THE STATISTICS OF MODEL COLLAPSE: RESAMPLING, RETENTION, AND VARIANCE RECURSIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Model collapse — the degradation of generative models trained recursively on their own output (Shumailov et al., 2024) — is usually explained by properties of the neural networks involved. We show the dominant mechanism is instead statistical: plain kernel density estimation (KDE), repeatedly refit to its own samples, reproduces qualitative collapse with no neural network involved. Using this minimal setting, we adjudicate between two follow-up results that pull in opposite directions — Gerstgrasser et al. (2024)'s claim that *accumulating* synthetic data avoids collapse, and Dohmatob et al. (2025)'s claim that no synthetic fraction is safe — and show that *retention* of past data, not the injection rate, is what separates convergence from divergence. This retention effect survives sweeps over dimensionality, tail weight, and seven estimator families — including a normalizing flow, a GRU, and a causal self-attention Transformer — and reproduces on real images and real text-like sequences. We then derive *exact* variance recursions for a tractable Gaussian-MLE idealization of both regimes, verified against Monte Carlo simulation, and test these predictions directly against a small VAE trained by gradient descent: the accumulation-regime prediction matches closely, while under replacement the neural estimator collapses *faster* than the idealized statistical mechanism's own worst case. Neural networks, in other words, do not create model collapse so much as amplify a mechanism that operates identically in their complete absence.

1 INTRODUCTION

Shumailov et al. (2024) demonstrated that AI models collapse when trained on recursively generated data: feedback loops cause models to rely increasingly on lower-quality synthetic data, compounding errors across generations. The finding matters because high-quality human-generated training data is nearly exhausted, so future models may unavoidably be trained, at least in part, on the output of existing ones.

The search for an explanation has focused almost entirely on properties of the neural networks involved — overfitting to a model's own outputs, mode-seeking gradient descent, representation drift — which locates the cause inside the training procedure itself. We show this framing is incomplete: the same qualitative collapse, and in a tractable idealization the exact quantitative rate, arises from an estimator with no learned parameters, no gradient descent, and no training objective at all (a kernel density estimate, repeatedly refit to its own resamples). Neural networks are not required to produce collapse; where we can compare directly (Section 9), they do not even add a qualitatively new failure mode so much as amplify the same one — a real neural estimator collapses *faster* under replacement than the idealized statistical mechanism's own worst case, precisely because it lacks a guarantee (unbiasedness) that the statistical mechanism assumes. What looks like a training pathology is, at least in part, a resampling pathology that *any* recursively-refit estimator inherits, neural or not.

This reframing changes how recursive learning failures should be studied. Model collapse (Shumailov et al., 2024), self-consuming generative loops (Alemohammad et al., 2024), sample impoverishment in sequential Monte Carlo (Gordon et al., 1993), and iterative-retraining instability (Bertrand et al., 2024) are usually treated as separate phenomena tied to their respective architectures or algorithms. Our results suggest they are instances of one statistical mechanism: an estimator repeatedly refit to its own output accumulates sampling variance without bound unless some anchor to the original data distribution survives every refitting step. Under this view, the question to ask

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

of a new recursive training scheme is not "does this architecture have a failure mode" but "does this loop retain an anchor to un-collapsed data, and for how many steps does that anchor's influence persist" — a question our theory (Section 9) and leaky-memory experiment (Section 7) answer quantitatively, not just qualitatively.

