    yielding n_k verified samples $(x_i'^{k+1,j}, y_i'^{k+1,j})$.
    Compute one-dimensional estimator

$$\hat{\theta}^{k+1,proj,j} = \bar{y'}^{k+1,j} \tag{37}$$

  end for
  Update overall estimator:

$$\hat{\theta}^{k+1} = \sum_{j=1}^{p} v_j \hat{\theta}^{k+1,proj,j} \tag{38}$$

end for

Proof of Theorem 3.1. We consider the one dimensional projection estimator of $\hat{\theta}^{1,proj,j}$ defined in equation 37. The filter condition equation 36 is equivalent to:

$$\begin{aligned} &|\sigma\xi_i^{1,j} + v_j^\top(\hat{\theta}^0 - \theta_c)| \le r + \sigma_c \\ \iff & y_i^{1,j} = \sigma\xi_i^{1,j} + v_j^\top\hat{\theta}^0 \in \left(-r - \frac{\sigma_c}{\sigma} + v_j^\top\theta_c, r + \frac{\sigma_c}{\sigma} + v_j^\top\theta_c\right). \end{aligned} \tag{39}$$

Note that $\hat{\theta}^0 \sim \mathcal{N}(\theta^\star, ({X^0}^\top X^0)^{-1}\sigma^2)$ and v_j is the j-th right singular vector of X^0, therefore $v_j^\top\hat{\theta}^0 \sim \mathcal{N}(v_j^\top\theta^\star, \sigma^2\mu_j^{-2})$. Therefore, $\hat{\theta}^{1,proj,j} = \bar{y}'^{1,j}$ correspond to the verifier-filtered mean estimator of a one-dimensional Gaussian mean estimation problem with true mean $v_j^\top\theta$, variance $\sigma^2\mu_j^{-2}$ and filtering interval $\left(-r - \frac{\sigma_c}{\sigma} + v_j^\top\theta_c, r + \frac{\sigma_c}{\sigma} + v_j^\top\theta_c\right)$. Let

$$\begin{aligned} \alpha_j &:= \frac{-r - \sigma_c + v_j^\top(\theta_c - \theta^\star)}{\sigma}, \\ \beta_j &:= \frac{r + \sigma_c + v_j^\top(\theta_c - \theta^\star)}{\sigma}. \end{aligned} \tag{40}$$

Under the assumption $\mu_j = \omega(\sqrt{n_0})$, there exists a constant $L > 0$, such that $\mu_j^2 > Ln_0$ for all $j = 1, \dots, p$. Therefore, by Theorem A.1, there exists constant K_j depending only on α_j, β_j such that if $n_1 > n_0 \ge 100$,

$$\begin{aligned} &\left|\frac{1}{\sigma^2}\mathbb{E}(\hat{\theta}^{1,proj,j} - v_j^\top\theta^\star)^2 - \frac{m_2(\alpha_j,\beta_j)}{n_1} - \left(m_1^2(\alpha_j,\beta_j) + \frac{m_2^2(\alpha_j,\beta_j) + m_3(\alpha_j,\beta_j)m_1(\alpha_j,\beta_j)}{\mu_j^2}\right)\right| \\ &< K_j\left(\frac{1}{n_1 n_0^{1/3}} + \frac{1}{n_0^{3/2}}\right) \end{aligned} \tag{41}$$

will hold with probability at least $1 - \exp(-Ln_0^{1/3})$. m_1, m_2, m_3 are defined in equation 14 and equation 15. By equation 3, we have $\hat{\theta}^{1,proj,j} = v_j^\top \hat{\theta}^1$. In addition, since $V = (v_1, v_2, \ldots, v_p)$ is an orthonormal matrice, we have

$$\sum_{j=1}^{p} \mathbb{E}(\hat{\theta}^{1,proj,j} - v_j^\top \theta^\star)^2 = \sum_{j=1}^{p} \mathbb{E}(v_j^\top \hat{\theta}^1 - v_j^\top \theta^\star)^2 = \mathbb{E}||V^\top(\hat{\theta}^1 - \theta^\star)||^2 = \mathbb{E}\|\hat{\theta}^1 - \theta^\star|^2. \quad (42)$$

