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

Bingji Yi*
Independent Researcher
yibingji@gmail.com

Qiyuan Liu*
Department of Statistics
University of Chicago
qiyuanliu@uchicago.edu

Yuwei Cheng
Department of Statistics
University of Chicago
yuweicheng@uchicago.edu

Haifeng Xu
Department of Computer Science
University of Chicago
haifengxu@uchicago.edu

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 setting, showing that verifier-guided retraining can yield near-term improvements but ultimately drives the parameter estimate to the verifier's "knowledge center" in the long run. Our theory further predicts that, unless the verifier is perfectly reliable, these early gains will plateau and may even reverse. Indeed, our experiments across linear regression, Variational Autoencoders (VAEs) trained on MNIST, and SmolLM2-135M on the XSUM task confirm these theoretical insights.

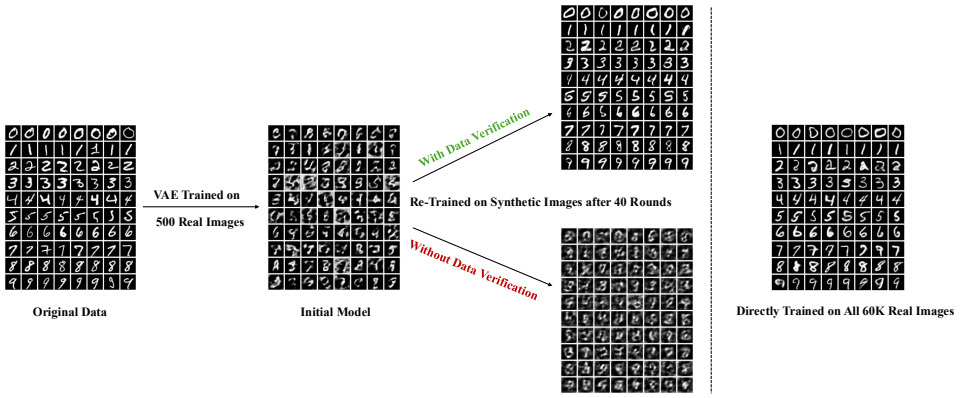


Figure 1: **Iterative VAE Retraining on MNIST. Left:** Original MNIST images (real data). **Middle:** Samples from a VAE trained on 500 real images. **Right:** Samples after 40 rounds of synthetic retraining. The *top branch* (green) uses verifier-filtered synthetic data, producing clearer and more realistic digits; the *bottom branch* (red) retrains without verification, leading to severe degradation and mode collapse. The final column shows a VAE trained on all 60K real images (upper bound on quality).

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; 2025; 2024b; Alemohammad et al., 2024; Gerstgrasser et al., 2024).

In practice, synthetic data are rarely used in raw form. Instead, practitioners typically apply filtering steps to remove low-quality samples before retraining. For example, in natural language generation, synthetic text 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; and in recommendation or preference learning, synthetic feedback is often validated using external heuristics or known user signals (Tu et al., 2025; Iskander et al., 2024; Lupidi et al., 2024; Lampis et al., 2023; Zhang et al., 2024). A common abstraction underlying these approaches is the use of a *verifier* that evaluates candidate synthetic samples and retains only those that pass 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., 2025) 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; 2025; 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.

Related Work. A detailed discussion of related work is provided in Appendix A.

2 MODELING VERIFIER-BASED SYNTHETIC RETRAINING: THE LINEAR REGRESSION CASE

In this section, we formalize our model of iterative retraining with verified synthetic data, coined *verifier-based synthetic retraining* for convenience. Following recent works in this space (Dohmatob et al., 2024a; Gerstgrasser et al., 2024; Garg et al., 2025; Zhu et al., 2025), we focus on the foundational

linear regression setting where the objective is to estimate a high-dimensional coefficient vector $\theta^\star$ in the following 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 use the standard Mean Squared Error (MSE), i.e., $\text{MSE}(\hat{\theta}) = \mathbb{E}_\xi \|\hat{\theta} - \theta^\star\|^2$, to evaluate estimators.

Modeling the verifier and data filtering rule. Suppose we have access to a verifier that possesses prior knowledge of $\theta^\star$, modeled by a knowledge set. Specifically, the verifier's knowledge is described by a spherical ball:

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

with fixed center θ_c and radius r. We assume this knowledge set indeed contains the true parameter, i.e., $\theta^\star \in B_r(\theta_c)$, but the true parameter $\theta^\star$ is unknown. The verifier does not reveal θ_c or r directly (see modeling motivations below). Instead, it only provides binary feedback indicating whether a given (real or synthetic) data point (x_i, y_i) is consistent with the knowledge $\theta^\star \in B_r(\theta_c)$ or not. Specifically, the verifier outputs *Yes* if

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

and *No* otherwise. Here σ_c is a constant related to the verifier's capability. This *filtering rule* is motivated by the following bound on expected errors: $\mathbb{E}\left[\, |y_i - x_i^\top \theta_c| \,\right] = \mathbb{E}\left[\, |x_i^\top (\theta^\star - \theta_c) + \xi_i| \,\right] \leq r\|x_i\| + \mathbb{E}|\xi_i| = r\|x_i\| + \sqrt{\frac{2}{\pi}}\sigma$. Since the true σ might be unknown in practice, σ_c serves as an estimate of the true σ.

We refer to $\Delta = \|\theta^\star - \theta_c\|$ as the *bias* of the verifier, whereas r captures the *selectivity* of the verifier – the smaller r is, less likely the verifier accepts a data point (x_i, y_i). The verifier only needs to provide Yes/No answers based on the above selection rule in equation 1, but does not needs to know the parameter θ_c, r of the knowledge set. The motivation of this modeling primarily comes from practice, as explained below.

Motivation of binary feedback from verifiers. We adopt the binary feedback from verifiers mainly for practical reasons. 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 real applications verifiers may not even know these quantities explicitly, which would correspond to model parameters if the verifier is a stronger teacher model or how the human reasons if the verifier is a human. This model choice is also aligned with the widely adopted comparison-based feedback in reinforcement learning from human feedback (RLHF) (Ouyang et al., 2022). Such binary feedback has become a standard approach 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 (Wettig et al., 2024). Although simple, our theory and empirical evaluations both show that this *single bit* of information for each sample can successfully be injected into the retraining process to improve models.

Synthetic Retraining with Verifier-based Filtering We begin with an initial 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) [1] $\hat{\theta}^0 = ({X^0}^\top X^0)^{-1} {X^0}^\top Y^0$. We then proceed with iterative synthetic retraining via the *generate–verify–retrain* procedure outlined in Figure 2, the rigorous retraining Scheme **??** and Algorithm 2 are provided in Appendix D.

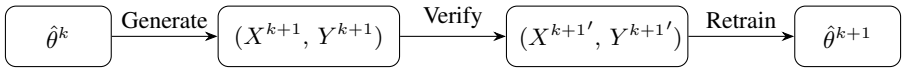


Figure 2: Generate-Verify-Retrain pipeline.

Since 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

[1] For ease of presentation, we assume $\text{Rank}(X^0) = p$. If $\text{Rank}(X^0) < p$, all our results are equally applicable by working in the subspace of X^0.

constraints. In principle, one could construct X^k arbitrarily; however, for mathematical clarity, below we describe a targeted though arguably natural design. In particular, we choose to 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:

$$X^k = (\underbrace{v_1, \ldots}_{\text{copies of } v_1}, \underbrace{v_2, \ldots}_{\text{copies of } v_2}, \ldots, \underbrace{v_p, \ldots}_{\text{copies of } v_p})^\top. \tag{2}$$

After verifier filtering, each orthogonal direction v_j retains exactly n_k samples with $n_0 \leq pn_1 \leq pn_2 \leq \cdots$.