We study the phenomenon in the simplest setting that exhibits it, and make four contributions:

1. **Collapse without neural networks (Section 3).** Repeatedly refitting a KDE to its own resamples reproduces model collapse's mode loss and divergence on 1-D mixtures, matching our variance-accumulation theory. Collapse is at root a statistical resampling phenomenon.
2. **Accumulation vs. strong collapse (Sections 4–6).** We reproduce both Gerstgrasser et al. (2024)'s accumulation result and Dohmatob et al. (2025)'s no-safe-fraction result in one nonparametric setting, then hold the injection rate α fixed while varying only retention. Retention, not injection rate, separates convergence from divergence, and a leaky-memory parameter shows it acts as a continuous knob, not a binary switch (Section 7).
3. **Generality (Section 8).** The retention effect survives sweeps over dimensionality, tail weight, and estimator family — including a neural density model (16 configurations) — and reproduces on real digit data and with two sequence architectures (GRU, causal Transformer), each showing similar retention-effect magnitude.
4. **An exact formal model, tested against a neural estimator (Section 9).** In a Gaussian-MLE idealization we derive exact variance recursions: replacement diverges, accumulation converges, for both mean and variance estimators, matching Monte Carlo simulation. A VAE trained on real images matches the accumulation prediction closely but collapses faster than the replacement prediction allows, pinpointing which assumption a real neural estimator violates.

2 Related Work

Shumailov et al. (2024) introduced the modern formulation of model collapse, demonstrating it in recursively-trained language models and analyzing recursive Gaussian estimation theoretically (Section 3.2). Gerstgrasser et al. (2024) challenged the replacement assumption: since the internet does not delete old content, a more realistic model has synthetic data accumulating alongside real data, and, analyzing a sequence of linear models, they prove accumulation keeps test error bounded while replacement lets it grow. Dohmatob et al. (2025) show that even a small synthetic fraction causes persistent test error under "strong model collapse" — there may be no safe mixing fraction, only a slower rate of degradation; Dohmatob et al. (2024) recast the same effect as a change in neural scaling laws. Seddik et al. (2024) model the mixing of real and synthetic text directly and derive a maximal safe synthetic fraction — a parametric, LLM-specific counterpart to the mixing-rate question we ask nonparametrically in Section 5.

The phenomenon spans modalities and model families: Alemohammad et al. (2024) show self-consuming image models degrade in quality or diversity without enough fresh real data each generation ("Model Autophagy Disorder"), and Briesch et al. (2024) observe the analogous loop in language models. Closest to our formal contribution, Bertrand et al. (2024) prove that iterative retraining has a stable fixed point near the true distribution given a sufficient real-data proportion — a parametric-learning counterpart to the retention effect we isolate nonparametrically.

This is not new in isolation: sample impoverishment in sequential Monte Carlo shows the same resampling-driven collapse of particle diversity three decades earlier (Gordon et al., 1993), consistent with our finding that no neural network is required. Two contemporaneous 2025 papers extend the picture: He et al. (2025) derive optimal weighting schemes for combining real and synthetic data across generations, complementary to our exact finite-sample recursions (Section 9) and leaky-memory interpolation (Section 7); Shi et al. (2025) reframe diffusion-model collapse as a generalization-to-memorization transition rather than variance inflation, echoing our own observation (Section 4) that collapse's signature differs across settings.

Our contribution is to reproduce both the accumulation and strong-collapse findings in one minimal, neural-network-free setting, disentangle the retention/injection-rate confound they leave open, and prove the underlying mechanism exactly.

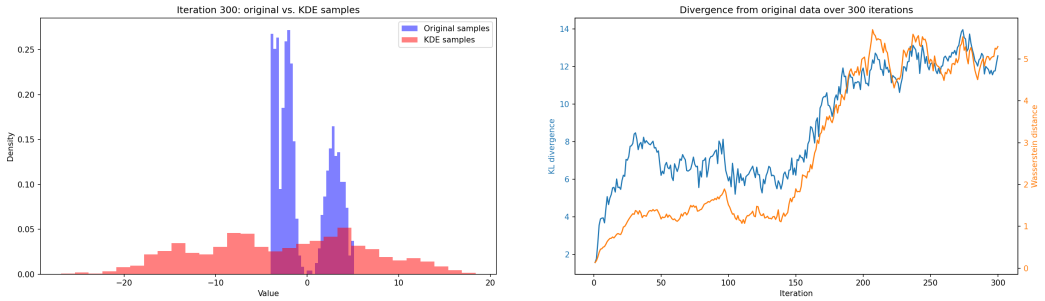

Figure 1: Recursive KDE fitting and sampling on a mixture of two Gaussians and a uniform component. Left: original distribution vs. KDE samples after 300 iterations. Right: KL divergence and Wasserstein distance to the original data over 300 iterations.