Therefore, by summing over j on both sides of equation 41 and using simple union bound, we established equation 8 with $K = \max_j K_j$ and

$$\begin{aligned} m_{1,j} &:= m_1(\alpha_j, \beta_j), \\ m_{2,j} &:= m_2(\alpha_j, \beta_j), \\ m_{3,j} &:= m_3(\alpha_j, \beta_j). \end{aligned}$$

□

Proof of Theorem 4.1. We consider the transition dynamics of $\hat{\theta}^k$ in Algorithm 2. Since we designed $X^{k,j}$ to be the rank one matrix correspond to singular vector v_j, therefore equation **??** reduces to a one-dimensional estimation equation:

$$\hat{\theta}^{k+1,proj,j} = v_j^\top \hat{\theta}^k + \frac{\sigma}{n_k} \sum_{i=1}^{n_k} \xi_i'^{k+1,j} \quad (43)$$

where $\xi_i'^{k+1,j}$ is the truncated noise term after verification. By equation 36, we have

$$\xi_i'^{k+1,j} \text{ i.i.d} \sim \mathcal{N}_{trunc}\left(-\frac{r}{\sigma} - \frac{\sigma_c}{\sigma} - v_j^\top \frac{\hat{\theta}^k - \theta_c}{\sigma}, \frac{r}{\sigma} + \frac{\sigma_c}{\sigma} - v_j^\top \frac{\hat{\theta}^k - \theta_c}{\sigma}\right). \quad (44)$$

We consider the rotated standardized estimator

$$\epsilon_j^k := v_j^\top \frac{\hat{\theta}^k - \theta_c}{\sigma} \quad \text{equivalently} \quad \epsilon^k := V^\top \frac{\hat{\theta}^k - \theta_c}{\sigma}.$$

Since $\hat{\theta}^{k+1,proj,j} = v_j^\top \hat{\theta}^{k+1}$ by equation 38, equation 43 can be standardized as

$$\epsilon_j^{k+1} = \epsilon_j^k + \frac{\sum_{i=1}^{n_k} \xi_i'^{k+1,j}}{n_k}, \qquad \xi_i'^{k+1,j} \text{ i.i.d} \sim \mathcal{N}_{trunc}\left(-\beta - \epsilon_j^k, \beta - \epsilon_j^k\right) \quad (45)$$

where $\beta = \frac{r}{\sigma} + \frac{\sigma_c}{\sigma}$. We note that equation 45 is exactly the same dynamics we consider in the proof of Theorem A.2 with $\beta = -\alpha < \infty$. In other words, the evolution of the iterative estimator ϵ^k is diagonal and each cordinates follows the same dynamics as the one dimensional gaussian iterative mean estimator. From Theorem A.2, we known that there exists a constant $\rho < 1$ such that

$$\mathbb{E}\|\epsilon_j^k\|^2 \le \rho^{2k}\mathbb{E}\|\epsilon_j^0\|^2 + \sum_{j=0}^{k-1} \frac{\rho^{2(k-j)-1}}{n_j}, \qquad j = 1, 2, \ldots, p.$$

This implies that

$$\mathbb{E}\|\hat{\theta}^k - \theta_c\|^2 \le \rho^{2k}\mathbb{E}\|\hat{\theta}^0 - \theta_c\|^2 + p\sigma^2 \sum_{j=0}^{k-1} \frac{\rho^{2(k-j)-1}}{n_j}.$$

□

C Additional Details on CVAE Experiments

Data preprocessing. We use MNIST (28×28 grayscale) and normalize pixel intensities to $[0, 1]$. Class labels are represented as one-hot vectors $y \in \{0, 1\}^K$ (K=10).