Notably, the estimation of parameter $\hat{\theta}^{k+1}$ using only synthetic data from the model with $\hat{\theta}^k$, though with filtering, leads to a Markovian transition $\hat{\theta}^k \mapsto \hat{\theta}^{k+1}$. The above block design essentially helps "diagonalize" the transition operator $\hat{\theta}^k \mapsto \hat{\theta}^{k+1}$. The conceptual benefit of this covariance design choice is that we remove the rotational variability that arbitrary designs would introduce across iterations and decouple the dynamics along orthogonal directions. In practice, this design mirrors curating data along approximately orthogonal latent spaces or topics (e.g., topical axes like politics, sports, mathematics). However, the choice of covariant X^k is not unique: alternatives (e.g., canonical basis, isotropic random directions) can yield similar qualitative conclusions, with potentially different constants or rates. We expect our theoretical insights to generalize to any reasonable design of the covariant X^k, though the rigorous proofs may be less tractable for some designs.

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 underlying mechanism is a verifier-induced bias-variance trade-off: filtering synthetic data *reduces variance* but may *introduce bias*.

3.1 Source of Improvement: Bias–Variance Trade-off

To understand *why* verifier-based retraining can improve upon the initial estimator $\hat{\theta}^0$, we must examine the fundamental bias–variance trade-off introduced by the filtering process. The initial estimator $\hat{\theta}^0$ is unbiased but suffers from high variance due to the limited real sample size n_0. When we generate synthetic data and apply the verifier, we effectively discard inconsistent samples. This filtering reduces the estimation variance. However, because the verifier itself may be imperfect, this filtering injects a systematic bias. Therefore, synthetic retraining yields a strictly better model precisely when the variance reduction achieved by filtering outweighs the injected bias and the sampling noise of the synthetic data itself.

3.2 Characterizing Improvement in One-Round Retraining

The following theorem rigorously quantifies this trade-off for the linear regression model, demonstrating exactly *when* the one-step estimator $\hat{\theta}^1$ improves or degrades upon the initial baseline. By characterizing the MSE of $\hat{\theta}^1$, it reveals how synthetic sample size, verifier bias, and verifier selectivity determine the final outcome.

Theorem 3.1. *Let $\{\mu_j\}_{j=1}^p$ denote the singular values of X^0, and assume each of them satisfies $\mu_j = \Omega(\sqrt{n_0})$.*[2] *Then there exist constants $m_{1,j}, m_{3,j} \in \mathbb{R}$ and $m_{2,j} \in (0,1)$ for $j = 1, \ldots, p$, as well as a constant $L > 0$ such that:*

$$\frac{1}{\sigma^2} MSE(\hat{\theta}^1) = \sum_{j=1}^{p} \left(\underbrace{\frac{m_{2,j}}{n_1}}_{\textit{Synthetic Variance}} + \underbrace{m_{1,j}^2 + \frac{m_{1,j} m_{3,j} + m_{2,j}^2}{\mu_j^2}}_{\textit{Verification Error}} \right) + \mathcal{O}\left(n_0^{-4/3}\right) \tag{3}$$

[2]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.

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

While the explicit forms of the constants are deferred to Appendix D, their roles are highly intuitive: $m_{1,j}$ and $m_{3,j}$ capture the directional bias between the verifier's knowledge center θ_c and the ground truth θ^* along the j-th singular direction (vanishing if $\theta_c = \theta^*$), while $m_{2,j} < 1$ quantifies the variance reduction along that direction. Theorem 3.1 mathematically guarantees when verifier-guided retraining improves the model. Since the scaled baseline error is $\frac{1}{\sigma^2}\text{MSE}(\hat{\theta}^0) = \sum_{j=1}^{p} \mu_j^{-2}$, we can directly compare it to Equation 3. When the verifier is highly accurate ($m_{1,j}, m_{3,j} \approx 0$), the verification error term becomes dominated by $\sum_{j=1}^{p} m_{2,j}^2/\mu_j^2$. Because $m_{2,j} < 1$, this verification error is strictly smaller than the real-data error. Thus, whenever the verified synthetic sample size n_1 is sufficiently large to drive the synthetic variance down, $\text{MSE}(\hat{\theta}^1)$ strictly improves upon the baseline.

This result highlights why verifier-based retraining is practically useful: in modern machine learning systems where real data collection is costly but generative models or simulators are available, a moderately accurate verifier can filter synthetic samples to effectively amplify limited real-world evidence and substantially reduce estimation error. Conceptually, this offers a sharp departure from classical model collapse literature, which typically models iterative synthetic data purely as a variance-inflating noise source (Shumailov et al., 2024; Alemohammad et al., 2024; Dohmatob et al., 2024a). Here, we prove that *verification transforms synthetic data into a variance-reducing resource*, provided the verifier's bias is sufficiently controlled. As we will demonstrate empirically in Section 5, this mechanism is not confined to the linear model; it manifests clearly in complex models such as VAEs and LLMs.

4 Iterative Retraining Converges to the Verifier's Knowledge Center

Having shown that a single round of verifier-based retraining can improve estimation through a bias–variance trade-off, a natural question is whether this improvement persists over multiple rounds and what the long-term outcome is. To connect with the model collapse literature, we formalize these phenomena in our linear regression setting. Specifically, we define **Model Degradation/Collapse** as $\limsup_{k\to\infty} \text{MSE}(\hat{\theta}^k) > \text{MSE}(\hat{\theta}^0)$ and **Model Improvement** as $\limsup_{k\to\infty} \text{MSE}(\hat{\theta}^k) < \text{MSE}(\hat{\theta}^0)$.

Our analysis shows that both outcomes can arise under iterative retraining. The long-term behavior depends on three key factors: the growth rate of synthetic data, the verifier bias, and the verifier capability (i.e., its variance-reduction power). As retraining proceeds, the influence of the original data gradually diminishes, while the verifier increasingly shapes the estimator. Consequently, the estimator $\hat{\theta}^k$ converges toward a fixed point determined by the verifier's knowledge center θ_c.

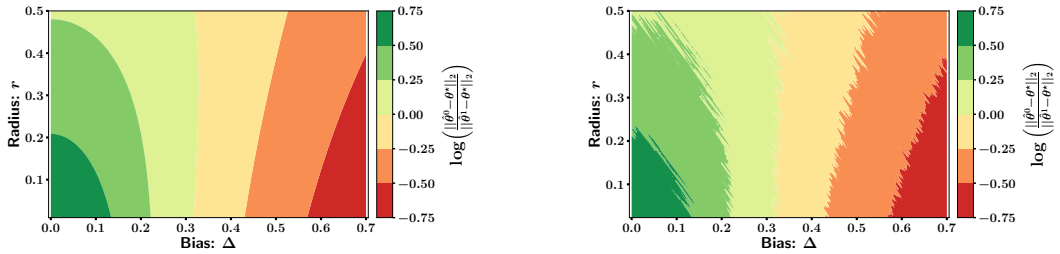
This leads to three regimes: **(1) Unbiased verifier:** if $\theta_c = \theta^\star$, iterative retraining yields sustained improvement and convergence to the true parameter. **(2) Mildly biased verifier:** small bias can produce short-term gains via variance reduction, but performance eventually plateaus or degrades as bias accumulates. **(3) Strongly biased verifier:** large bias leads to degradation and potential collapse. Among these, the mildly biased regime is most realistic in practice. While synthetic retraining may initially improve accuracy, an unbiased verifier is unlikely, and accumulated bias ultimately prevents sustained improvement.