3 Model Collapse with Plain KDE Resampling

3.1 Mixture Experiments

We strip the phenomenon to its statistical core: fit a density to data with a Gaussian-kernel KDE, draw samples from the fit, refit on those samples, and repeat — the direct analogue of training a generative model on its own output, with no neural network anywhere. We apply this to three 1-D composite distributions: two Gaussians plus a uniform component; three Gaussians plus a uniform; and a Gamma, two Gaussians, and a uniform (Appendix D, Figures A3–A5; code in Appendix E). In every case, repeated resampling and refitting smooths away the original structure — the sharp uniform block, the separation between modes — and the samples collapse to a single broad, Gaussian-looking shape (Figure 1, left). With each iteration the divergence from the original data grows (Figure 1, right): KL divergence and Wasserstein distance both climb steadily and track each other closely throughout.

3.2 Connection to the Theory of Recursive Gaussian Fitting

This drift is what recursive parametric fitting predicts. For data $X^0 \sim \mathcal{N}(\mu, \sigma^2)$ refit each generation by sample mean and variance on M_i samples, Shumailov et al. (2024) show that

$$\frac{1}{\sigma^2}\operatorname{Var}(X_j^n) = \frac{1}{M_0} + \frac{1}{M_1} + \cdots + \frac{1}{M_{n-1}} + 1 + \mathcal{O}\left(M_i^{-2}\right), \tag{1}$$

so with constant $M_i = M$ the variance diverges linearly, $\operatorname{Var}(X_j^n) = \sigma^2(1 + n/M)$ with $\mathbb{E}(X_j^n) = \mu$, and the expected Wasserstein-2 distance to the truth grows with every step (full expressions in Appendix A). Our KDE simulation of a 1-D Gaussian matches this exactly: the mean of the generated samples stays near the original while their variance inflates and the Wasserstein distance climbs monotonically (Figure A6, Appendix D, alongside per-iteration snapshots). The behavior is insensitive to the KDE bandwidth (0.5 for the mixtures above, 0.1 here).

Collapse is thus reproduced by finite-sample resampling alone, independent of anything specific to neural models. The question becomes: under what conditions does this statistical process diverge?

4 Accumulation vs. Replacement

For the remaining experiments the ground truth p_0 is a two-component Gaussian mixture with component means ± 4 and unit variance (true variance 17.0).

We fit an initial KDE to $n_0 = 2000$ samples from p_0. At each generation $t = 1, \ldots, 25$, we draw $n = 500$ samples from the current KDE. Under *replacement*, the next KDE is fit only on these n new samples (the setting of Section 3 and Shumailov et al. (2024)); under *accumulation* (Gerstgrasser et al., 2024), on the union of all samples drawn so far (real and synthetic). We track $\mathrm{KL}(p_0 \| \hat{p}_t)$ on a fine grid, the sample variance, and the Wasserstein-1 distance to p_0, averaged over 15 trials.