Experiment Details. We use a convolutional CVAE model consisting of an Encoder with two convolutional layers (1→32 and 32→64 channels, 4×4 kernels, stride 2, with GELU activations), followed by a linear projection that outputs the mean and log-variance of a $d_z = 20$-dimensional Gaussian latent space. The Decoder mirrors this structure: a linear layer maps the latent code to a $64 \times 7 \times 7$ tensor, which is upsampled by two transposed convolutional layers (64→32 and 32→1 channels, 4×4 kernels, stride 2, with GELU activations) to reconstruct 28×28 images. We train the CVAE with the standard objective, i.e., binary cross-entropy reconstruction loss plus KL divergence regularization.

Discriminator for filtering. We additionally train a discriminator D to distinguish real from synthetic samples. D is implemented as a multi-layer perceptron: five fully connected layers with hidden sizes 512, 256, 128, and 64, each followed by a LeakyReLU activation, and a final linear layer mapping to a single logit. The output is passed through a sigmoid to yield the probability of the input being real. The discriminator is trained with binary cross-entropy, labeling real MNIST digits as positive and CVAE-generated digits as negative.

Synthetic generation and filtering. After each training round, we generate conditioned samples by drawing $z \sim \mathcal{N}(0, I)$, choosing labels y (uniform over classes unless specified), and decoding $\tilde{x} = g_\theta(z, y)$. To control sample quality, we score each $(\tilde{x}, y)$ with the discriminator $D(\tilde{x}, y)$. For each class, we retain only the top 10% of generated samples with the highest discriminator scores. These filtered synthetic samples are then combined with the real dataset to form the training data for the next round.

Supplementary Results on Test ELBO We also evaluate generative performance using the test ELBO, a standard metric for VAEs. Compared to FID, ELBO proves substantially harder to improve—likely because ELBO penalizes per-pixel deviations, while FID emphasizes perceptual quality. We adopted a much more aggressive synthetic size schedule than in our earlier experiments. Starting from 500 real samples, we first increase the synthetic size to 30K—a point at which further increases yield diminishing returns—then linearly scale over 20 rounds until reaching 1M synthetic samples, which already stretched our computational budget.

Figure 5 reports test ELBO over these 20 rounds. Consistent with our bias-variance analysis, we observe clear improvement in the early stages (up to about round 5-10). After that, however, ELBO deteriorates beyond round 10.

We attribute this both to the verifier's limitations, as discussed in the main text, and to the fact our verifier (implemented via a discriminator) emphasizes more on perceptual quality rather than likelihood-based reconstruction. This observation is also consistent with our theoretical prediction: verifier bias can lead to a reversal in loss trends, negating the early gains realized by bias-variance trade-offs.

As a result, our retrained models achieve much sharper, cleaner digits with significantly improved FID, even when ELBO stagnates or worsens. We believe that with stronger verifiers better aligned with the true data distribution, iterative retraining could improve not only perceptual metrics like FID but also likelihood-based metrics such as ELBO.