The following theorem formally characterizes the long-term behavior of $\hat{\theta}^k$ under verifier-based synthetic retraining in linear regression.

Theorem 4.1. *There exist synthetic retraining processes (Algorithm 2) and some constant* $0 < \rho < 1$ *such that:*

$$\mathbb{E}\|\hat{\theta}^k - \theta_c\|^2 \leq \rho^{2k}\mathbb{E}\|\hat{\theta}^0 - \theta_c\|^2 + p\sigma^2 \sum_{j=0}^{k-1} \frac{\rho^{2(k-j)-1}}{n_j}. \tag{4}$$

where $n_1 \leq n_2 \leq \cdots \leq n_k \cdots$ *denote the number of verified synthetic samples per direction at each iteration. In particular, if* $\lim_{k\to\infty} n_k = \infty$*, then* $\lim_{k\to\infty} \mathbb{E}\|\hat{\theta}^k - \theta_c\|^2 = 0$.



(a) Loss Landscape Predicted by Theory

(b) Empirical Loss Landscape from Simulations

Figure 3: 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).

The proof of Theorem 4.1 is provided in Appendix D, utilizing concentration bounds and supermartingale inequalities to establish convergence. A key novelty of our analysis is introducing a Markov process perspective to characterize the estimator's evolution, showing that the verifier makes the transition mapping a contraction.

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.

5 Experiments

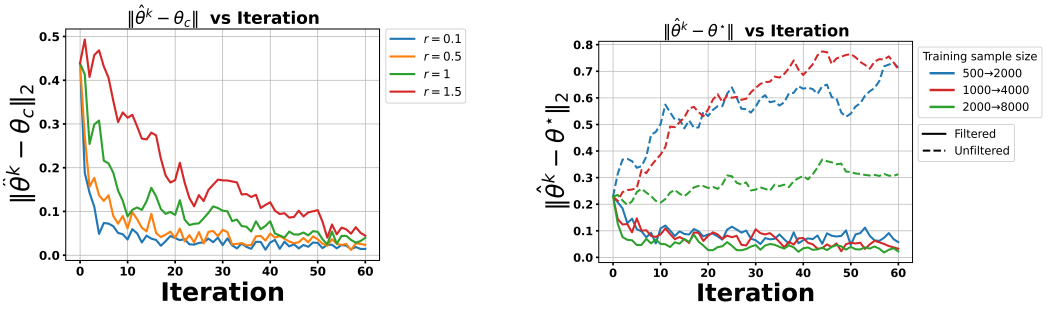
In this section, we evaluate our method across several experimental settings. We consider *a linear regression simulation* that directly reflects the theoretical assumptions, a *Variational Autoencoder (VAE)* trained on MNIST to study iterative retraining in an image-based generative model, and a large-scale news summarization task using a *pretrained language model*. Across all settings, the empirical results closely align with our theoretical predictions. Experimental code used to generate the results is publicly available at https://github.com/liuqiyuanhhh/Verified-Synthetic-Data.

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 3a shows that the error reduction predicted by Theorem 3.1 closely matches the empirical results in Figure 3b, validating the sharpness of our theoretical bounds. We set $\theta^\star = \mathbf{1}_8$ and define the verifier belief center as $\theta_c = \theta^\star + \Delta\mathbf{1}$, where Δ controls verifier bias, while the verification radius r determines filtering strictness. Using 100 real samples and 200 verified synthetic samples per singular direction, verifier-guided retraining outperforms the real-only baseline when the verifier bias is small (green region) and degrades when the bias is large (red region), empirically confirming the short-term bias–variance trade-off formalized in Theorem 3.1.

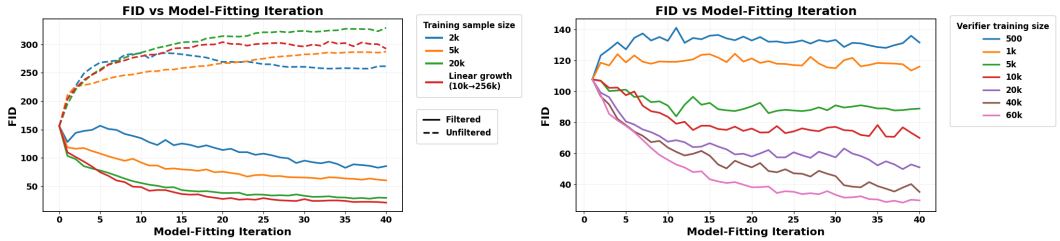
Iterative Synthetic Retraining. Similarly, Figure 4a confirms Theorem 4.1 by showing that, under a biased verifier, the retraining estimator converges to the verifier's knowledge center θ_c. In this experiment, the sample size increases linearly from 100 to 5500 over 60 rounds, with $\theta^\star = \mathbf{1}_8$ and $\theta_c = \theta^\star + 0.1\mathbf{1}$, and convergence is faster for smaller verification radii. Figure 4b repeats the experiment with an unbiased verifier ($\theta_c = \theta^\star$), where verifier-guided retraining consistently outperforms retraining without verification. These results empirically support the long-term contraction effect characterized in Theorem 4.1. Additional experiments in Appendix F.1 and Appendix F.2 demonstrate robustness to the synthetic data design and verifier shape.



(a) Verifier with bias

(b) Verifier without bias

Figure 4: Iterative synthetic retraining with and without bias.



(a) Filtering and training sample size.

(b) Verifier quality.

Figure 5: 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.

5.2 Conditional Variational Autoencoders (CVAE) on MNIST

We also evaluate our theory on real-world image generation tasks, extending beyond linear regression.

Setting. To expose the bias–variance trade-off and verifier-injection effects, we initialize the CVAE using only 500 real MNIST images, creating a low-resource regime where our theory predicts the largest gains from verifier-guided retraining. Additional results for varying initial sample sizes are presented in Appendix **??**. 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 that balances sample quality and diversity. The CVAE is then retrained on the retained synthetic samples, and this generate–filter–retrain procedure is repeated 40 rounds. Generative quality is evaluated using FID, and further implementation details are provided in Appendix E.

Results. Because our verifier provides feedback biased toward perceptual realism rather than likelihood calibration, we report **FID** as the primary metric and defer likelihood-based reconstruction metrics (ELBO) to Appendix E. Figure 5a shows FID across retraining iterations. With a strong verifier, we observe rapid FID improvement during early rounds, followed by saturation, whereas retraining without a verifier leads to severe degradation. Results using the MNIST-specific FID are consistent and reported in Appendix F.3. This behavior aligns with our theory: early improvements reflect the short-term bias–variance trade-off (Theorem 3.1), while long-term stability is governed by the contraction effect of verifier filtering (Theorem 4.1). The observed plateau reflects verifier limitations: a relatively simple verifier may overemphasize easily distinguishable patterns, introducing bias. For reference, a CVAE trained on all 60K real samples achieves an FID of 17.56, while the best verified synthetic model reaches 21.17 after 40 retraining iterations. Finally, Figure 5b studies the effect of verifier strength. Using a fixed synthetic batch size of 20K per round, stronger verifiers yield larger FID improvements, whereas weaker verifiers lead to early saturation or performance degradation. Qualitative results are shown in Figure 1.