Figure 2 shows a clean separation. Under replacement, KL and Wasserstein distance climb steadily and the variance inflates roughly $7\times$ beyond the true 17.0 by generation 25 (≈ 125). Under accumulation, all three metrics plateau near their true values (final KL 0.15 vs. 0.96; final variance 18.1 vs. 124.9). This confirms Gerstgrasser et al. (2024)'s result with a different estimator and toy problem, indicating their finding is not an artifact of the linear model they analyzed. Interestingly, collapse

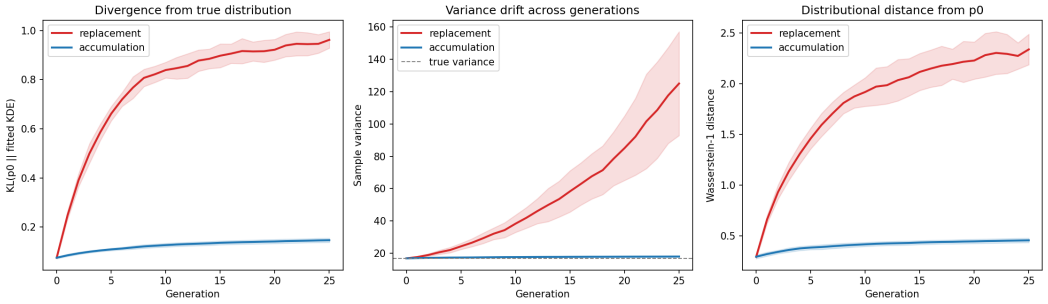

Figure 2: KL divergence, sample variance, and Wasserstein-1 distance from p_0 across 25 generations, replacement (red) vs. accumulation (blue). Shading: ± 1 std over 15 trials. Dashed line: true variance (17.0).

here manifests as *growing* variance rather than the tail-truncation emphasized by Shumailov et al. (2024); we attribute this to bandwidth-selection dynamics performing a random walk across generations — consistent with collapse being a general statistical phenomenon with several distinct failure modes rather than a single universal law.

5 Is There a Safe Injection Rate?

We next check Dohmatob et al. (2025)'s "strong model collapse" claim — persistent degradation from even a small synthetic fraction, no fully safe mixing rate — in the KDE setting, sweeping the real-data injection rate $\alpha \in \{0, \ldots, 1.0\}$ (10 values) under a forgetting-style regime (no retention across generations) over $T = 40$ generations, 12 trials each. Final KL decreases smoothly and monotonically in α, with no sharp threshold: even at $\alpha = 0.05$, variance is still climbing linearly at generation 40 — still diverging, merely more slowly (full sweep and figure in Appendix C). Under this forgetting regime, no fixed nonzero $\alpha < 1$ appears fully safe; smaller α only buys a slower rate of divergence.

6 Isolating Retention from Injection Rate

Sections 4 and 5 differ along two dimensions at once: the accumulation regime both retains all past data *and* accumulates a growing amount of real data, while the forgetting regime discards history regardless of α. Is it the *amount* of real data, or the *retention itself*, that prevents divergence?

We hold α and $T = 20$ fixed and compare two regimes fed *identical* real and synthetic batches of 300 samples at every generation (same seeds): *forget* fits the KDE only on the current batch; *accumulate* fits it on the union of all batches so far (capped at 4000 samples by random subsampling).

Table 1 and Figure 3 show a stark separation that is essentially insensitive to α: under accumulation final KL is 0.15–0.17 whether α is 0.02 or 0.1 — a fivefold range in real-data rate barely moves the outcome — while under forgetting it is 0.77–0.90 and never approaches the accumulation values. In this recursive KDE setting, *retention*, not injection rate, is the primary lever: a small amount of real data that is never forgotten anchors the fitted density indefinitely, while a larger amount that is discarded every generation still allows divergence. We do not claim this ordering holds under every mechanism that can degrade a recursively-trained model (e.g., optimization failures, catastrophic forgetting, or memorization dynamics specific to a particular architecture may interact with α and retention differently); within the resampling mechanism isolated here, though, it refines both prior claims — it is compatible with accumulation preventing collapse (Gerstgrasser et al., 2024) and with no forgetting-regime α being safe (Dohmatob et al., 2025) — while suggesting the deeper common mechanism is memory rather than mixing fraction.