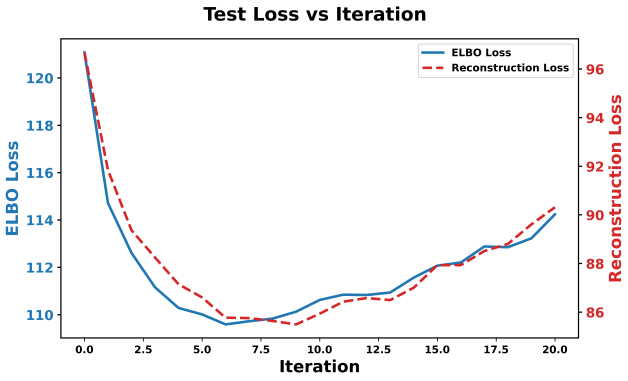


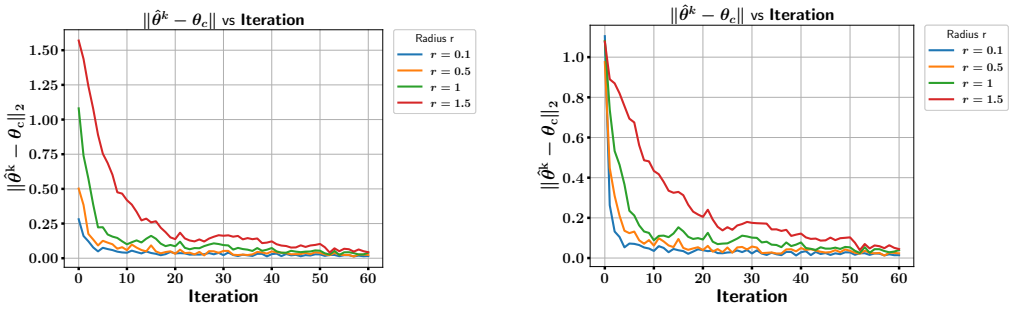
Figure 5: Test ELBO and reconstruction loss across retraining rounds.

D Additional Experimental Results

D.1 Random Synthetic Data in Linear Regression

In the main text, the synthetic covariates were aligned with a fixed orthonormal basis to simplify analysis and make the retraining dynamics easier to interpret. To show that the observed behavior is not tied to this structured design, we repeat the same iterative retraining experiment using fully random synthetic covariates sampled i.i.d. from a standard Gaussian distribution.

Figure 6 presents the results, corresponding directly to the two panels in Figure 2 of the main text, but under the random-design setting. The qualitative behavior remains the same: with a well-specified verifier, retraining contracts toward the verifier's knowledge center and avoids collapse, whereas unfiltered retraining diverges. This confirms that the verifier-induced stability and improvement patterns hold beyond the orthonormal-design assumption.



(a) Verifier with bias (random design)

(b) Verifier without bias (random design)

Figure 6: Iterative synthetic retraining under random synthetic covariates, corresponding to the structured-design results in Figure 2.

D.2 Different Verifier Shapes

We further analyze how different geometric choices of the verifier region affect the acceptance rule and the resulting retraining dynamics. For any region $\mathcal{R}_\theta$ around a center θ_c, a synthetic point (x, y) is accepted whenever there exists a parameter perturbation Δ in the region that can explain y, i.e.

$$y = x^\top(\theta_c + \Delta) + \xi, \qquad \Delta \in \mathcal{R}_\theta.$$

This leads to the general acceptance requirement

$$|y - x^\top \theta_c| \le \sup_{\Delta \in \mathcal{R}_\theta} |x^\top \Delta| + \sigma_c.$$

Different verifier shapes correspond to different support functions $\sup_{\Delta \in \mathcal{R}_\theta} |x^\top \Delta|$.

(1) Ellipsoidal verifier. Consider the anisotropic ellipsoid

$$\mathcal{R}_\theta = \{\theta : (\theta - \theta_c)^\top A(\theta - \theta_c) \le r^2\}, \qquad A \succ 0.$$

Let $\Delta = \theta - \theta_c$. Changing variables $\Delta = A^{-1/2}u$ with $\|u\|_2 \le r$ yields

$$\sup_{\Delta^\top A \Delta \le r^2} |x^\top \Delta| = r\|A^{-1/2}x\|_2 = r\sqrt{x^\top A^{-1} x}.$$