5.3 Large-Scale News Summarization

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. (2025), 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. Following initial fine-tuning, we apply an iterative generate–filter–retrain procedure. In each iteration, the model generates synthetic summaries for the training corpus via greedy decoding. These summaries are scored using ROUGE-1 against ground-truth references, and the top 12.5% are selected as an oracle-filtered synthetic dataset. The model is then retrained on this subset, and test ROUGE-1 is recorded after each iteration.

Results. Figure 6 reports ROUGE-1 across 15 rounds of synthetic retraining for both filtered and unfiltered conditions, using the same number of synthetic examples. The filtered retraining procedure yields consistent, monotonic improvements during the early 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, indicating that synthetic retraining without quality control does not enhance performance. These results further demonstrate that our theory extends to large-scale settings, where synthetic retraining dynamics closely match our predictions.

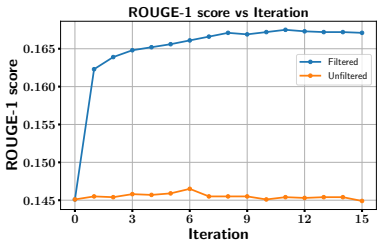


Figure 6: **ROUGE-1 score vs. iteration on the XSUM dataset.** Filtered retraining yields consistent early improvements, whereas unfiltered retraining shows no significant gain.

6 Discussion

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

Our analysis also has limitations. The theoretical framework relies on a well-specified parametric setting (linear regression) with a global ground-truth parameter θ^*. This abstraction compresses complex model qualities—such as diversity and generation quality—into a single distance metric. In practice, while models like LLMs are parametric, they approximate an unknown and likely non-parametric data-generating process, and a single "true model" may not exist. Although experiments on VAEs and LLMs qualitatively support our theory, extending the analysis to richer model classes (e.g., exponential families or simple neural networks) remains an important direction. Future work may also develop sharper guarantees for nonlinear models, explore alternative synthetic data design strategies beyond block orthogonalization, and study verifier dynamics in LLMs and vision models.

REFERENCES

Sina Alemohammad, Josue Casco-Rodriguez, Lorenzo Luzi, Ahmed Imtiaz Humayun, Hossein Babaei, Daniel LeJeune, Ali Siahkoohi, and Richard Baraniuk. Self-consuming generative models go mad. In *The Twelfth International Conference on Learning Representations*, 2024.

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.

Daniel Barzilai and Ohad Shamir. When models don't collapse: On the consistency of iterative mle. *arXiv preprint arXiv:2505.19046*, 2025.

Quentin Bertrand, Avishek Joey Bose, Alexandre Duplessis, Marco Jiralerspong, and Gauthier Gidel. On the stability of iterative retraining of generative models on their own data. In *ICLR*, 2024.

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. *Advances in Neural Information Processing Systems*, 37:46979–47013, 2024a.

Elvis Dohmatob, Yunzhen Feng, Pu Yang, Francois Charton, and Julia Kempe. A tale of tails: Model collapse as a change of scaling laws. In *International Conference on Machine Learning*, pp. 11165–11197. PMLR, 2024b.

Elvis Dohmatob, Yunzhen Feng, Arjun Subramonian, and Julia Kempe. Strong model collapse. In *The Thirteenth International Conference on Learning Representations*, 2025.

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

Damien Ferbach, Quentin Bertrand, Joey Bose, and Gauthier Gidel. Self-consuming generative models with curated data provably optimize human preferences. *Advances in Neural Information Processing Systems*, 37:102531–102567, 2024.

Shi Fu, Sen Zhang, Yingjie Wang, Xinmei Tian, and Dacheng Tao. Towards theoretical understandings of self-consuming generative models. In *International Conference on Machine Learning*, pp. 14228–14255. PMLR, 2024.

Shi Fu, Yingjie Wang, Yuzhu Chen, Xinmei Tian, and Dacheng Tao. A theoretical perspective: How to prevent model collapse in self-consuming training loops. In *The Thirteenth International Conference on Learning Representations*, 2025.

Anvit Garg, Sohom Bhattacharya, and Pragya Sur. Preventing model collapse under overparametrization: Optimal mixing ratios for interpolation learning and ridge regression. *arXiv preprint arXiv:2509.22341*, 2025.

Matthias Gerstgrasser, Rylan Schaeffer, Apratim Dey, Rafael Rafailov, Tomasz Korbak, Henry Sleight, Rajashree Agrawal, John Hughes, Dhruv Bhandarkar Pai, Andrey Gromov, et al. Is model collapse inevitable? breaking the curse of recursion by accumulating real and synthetic data. In *First Conference on Language Modeling (COLM)*, 2024.

Daya Guo, Qihao Zhu, Dejian Yang, Zhenda Xie, Kai Dong, Wentao Zhang, Guanting Chen, Xiao Bi, Y Wu, YK Li, et al. Deepseek-coder: When the large language model meets programming-the rise of code intelligence. *CoRR*, 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. In *The Eleventh International Conference on Learning Representations*, 2023.

Hengzhi He, Shirong Xu, and Guang Cheng. Golden ratio weighting prevents model collapse. *arXiv preprint arXiv:2502.18049*, 2025.

Ruifei He, Shuyang Sun, Xin Yu, Chuhui Xue, Wenqing Zhang, Philip Torr, Song Bai, and Xiaojuan Qi. Is synthetic data from generative models ready for image recognition? In *The Eleventh International Conference on Learning Representations*, 2023.

Shadi Iskander, Sofia Tolmach, Ori Shapira, Nachshon Cohen, and Zohar Karnin. Quality matters: Evaluating synthetic data for tool-using llms. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 4958–4976, 2024.

Joshua Kazdan, Rylan Schaeffer, Apratim Dey, Matthias Gerstgrasser, Rafael Rafailov, David L Donoho, and Sanmi Koyejo. Collapse or thrive: Perils and promises of synthetic data in a self-generating world. In *Forty-second International Conference on Machine Learning*, 2025.

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. *CoRR*, 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.

Rylan Schaeffer, Joshua Kazdan, Alvan Caleb Arulandu, and Sanmi Koyejo. Position: Model collapse does not mean what you think. *arXiv preprint arXiv:2503.03150*, 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.

Yonglong Tian, Lijie Fan, Phillip Isola, Huiwen Chang, and Dilip Krishnan. Stablerep: Synthetic images from text-to-image models make strong visual representation learners. *Advances in Neural Information Processing Systems*, 36:48382–48402, 2023.

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: Fine-grained synthetic data filtering for large language models through data-parameter resonance analysis. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 5414–5428, 2025.

Xiukun Wei and Xueru Zhang. Self-consuming generative models with adversarially curated data. In *Forty-second International Conference on Machine Learning*.

Alexander Wettig, Aatmik Gupta, Saumya Malik, and Danqi Chen. Qurating: Selecting high-quality data for training language models. In *Forty-first International Conference on Machine Learning*, 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, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. In *Advances in Neural Information Processing Systems*, volume 38, 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? In *Forty-second International Conference on Machine Learning, ICML*, 2025.