7 Partial Retention: A Continuous Lever

Section 6 compared only the two endpoints — forgetting everything vs. retaining everything. The natural question is what lies between: is retention a binary switch, or a knob that trades off smoothly? We interpolate with a single *leaky-memory* parameter ρ: at each generation, every

Table 1: Final $\mathrm{KL}(p_0 \,\|\, \hat{p}_T)$ after $T = 20$ generations, holding α fixed and varying only retention.

	$\alpha = 0.02$	$\alpha = 0.05$	$\alpha = 0.10$
Forget	0.904	0.886	0.769
Accumulate	0.171	0.155	0.151

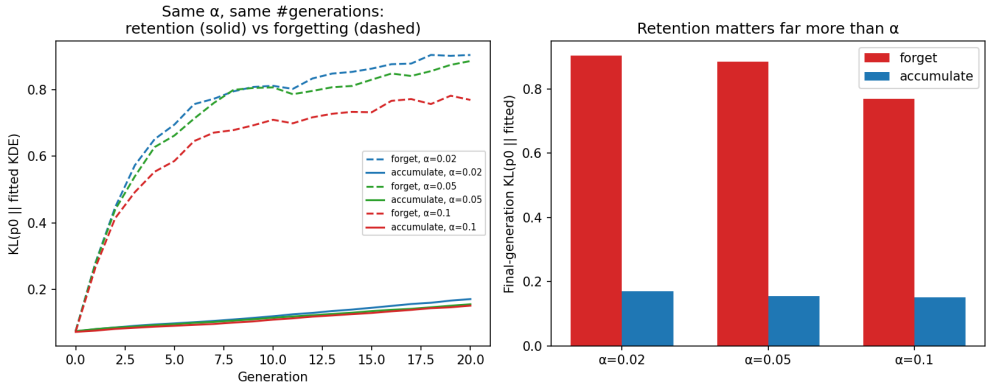

Figure 3: Left: KL for matched α under forgetting (dashed) vs. accumulation (solid), same batches every generation. Right: final KL by regime and α; retention changes the outcome far more than a $5\times$ change in α.

sample currently in the pool survives to the next generation independently with probability ρ, and the new batch is added on top. Thus a sample first seen at generation s survives to generation t with probability ρ^{t-s} — an exponential memory with half-life $-\ln 2/\ln \rho$ generations. The endpoints are exactly the two regimes of Section 6: $\rho = 0$ keeps only the new batch (forget) and $\rho = 1$ keeps the running union (accumulate).

Everything else matches Section 6: the two-component Gaussian-mixture ground truth (means ± 4, unit variance), $\alpha = 0.05$, $T = 20$, $n_0 = 2000$, batch 300/generation, pool capped at 4000. We sweep $\rho \in \{0, 0.3, 0.5, 0.7, 0.85, 0.95, 0.99, 1\}$, 8 trials each, and report final $\mathrm{KL}(p_0 \,\|\, \hat{p}_T)$. By construction the $\rho = 0$ and $\rho = 1$ endpoints reproduce Table 1's $\alpha = 0.05$ column, a built-in consistency check.

Figure 4 shows a smooth, monotonic descent from the forget level (final KL 0.87 at $\rho = 0$) to the accumulate floor (0.16 at $\rho = 1$), the endpoints matching Table 1 (0.886 and 0.155) within trial noise. Sample variance tracks the same path, falling monotonically from 74 (inflated $\approx 4.4\times$ over the true 17.0) at $\rho = 0$ to 17.8 at $\rho = 1$. Two things stand out. First, there is no sharp safe threshold in ρ — degradation is graded, echoing the thresholdless behavior we found for the injection rate (Section 5), but now in the *retention* control parameter. Second, the curve is markedly *concave*: the payoff is concentrated at high retention. A half-life of ≈ 4 generations ($\rho = 0.85$) still leaves final KL at roughly twice the accumulate floor, and only once the half-life reaches ≈ 14 generations ($\rho = 0.95$) does the outcome nearly saturate at the accumulate value. Retention prevents collapse only when past real data persists across *many* generations — consistent with the theory of Section 9, where anchoring requires the original real sample to remain in the pool long enough for the $1/N_t^2$ damping to accumulate. Retention is thus a continuous lever, not a binary switch, but one whose benefit is realized only near the high-retention end.