Thus the acceptance condition becomes

$$|y - x^\top \theta_c| \le r\sqrt{x^\top A^{-1} x} + \sigma_c.$$

(2) Polyhedral ℓ_1 verifier. For the ℓ_1 knowledge region

$$\mathcal{R}_\theta = \{\|\theta - \theta_c\|_1 \le r\},$$

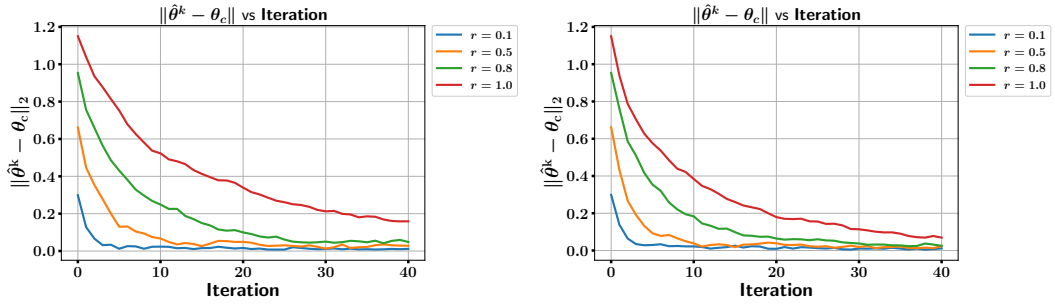
the perturbation satisfies $\|\Delta\|_1 \le r$. Using Hölder duality,

$$\sup_{\|\Delta\|_1 \le r} |x^\top \Delta| = r\|x\|_\infty.$$

The corresponding acceptance rule is

$$|y - x^\top \theta_c| \le r\|x\|_\infty + \sigma_c.$$

Although ellipsoidal and ℓ_1 (polyhedral) regions induce different forms of acceptance sets, both yield the same qualitative retraining behavior: $\hat{\theta}^{(k)}$ *consistently move toward the verifier center* θ_c. The empirical trajectories under both shapes are shown in Figure 7.



(a) Ellipsoidal verifier (b) Polyhedral ℓ_1 verifier

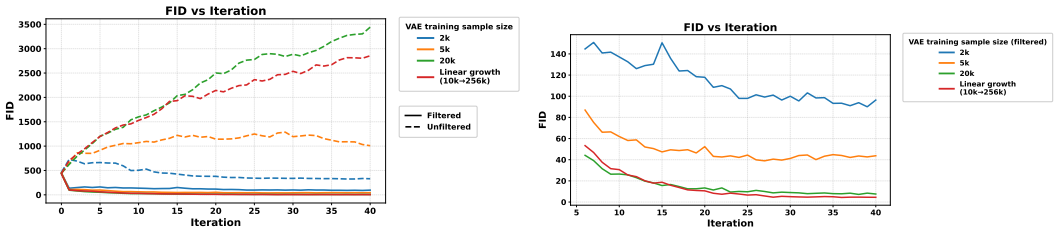
Figure 7: Retraining trajectories under two different verifier shapes. In both cases, $\hat{\theta}^{(k)}$ empirically converges toward the verifier center θ_c.

D.3 MNIST-Specific FID Evaluation

The standard Fréchet Inception Distance (FID) is widely used in generative modeling, including on MNIST, following prior work such as Dai & Wipf (2019); Leontev et al. (2020); Chan & Sithungu (2024). Nonetheless, we agree that Inception embeddings are not tailored to handwritten digits and may not fully capture perceptual similarity on MNIST.

To address this point, we introduce a **MNIST-specific FID** variant. We train a lightweight convolutional network directly on MNIST classification, and compute FID using the penultimate-layer activations as the embedding space. This produces a domain-appropriate FID measure while preserving the same statistical structure as the original metric. These results confirm that our conclusions are robust to the choice of embedding and do not depend on the use of vanilla FID.

Results. Figures 8a and 8b report the new FID scores under our retraining framework for all verifier sizes. Consistent with the standard FID curves in the main paper.



(a) **MNIST-specific FID over retraining iterations.** (b) **Zoom-in view for filtered runs.**

Figure 8: **MNIST-specific FID using our MNIST-trained feature embedding.** Both results confirm that our conclusions remain unchanged when replacing standard FID with a domain-specific metric.

D.4 Qualitative Evaluation of Generated Samples

To complement the quantitative FID analysis, we present a compact qualitative comparison of representative MNIST samples generated under different retraining and verification regimes. All models begin from the same initialization trained on 500 real MNIST images. During retraining, the quality of the verifier plays a crucial role: in our experiments, a *weak verifier* is trained on only 500 real images together with an equal number of synthetic images, whereas a *strong verifier* has access to all 60,000 real MNIST images. This contrast provides a clear illustration of how verification quality affects synthetic retraining.