APPENDIX OVERVIEW

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

A RELATED WORK

Understanding and mitigating model collapse. Recent work shows that heavy reliance on synthetic data in iterative training can cause *model collapse*—the degradation of performance when a model is repeatedly retrained on its own synthetic outputs (possibly mixed with real data).[3] Empirical evidence supports this phenomenon: Shumailov et al. (2024) show that recursive training on unfiltered synthetic data induces distribution shift and mode collapse, while Dohmatob et al. (2025) find that even small synthetic proportions can harm performance. In linear settings, Dohmatob et al. (2024a) analyze collapse mechanisms explicitly, and Dohmatob et al. (2024b) connect degradation to altered neural scaling laws.

To mitigate collapse, prior work broadly explores three strategies. First, accumulating data or gradually increasing the synthetic dataset size across iterations can suppress noise and bound errors (Gerstgrasser et al., 2024; Dey & Donoho, 2024; Xu et al., 2025; Kazdan et al., 2025; Barzilai & Shamir, 2025). Second, mixing synthetic data with real data stabilizes retraining (Bertrand et al., 2024; Fu et al., 2024; 2025), as performance progressively degrades without sufficient fresh real data (Alemohammad et al., 2024). Recent studies have even derived optimal mixing ratios to maximize this stabilizing effect (He et al., 2025; Garg et al., 2025). Finally, algorithmic interventions, such as the token re-sampling procedures proposed by Zhu et al. (2025), offer alternative pathways to avoid collapse.

Unlike prior work that relies on unfiltered synthetic data, our framework incorporates an external verifier to remove low-quality samples. Such verifiers may be human annotators or stronger teacher models. Filtering is widely used in iterative retraining and has shown empirical success in preventing model degradation and even improving performance (He et al., 2023; Tian et al., 2023; Guo et al., 2024; Zelikman et al., 2022; Zhang et al., 2024; Lampis et al., 2023; Haluptzok et al., 2023; Patwa et al., 2024). Motivated by this, we develop a principled understanding of when improvement is possible—namely, whether a generative model can leverage the verifier's feedback, embedded in the selected synthetic subset, to achieve sustained gains.

Filtering and selecting synthetic data. While a rich line of empirical work demonstrates that these filtering strategies can improve model performance, theoretical understanding about iterative retraining with filtered synthetic data remains largely unexplored, with only a few recent exceptions. Amin et al. (2025) assume a strong, reliable quality function and focus on how an external labeler aids learning under this fixed filtering mechanism. Feng et al. (2025) study a classification problem and identify a sharp phase transition. However, modeling synthetic data merely as noisy labels abstracts away the structural dependencies between features and labels inherent to true generative processes. Finally, Ferbach et al. (2024); Wei & Zhang considers a conceptually similar problem of learning a discrete preference distribution from human feedback by using humans' preferred choices as a filtering strategy. In their population-level analysis, curating synthetic data via an external reward function forces the model distribution to converge to the highest-reward level set, maximizing expected reward but ultimately collapsing in diversity.

Similar to many of the aforementioned studies above, our theoretical analysis also focuses on linear models (Feng et al., 2025; Dohmatob et al., 2025; Gerstgrasser et al., 2024). However, our model allows inaccuracy of the verifier in terms of both bias and variance. Errors in the synthetic data primarily stem from the inaccuracy of the generative model itself rather than exogenous noise. We show that model's short-term performance varies smoothly with the verifier's bias, selectivity, and size of synthetic data, rather than exhibiting a sharp phrase transition from complete failure to perfect accuracy as in Feng et al. (2025). In the long run, the model's performance converges to the verifier's knowledge center whereas verifier's selectivity only affects convergence speed. Our results bridge

[3]There is no widely agreed-upon formal definition; see (Schaeffer et al., 2025) for discussion.

short-term and long-term perspectives of iterative retraining, illustrating how varied verifier qualities give rise to distinct performances of iterative retraining.

Comparison with reward maximization frameworks. While sharing the conceptual similarity of evaluating generated data via an external feedback mechanism, our modeling approach substantially differs from reward maximization frameworks in various aspects, including preference matching (Ferbach et al., 2024; Wei & Zhang) and recent Reinforcement Learning with Verified Rewards (RLVR)(Guo et al., 2024; Yu et al., 2025). Theoretically, these methods frame the problem as policy optimization, where the objective is to maximize a provided reward signal. While highly effective for alignment, the definition of a "good model" is tied to the specific reward formulation, which may not correspond to recovering the true data-generating distribution. Practically, reward optimization relies on assigning scalar rewards or pairwise comparison signals (Ouyang et al., 2022), which 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. While recent methods like RLVR avoid this issue by restricting themselves to domains with clearly verifiable rewards (Guo et al., 2025; Wu & Choi; Yu et al., 2025), many important training settings lack such reliable reward functions. In contrast, we study the widely used "generate-verify-retrain" paradigm, which utilizes binary accept/reject filtering. Theoretically, our framework defines a "good model" at the parameter level, explicitly modeling the relationship between the verifier's filtering rule and the ground truth. By formalizing this link, we can directly analyze model performance during iterative retraining, even for an imperfect or biased verifier. Practically, this filtering mechanism is less noisy, highly stable, and serves as a core scalable primitive in modern LLM pipelines like DeepSeek-Coder (Guo et al., 2025).

B MSE DECOMPOSITION

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 $\hat{\theta}^0$.

C 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 D.

C.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:

$$\begin{aligned}
\mathbb{E}[X'|\mu] &= \mu - \sigma \frac{\phi(\frac{b-\mu}{\sigma}) - \phi(\frac{a-\mu}{\sigma})}{\Phi(\frac{b-\mu}{\sigma}) - \Phi(\frac{a-\mu}{\sigma})} := \mu + \sigma m_1(\frac{a-\mu}{\sigma}, \frac{b-\mu}{\sigma}) \\
\mathrm{Var}(X'|\mu) &= \sigma^2 \left[1 - \frac{\frac{b-\mu}{\sigma}\phi(\frac{b-\mu}{\sigma}) - \frac{a-\mu}{\sigma}\phi(\frac{a-\mu}{\sigma})}{\Phi(\frac{b-\mu}{\sigma}) - \Phi(\frac{a-\mu}{\sigma})} - \left(\frac{\phi(\frac{b-\mu}{\sigma}) - \phi(\frac{a-\mu}{\sigma})}{\Phi(\frac{b-\mu}{\sigma}) - \Phi(\frac{a-\mu}{\sigma})} \right)^2 \right] \\
&:= \sigma^2 m_2(\frac{a-\mu}{\sigma}, \frac{b-\mu}{\sigma})
\end{aligned} \tag{8}$$

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

$$\alpha = \frac{a-\mu}{\sigma}, \quad \beta = \frac{b-\mu}{\sigma}, \tag{9}$$

the expression in equation 8 become:

$$\begin{aligned}
\mathbb{E}[Z'] &= m_1(\alpha, \beta) \\
\mathrm{Var}(Z') &= m_2(\alpha, \beta)
\end{aligned} \tag{10}$$

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:

$$\begin{aligned} m_3(\alpha,\beta) &:= \mathbb{E}(Z' - \mathbb{E}Z')^3 \\ &= -\frac{(\beta^2-1)\phi(\beta) - (\alpha^2-1)\phi(\alpha)}{(\Phi(\beta)-\Phi(\alpha))} - \frac{3(\phi(\beta)-\phi(\alpha))(\beta\phi(\beta)-\alpha\phi(\alpha))}{(\Phi(\beta)-\Phi(\alpha))^2} \\ &\quad - \frac{2(\phi(\beta)-\phi(\alpha))^3}{(\Phi(\beta)-\Phi(\alpha))^3}. \end{aligned} \tag{11}$$