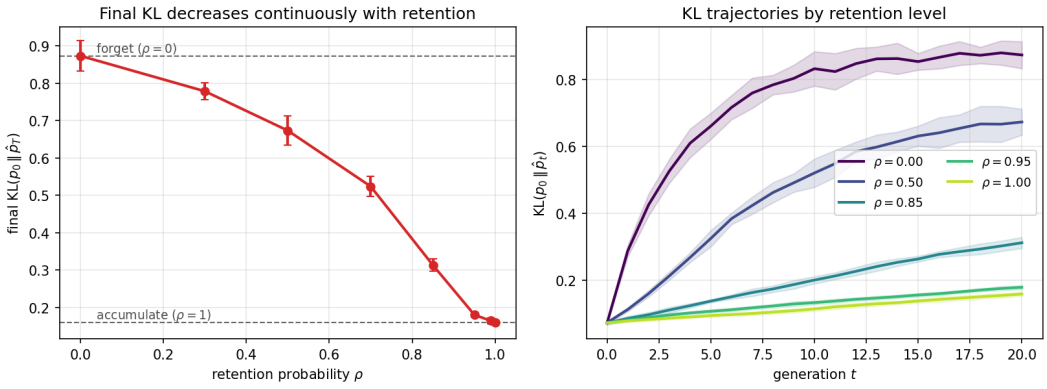

Figure 4: Partial retention via leaky memory (survival probability ρ). Left: final $\mathrm{KL}(p_0 \,\|\, \hat{p}_T)$ vs. ρ, error bars ± 1 std over 8 trials; dashed lines mark the forget ($\rho = 0$) and accumulate ($\rho = 1$) endpoints, which reproduce Table 1's $\alpha = 0.05$ column. The descent is smooth, monotonic, and concave. Right: KL trajectories over generations for representative ρ; higher retention slows and eventually bounds the growth.