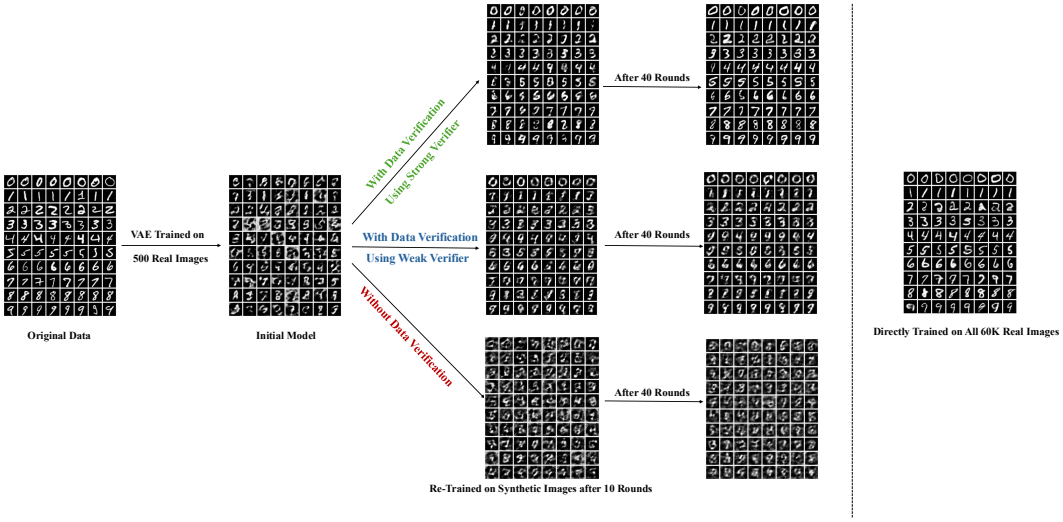


Figure 9: **Qualitative comparison of generated MNIST samples across retraining and verification settings.** From left to right: the initial model trained on 500 real images; results after 10 rounds of synthetic-data retraining; and results after 40 rounds under three conditions—strong verifier (trained with 60k real images), weak verifier (trained with 500 real + 500 synthetic images), and no verification.

The patterns in Figure 10 clearly illustrate the role of verifier. The strong verifier enables the generator to progressively sharpen digit structure and contrast, ultimately approaching the quality of a model trained directly on all 60,000 real MNIST images. In contrast, weak or absent verification allows synthetic gradients to drift, resulting in blurry or distorted digits. These qualitative observations closely match the quantitative FID trends reported in the main text.

D.5 Different Initial Sample Sizes

We assess the robustness of verifier-guided retraining by varying the number of real MNIST images used to train the initial CVAE ($1k, 2k, 3k, 4k, 60k$). For small and medium initial sample sizes, verifier filtering yields clear early FID improvements and then stabilizes performance, whereas unfiltered retraining quickly degrades. When the generator is initialized on all 60k real images, verifier filtering no longer improves FID over the initial model, but it still effectively prevents the severe collapse observed under unfiltered retraining.

We perform our main experiments in a low–real-data regime (e.g., with 500 initial images), where the verifier, having been trained on a much larger subset of MNIST, holds strictly more external knowledge than the generator. According to our theory, this is exactly the regime in which verifier-guided retraining should provide true improvement rather than simple stabilization, because the verifier contributes additional information. In contrast, when the generator is initialized on the full 60k training images, the verifier would need access to an even stronger source of external information to achieve improvement; otherwise it can only prevent collapse. For this reason, the small initial sample size serves as the most informative regime for highlighting the verifier's knowledge-injection effect and demonstrating the improvement phenomena predicted by our theoretical framework.

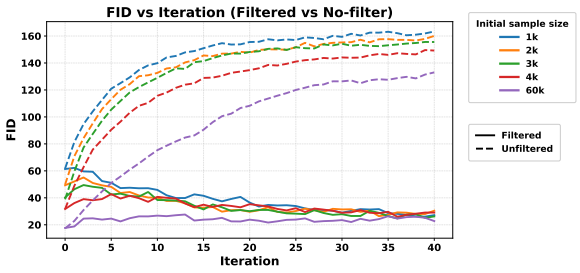


Figure 10: FID across retraining iterations under different initial sample sizes, comparing verifier-filtered and unfiltered synthetic retraining.