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

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

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

$$\begin{aligned} &\left| \frac{1}{\sigma^2}\mathbb{E}(\bar{X}^1-\mu)^2 - \underbrace{\frac{m_2(\alpha,\beta)}{n_1}}_{\textit{Synthetic Variance}} - \underbrace{\left(m_1^2(\alpha,\beta) + \frac{m_2^2(\alpha,\beta) + m_3(\alpha,\beta)m_1(\alpha,\beta)}{n_0}\right)}_{\textit{Verification Bias+Variance}} \right| \\ &< K\left(\frac{1}{n_1 n_0^{1/3}} + \frac{1}{n_0^{3/2}}\right) \end{aligned} \tag{12}$$

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

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

$$\epsilon_1 := \frac{\bar{X}^0 - \mu}{\sigma}, \quad \epsilon_1 \sim \mathcal{N}(0, \frac{1}{n_0}). \tag{13}$$

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:

$$\begin{aligned} \mathbb{E}[(\bar{X}^1-\mu)^2] &= \mathbb{E}_{\bar{X}^0}\,\mathbb{E}_{\bar{X}^1|\bar{X}^0}\left[(\bar{X}^1-\mu)^2\right] \\ &= \mathbb{E}_{\bar{X}^0}\left[\mathrm{Var}(\bar{X}^1\,|,\bar{X}^0) + \left(\mathbb{E}[\bar{X}^1 \mid \bar{X}^0] - \mu\right)^2\right] \\ &= \sigma^2\mathbb{E}_{\bar{X}^0}\left[\frac{m_2(\alpha-\epsilon_1,\beta-\epsilon_1)}{n_1}\right] + \mathbb{E}_{\bar{X}^0}\left[(\bar{X}^0 - \mu - \sigma m_1(\alpha-\epsilon_1,\beta-\epsilon_1))^2\right] \\ &= \frac{\sigma^2}{n_1}\mathbb{E}_{\epsilon_1}\left[m_2(\alpha-\epsilon_1,\beta-\epsilon_1)\right] + \sigma^2\,\mathbb{E}_{\epsilon_1}\left[\left(m_1(\alpha-\epsilon_1,\beta-\epsilon_1)+\epsilon_1\right)^2\right] \end{aligned} \tag{14}$$

For the first term in 14, 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

$$|m_2(\alpha-\epsilon_1,\beta-\epsilon_1) - m_2(\alpha,\beta)| = |\epsilon_1| \cdot \left|m_2^{(1)}(\alpha-\xi,\beta-\xi)\right| < \frac{M_1}{n_0^{1/3}}, \tag{15}$$

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 14. 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:

$$m_1(\epsilon_1) = m_1(\alpha, \beta) - [1 - m_2(\alpha, \beta)]\,\epsilon_1 + \frac{1}{2} m_3(\alpha, \beta)\epsilon_1^2 + \frac{1}{6} m_1^{(3)}(\xi)\epsilon_1^3, \quad \text{for some } \xi \in (0, \epsilon_1), \tag{16}$$

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

$$\begin{aligned}
\mathbb{E}_{\epsilon_1}\left[(m_1(\alpha - \epsilon_1, \beta - \epsilon_1) + \epsilon_1)^2\right] &= \mathbb{E}\left(m_1(\alpha, \beta) + m_2(\alpha, \beta)\epsilon_1 + \frac{1}{2} m_3(\alpha, \beta)\epsilon_1^2 + \frac{1}{6} m_1^{(3)}(\xi)\epsilon_1^3\right)^2 \\
&= m_1^2(\alpha, \beta) + \frac{m_2^2(\alpha, \beta) + m_1(\alpha, \beta) m_3(\alpha, \beta)}{n_0} + \frac{3 m_3^2(\alpha, \beta)}{4 n_0^2} \\
&\quad + \mathbb{E}\left(m_1(\alpha, \beta) + m_2(\alpha, \beta)\epsilon_1 + \frac{1}{2} m_3(\alpha, \beta)\epsilon_1^2\right) \frac{m_1^{(3)}(\xi)}{3} \epsilon_1^3 \\
&\quad + \mathbb{E}\left(\frac{{m_1^{(3)}}^2(\xi)}{36} \epsilon_1^6\right).
\end{aligned} \tag{17}$$

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 17 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 14 is bounded by

$$\left|\mathbb{E}_{\epsilon_1}\left[(m_1(\alpha - \epsilon_1, \beta - \epsilon_1) + \epsilon_1)^2\right] - m_1^2(\alpha, \beta) - \frac{m_2^2(\alpha, \beta) + m_1(\alpha, \beta) m_3(\alpha, \beta)}{n_0}\right| < \frac{K}{n_0^{3/2}} \tag{18}$$

for some constant K.

Combining 14, 15, and 18 completes the proof.

□

C.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** $k = 0, 1, 2, \ldots$ **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 $\{X_i'^k : i = 1, 2, \ldots n_k\}$.
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 $\{\bar{X}^0, \bar{X}^1, \ldots \bar{X}^k, \ldots\}$, where the conditional distribution $p(\bar{X}^{k+1}|\bar{X}^k)$ is given by

$$p(\bar{X}^{k+1}|\bar{X}^k) : \bar{X}^{k+1} = \bar{X}^k + \sigma \frac{\sum_{i=1}^{n_k} \xi_i'^{k+1}}{n_k}, \qquad \xi_i'^{k+1} \text{ i.i.d } \sim \mathcal{N}_{trunc}(\frac{a - \bar{X}^k}{\sigma}, \frac{b - \bar{X}^k}{\sigma}) \tag{19}$$

The following theorem summarizes these findings:

Theorem C.2. *Let $\bar{X}^k$ be the Markov process determined by equation 19 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 C.2. Define

$$\epsilon_k = \frac{\bar{X}^k - \mu}{\sigma}, \tag{20}$$

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 9. Therefore, it suffices to consider the standardized process $\{\epsilon_k, k = 0, 1, 2, \ldots\}$. equation 19 can be standardized as:

$$\epsilon_{k+1} = \epsilon_k + \frac{\sum_{i=1}^{n_k} \xi_i'^{k+1}}{n_k}, \qquad \xi_i'^{k+1} \sim \mathcal{N}_{\text{trunc}}\left(\alpha - \epsilon_k, \beta - \epsilon_k\right), \tag{21}$$

For convenience, we shift the noise terms $\xi_i'^{k+1}$ in equation 21 to have mean zero. Therefore, we introduce

$$T_{\alpha,\beta}(x) := x + \mathbb{E}\big[Z \mid \alpha - x \leq Z \leq \beta - x\big], \qquad v_{\alpha,\beta}(x) := \mathrm{Var}\big(Z \mid \alpha - x \leq Z \leq \beta - x\big). \tag{22}$$

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

Therefore, equation 21 can be rewritten as

$$\epsilon_{k+1} = T_{\alpha,\beta}(\epsilon_k) + \eta_{k+1} \tag{23}$$

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 21. 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 23 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

$$x^* = T_{\alpha,\beta}(x^*) \implies x^* = x^* + \mathbb{E}(Z\big|\alpha - x^* \le Z \le \beta - x^*) \implies x^* = \frac{\alpha+\beta}{2}. \tag{24}$$

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

$$\mathbb{E}V_{k+1} \ \le \ \rho^2\,\mathbb{E}V_k \ + \ \frac{\rho}{n_k}\,. \tag{25}$$