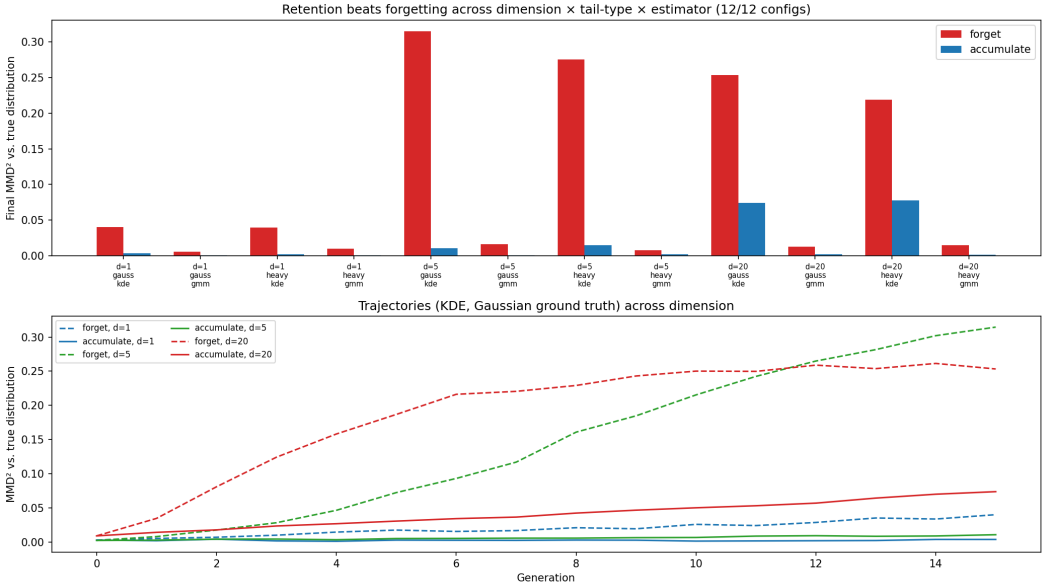

Figure 5: Top: final MMD^2 (forget vs. accumulate) across all 12 configurations (dimension × tail × estimator), $\alpha = 0.05$. Accumulate wins in every configuration. Bottom: representative trajectories (KDE, Gaussian ground truth).

8 Does Retention Generalize?

8.1 Synthetic Generalization: Dimension, Tail Weight, and Estimator Family

Dimension, tails, estimator family. We extend the ground truth to d-dimensional two-component mixtures (means ± 4 along the first coordinate, unit covariance), with Gaussian or heavy-tailed Student-t ($\mathrm{df} = 3$) components, fit by either a multivariate KDE or a parametric GMM (2 components, full covariance, EM). Since grid-based KL does not scale beyond $d = 1$, we use squared Maximum Mean Discrepancy (MMD^2; RBF kernel, median-heuristic bandwidth, unbiased estimator) against a held-out reference sample. We sweep $d \in \{1, 5, 20\}$ × both tails × both estimators (12 configurations), with $\alpha = 0.05$, $T = 15$, 5 trials each. Accumulation achieves lower final MMD^2 in *all twelve* configurations (Figure 5), with forget-to-accumulate ratios from $2.8\times$ ($d = 20$, Gaussian, KDE) to $28.9\times$ ($d = 5$, Gaussian, KDE). The advantage is smallest for KDE at $d = 20$ — consistent with the known degradation of nonparametric estimation in high dimensions — but never disappears or reverses, and the parametric GMM shows the same ordering everywhere, so the effect is not specific to nonparametric estimation.

8.2 Neural and Real-Data Generalization

A neural density estimator. KDE and GMM are both classical estimators; to check the effect is not an artifact of avoiding gradient-based fitting, we repeat the sweep with a small neural density model: a 4-layer RealNVP flow (affine coupling layers, hidden width 32), trained from scratch each generation by Adam on the NLL. Coupling layers need $d \geq 2$, so this covers $d \in \{5, 20\}\times$ both tails (4 configurations), $\alpha = 0.05$, $T = 12$, 4 trials, otherwise matching the KDE/GMM sweep. Accumulation wins in *all four* configurations (Figure A8, Appendix D), forget-to-accumulate ratios $7.1\times$–$24.7\times$ — if anything larger than the KDE/GMM ratios at the same dimensions. The retention effect is not contingent on closed-form or EM-based fitting; a trained neural network shows the same asymmetry.

A sequence-model bridge. Every estimator so far fits a density over $\mathbb{R}^d$ directly; model collapse was originally raised for autoregressive language models, so we test the closest analogue we can keep exact and fast: a small GRU language model (≈10k parameters), retrained from scratch each generation by teacher-forced cross-entropy. Ground truth p_0 is a fixed order-1 Markov chain over a 10-symbol alphabet, whose true transition matrix is known exactly, so we measure exact KL between it and the transition matrix estimated from the model's own generated sequences — the textual analogue of the KL-to-truth metric used for KDE/GMM. With $\alpha = 0.05$, $T = 12$, 4 trials, final KL

is 0.120 under forgetting vs. 0.046 under accumulation — a $2.6\times$ gap, smaller than the KDE/GMM/flow ratios but with clean, non-overlapping separation in every trial (Figure A9, Appendix D). The smaller margin is itself informative: a network retrained from scratch by stochastic optimization carries extra fit-to-fit noise that should work against a clean signal, yet the effect still survives.