D.6 Additional Experiments on Large-Scale News Summarization

To further assess the generality of our theoretical predictions beyond image-based generative settings, we conducted additional experiments on the XSUM news-summarization dataset Narayan et al. (2018), a widely used natural-language benchmark. Our goal is to evaluate whether verifier-filtered synthetic retraining improves a pretrained language model's performance on realistic natural-language tasks.

Base model and training setup. We use the `SmolLM2-135M` model Allal et al. (2025) as our generator. We follow a similar experimental setup to Feng et al. (2024), who evaluate a single round of retraining on synthetic summaries. In contrast, our study focuses on the *multi-iteration* retraining regime, enabling us to examine how performance evolves over repeated generate–filter–retrain cycles. The model is first fine-tuned on 12.5% of the XSUM training set for one epoch using full-parameter training. We follow common summarization practice and employ greedy decoding for both generation and evaluation, given the low-entropy nature of news summarization.

Synthetic retraining procedure. After the initial fine-tuning step, we perform iterative synthetic retraining as follows:

1. Generate a synthetic summary y' for each of the 204,045 training articles using greedy decoding.
2. For each article–summary pair (x, y'), compute the ROUGE-1 score between y' and its ground-truth summary y.
3. Select the top 12.5% of synthetic examples according to ROUGE-1 (oracle filter).
4. Retrain the model on these filtered synthetic examples.
5. Repeat the generate–filter–retrain loop for multiple iterations, recording the ROUGE-1 score on the held-out test set after each iteration.

This setup directly mirrors our theoretical framework: high-quality filtering should improve performance during early iterations, followed by eventual plateauing as the model approaches the verifier's "knowledge center".

Results. Figure 11 reports the ROUGE-1 score across 15 rounds of synthetic retraining for both the filtered and unfiltered conditions. For the unfiltered condition, we retrain on the same number of synthetic examples as the filtered condition. The filtered retraining procedure yields a consistent, monotonic improvement during the first several iterations before stabilizing, in agreement with our theoretical predictions. In contrast, the unfiltered retraining baseline shows no meaningful improvement and fluctuates around its initial performance level, illustrating that synthetic retraining without quality control does not enhance performance. These results further show that our theory holds in large-scale settings, where the dynamics of synthetic retraining closely match our theoretical predictions.

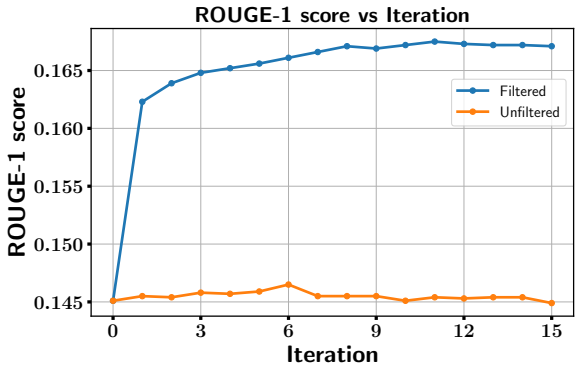


Figure 11: **ROUGE-1 score vs. iteration on the XSUM dataset.** Filtered retraining leads to consistent improvement across early rounds, while unfiltered retraining exhibits no significant gain.

Computational resources. All experiments were conducted on a dedicated cluster equipped with 4 NVIDIA A800 GPUs (256 GB memory each). Training and inference follow a standard full-finetuning pipeline with batch size 32, learning rate 5×10^{-5}, cosine scheduler, one training epoch, and maximum sequence length of 256.

E USE OF LARGE LANGUAGE MODELS

The authors acknowledge the use of ChatGPT for assistance in improving plot figures, as well as for checking grammar and spelling. All scientific contributions, analyses, and interpretations are solely the work of the authors.