Unrolling equation 25,

$$\mathbb{E}V_k \ \le \ \rho^{2k}\mathbb{E}V_0 \ + \ \rho\sum_{j=0}^{k-1}\frac{\rho^{2(k-1-j)}}{n_j}. \tag{26}$$

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 23 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,

$$\{\tau_M > K\} \subseteq \Big\{S_K \geq t_0 + K\, m(M) - M\Big\}. \tag{27}$$

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 27 gives, conditionally on $\mathcal{F}_0$,

$$\{\tau_M > K\} \subseteq \Big\{S_K \geq \frac{m(M)}{2} K\Big\}, \qquad \text{for all } K \geq K_0. \tag{28}$$

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

$$\mathbb{P}\left(S_K \geq t\right) \leq \exp\left(-\frac{1}{2} \min\{\frac{t^2}{\sum_{j=0}^{K-1} 1/n_j}, \frac{n_1 t}{2b(M)}\}\right) \leq \exp\left(-\frac{1}{2} \min\{\frac{n_1 t^2}{K}, \frac{n_1 t}{2b(M)}\}\right). \tag{29}$$

Use the tail bound equation 29 in equation 28, we have

$$\mathbb{P}\left(\tau_M > K \mid \mathcal{F}_0\right) \leq \mathbb{P}\left(S_K \geq \frac{m(M)}{2}K\right) \leq \exp\Big(-c(M)n_1K\Big) \tag{30}$$

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

$$\begin{aligned}\mathbb{P}\left(\tau_M > K\right) &= \mathbb{E}\left[\mathbb{P}\left(\tau_M > K \mid \mathcal{F}_0\right)\right] \\ &\leq \mathbb{E}\left[\exp\Big(-c(M)n_1K\Big)\mathbb{1}_{\{K>K_0\}}\right] + \mathbb{P}\left(K \leq K_0\right)\end{aligned} \tag{31}$$

Let $K \to \infty$ in equation 31, 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.

□

D PROOFS OF ALL THEOREMS IN SECTION 2

Given the orthogonality of $\{v_j\}$ in the block design equation 2, the OLS estimator decomposes into a set of one-dimensional problems, each estimating the coordinate of θ along direction v_j. Accordingly, the retraining procedure can be formalized 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 $Y^{k+1,j} = X^{k+1,j}\hat{\theta}^k + \xi^{k+1,j}$, where $\xi^{k+1,j} \sim \mathcal{N}(0, \sigma^2 I)$
    Apply verifier to each $(x_i^{k+1,j}, y_i^{k+1,j})$ and retain valid samples satisfying

$$|y_i^{k+1,j} - (x_i^{k+1,j})^\top \theta_c| \le r\|x_i^{k+1,j}\| + \sigma_c, \tag{32}$$

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

$$\hat{\theta}^{k+1,proj,j} = \bar{y}'^{k+1,j} \tag{33}$$

  end for
  Update overall estimator:

$$\hat{\theta}^{k+1} = \sum_{j=1}^{p} v_j \hat{\theta}^{k+1,proj,j} \tag{34}$$

end for

Proof of Theorem 3.1. We consider the one dimensional projection estimator of $\hat{\theta}^{1,proj,j}$ defined in equation 33. The filter condition equation 32 is equivalent to:

$$\begin{aligned} &|\sigma\xi_i^{1,j} + v_j^\top(\hat{\theta}^0 - \theta_c)| \le r + \sigma_c \\ \iff &y_i^{1,j} = \sigma\xi_i^{1,j} + v_j^\top\hat{\theta}^0 \in \left(-r - \frac{\sigma_c}{\sigma} + v_j^\top\theta_c, r + \frac{\sigma_c}{\sigma} + v_j^\top\theta_c\right). \end{aligned} \tag{35}$$

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

$$\begin{aligned} \alpha_j &:= \frac{-r - \sigma_c + v_j^\top(\theta_c - \theta^\star)}{\sigma}, \\ \beta_j &:= \frac{r + \sigma_c + v_j^\top(\theta_c - \theta^\star)}{\sigma}. \end{aligned} \tag{36}$$

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 C.1, there exists constant K_j depending only on α_j, β_j such that if $n_1 > n_0 \ge 100$,

$$\begin{aligned} &\left|\frac{1}{\sigma^2}\mathbb{E}(\hat{\theta}^{1,proj,j} - v_j^\top\theta^\star)^2 - \frac{m_2(\alpha_j,\beta_j)}{n_1} - \left(m_1^2(\alpha_j,\beta_j) + \frac{m_2^2(\alpha_j,\beta_j) + m_3(\alpha_j,\beta_j)m_1(\alpha_j,\beta_j)}{\mu_j^2}\right)\right| \\ &< K_j\left(\frac{1}{n_1 n_0^{1/3}} + \frac{1}{n_0^{3/2}}\right) \end{aligned} \tag{37}$$

will hold with probability at least $1 - \exp(-Ln_0^{1/3})$. m_1, m_2, m_3 are defined in equation 10 and equation 11. By equation **??**, 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

$$\sum_{j=1}^{p} \mathbb{E}(\hat{\theta}^{1,proj,j} - v_j^\top \theta^\star)^2 = \sum_{j=1}^{p} \mathbb{E}(v_j^\top \hat{\theta}^1 - v_j^\top \theta^\star)^2 = \mathbb{E}||V^\top(\hat{\theta}^1 - \theta^\star)||^2 = \mathbb{E}\|\hat{\theta}^1 - \theta^\star|^2. \quad (38)$$

Therefore, by summing over j on both sides of equation 37 and using simple union bound, we have

$$\left| \frac{1}{\sigma^2}\mathbb{E}||\hat{\theta}^1 - \theta^\star||^2 - \sum_{j=1}^{p} \left(\underbrace{\frac{m_{2,j}}{n_1}}_{\text{Synthetic Variance}} + \underbrace{m_{1,j}^2 + \frac{m_{1,j}m_{3,j} + m_{2,j}^2}{\mu_j^2}}_{\text{Verification Bias+Variance}} \right) \right| < K \left(\frac{1}{n_1 n_0^{1/3}} + \frac{1}{n_0^{3/2}} \right) \quad (39)$$

with $K = \max_j K_j$ and

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

□

The central observation to establish Theorem 4.1 is that the iterative retraining procedure equation **??** 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

$$\hat{\theta}^{k+1} = T(\hat{\theta}^k) + \eta_{k+1}, \quad (40)$$

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 40 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.

Proof of Theorem 4.1. We consider the transition dynamics of $\hat{\theta}^k$ in Algorithm 2. Since we designed $X^{k+1,j}$ to be the rank one matrix correspond to singular vector v_j, therefore equation 33 can be rewritten as

$$\hat{\theta}^{k+1,proj,j} = v_j^\top \hat{\theta}^k + \frac{\sigma}{n_k} \sum_{i=1}^{n_k} \xi_i'^{k+1,j} \quad (41)$$

where $\xi_i'^{k+1,j}$ is the truncated noise term after verification. By equation 32, we have

$$\xi_i'^{k+1,j} \text{ i.i.d} \sim \mathcal{N}_{trunc}\left(-\frac{r}{\sigma} - \frac{\sigma_c}{\sigma} - v_j^\top \frac{\hat{\theta}^k - \theta_c}{\sigma}, \frac{r}{\sigma} + \frac{\sigma_c}{\sigma} - v_j^\top \frac{\hat{\theta}^k - \theta_c}{\sigma} \right). \quad (42)$$

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 34, equation 41 can be standardized as

$$\epsilon_j^{k+1} = \epsilon_j^k + \frac{\sum_{i=1}^{n_k} \xi_i'^{k+1,j}}{n_k}, \qquad \xi_i'^{k+1,j} \text{ i.i.d} \sim \mathcal{N}_{trunc}\left(-\beta - \epsilon_j^k, \beta - \epsilon_j^k\right) \tag{43}$$

where $\beta = \frac{r}{\sigma} + \frac{\sigma_c}{\sigma}$. We note that equation 43 is exactly the same dynamics we consider in the proof of Theorem C.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 C.2, we known that there exists a constant $\rho < 1$ such that

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

This implies that

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

□

E 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 ELBO We also evaluate generative performance using the test negative ELBO, a standard likelihood-based loss metric for VAEs. To prevent overfitting the discriminator (i.e., the verifier) and ensure stability during the retraining cycles, we incorporate standard regularization techniques, specifically applying a dropout rate of 0.1 and label smoothing with a parameter of 0.05 when training the discriminator. To investigate the effect of synthetic data size n_k, we employ three linearly increasing sample size schedules. Starting with an initial CVAE trained on only 500 samples, we scale up the retraining size by adding 5K, 30K, or 50K synthetic samples per iteration, respectively. The models are retrained for 50 iterations until the test negative ELBO stabilizes.

Figure 7 reports the test negative ELBO over these 50 rounds. Consistent with our bias–variance analysis, we observe a clear improvement (a decrease in loss) in the early stages (up to roughly iterations 10–15). Furthermore, the trajectories reveal a critical dynamic: while larger synthetic size schedules significantly accelerate this early convergence, all three schedules ultimately plateau and converge to a similar negative ELBO value by iteration 50. This observation validates our theoretical framework: drawing more synthetic samples expedites the initial variance reduction phase, but the asymptotic performance limit is dictated by the verifier, not the volume of synthetic data. After the initial variance-reduction gains, the negative ELBO eventually reverses its trend and increases (deteriorates) as the model inevitably converges toward the verifier's knowledge center.

As discussed in the main text regarding our verifier's limitations, this knowledge center is demonstrably biased. For reference, across all three size schedules, the final retrained CVAEs at iteration 50 converge to a test negative ELBO of approximately 111. In contrast, a baseline CVAE trained on the entire 60K real image dataset attains a test negative ELBO of 92.12 (lower is better). Because our verifier emphasizes perceptual quality over likelihood-based reconstruction, the negative ELBO proves harder to improve than FID. As a result, even as the negative ELBO stagnates or worsens in later iterations, our retrained models continue to improve FID, achieving sharper, cleaner digits. We believe that deploying stronger verifiers with, e.g., diversity preservation capabilities could enable iterative retraining to further improve the negative ELBO.

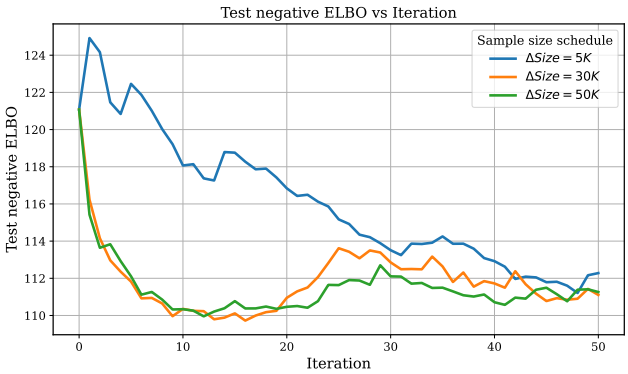


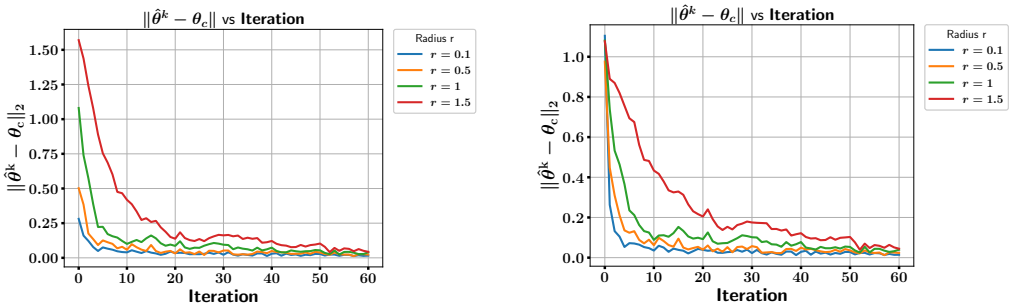
Figure 7: Test negative ELBO across retraining iterations.

F Additional Experimental Results

F.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 8 presents the results, corresponding directly to the two panels in Figure 4 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 8: Iterative synthetic retraining under random synthetic covariates, corresponding to the structured-design results in Figure 4.

F.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

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

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

(1) Ellipsoidal verifier. Consider the anisotropic ellipsoid

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

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

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

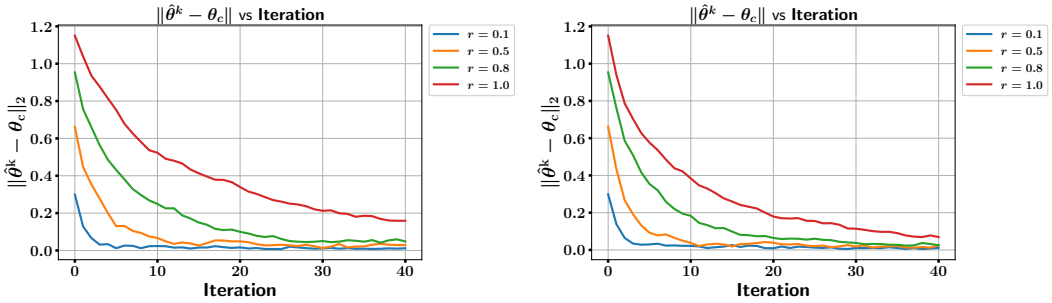
the perturbation satisfies $\|\Delta\|_1 \le r$. Using Hölder duality,

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

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 9.



(a) Ellipsoidal verifier

(b) Polyhedral ℓ_1 verifier

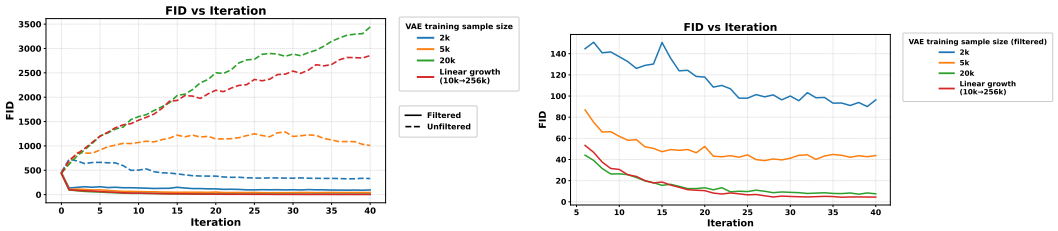
Figure 9: Retraining trajectories under two different verifier shapes. In both cases, $\hat{\theta}^{(k)}$ empirically converges toward the verifier center θ_c.

F.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 10a and 10b 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 10: **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.

G 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.