A genuinely attention-based sequence bridge. The GRU is recurrent, not the causal self-attention architecture (Vaswani et al., 2017) that actually underlies modern LLMs; to check the effect is not somehow specific to recurrence, we repeat the sequence bridge with a tiny decoder-only Transformer (token and positional embeddings, causal self-attention blocks, a linear output head; $\approx$19k parameters), trained from scratch each generation by teacher-forced cross-entropy, on the identical ground truth, alphabet, and sweep parameters as the GRU above. Final KL is 0.082 under forgetting vs. 0.031 under accumulation — a $2.7\times$ gap, matching the GRU's $2.6\times$ almost exactly (Figure A10, Appendix D), though the Transformer fits the true transition matrix somewhat more closely in both regimes. Swapping recurrence for genuine self-attention leaves the retention effect's magnitude essentially unchanged.

A real-data bridge. We also repeat the comparison on `sklearn`'s digits dataset with a classical PCA+GMM density model — final MMD^2 is 0.0176 under forgetting vs. 0.0017 under accumulation, a $10.4\times$ gap (full setup, decoded samples, and sharpness analysis in Appendix C), ruling out that Section 6's result is an artifact of 1-D KDE on a Gaussian mixture. The genuinely neural version below repeats this real-image check with a trained generative model rather than a classical estimator.

A genuinely neural real-data bridge. The real-data bridge above still uses a classical estimator (GMM); to combine a real neural network trained by gradient descent with real images, we also train a small variational autoencoder (Kingma & Welling, 2014) from scratch each generation on the same digit images (encoder/decoder MLPs, latent dimension 6, Adam on the ELBO), forget vs. accumulate, $n_0 = 800$, $n = 300$ per generation, $T = 10, 20$ trials. Unlike the experiments above, we inject *no* additional real data after generation 0 ($\alpha = 0$), matching Section 9's idealization exactly so its predictions can be tested quantitatively rather than only invoked qualitatively (see below). Final MMD^2 is 0.176 under forgetting vs. 0.127 under accumulation (Figure A12, left, Appendix D) — the retention effect reproducing with a literal neural generative model on real images — the setting closest to the question the rest of this section only gestures at: does the mechanism identified in toy settings actually explain collapse quantitatively, not just qualitatively, in something closer to a modern generative model? Section 9 answers this directly.

9 Formal Model: Exact Variance Recursions for Gaussian MLE

The KDE is analytically inconvenient (its bandwidth is a complicated function of the fitting sample), so we analyze a fully tractable idealization — MLE fitting of a single Gaussian — for which *exact* recursions can be derived, mirroring Gerstgrasser et al. (2024)'s use of tractable linear regression but targeting the mean/variance-estimation setting closer to density estimation. All proofs are in Appendix B.

Setup. Let $p_0 = N(\mu_0, \sigma_0^2)$. At generation 0, fit $(\hat{\mu}_0, \hat{s}_0^2)$ (sample mean, Bessel-corrected variance) on n_0 i.i.d. draws from p_0. At each generation t, draw n fresh synthetic samples from $N(\hat{\mu}_{t-1}, \hat{s}_{t-1}^2)$ and refit — under **replacement** on only the n new samples; under **accumulation** on the pool of the n_0 real samples plus all synthetic batches so far ($N_t = n_0 + tn$).

Proposition 1 (Replacement diverges) *Under replacement,*

$$\mathrm{Var}(\hat{\mu}_t) = \frac{\sigma_0^2}{n_0} + t\,\frac{\sigma_0^2}{n}, \qquad \mathrm{Var}(\hat{s}_t^2) = \left(\frac{n+1}{n-1}\right)^t \left(\mathrm{Var}(\hat{s}_0^2) + \sigma_0^4\right) - \sigma_0^4.$$

Both expressions are strictly increasing and unbounded: the mean estimator's variance grows *linearly* in t (recovering the law of Section 3.2) and the variance estimator's own variance grows *geometrically* (rate $\frac{n+1}{n-1}$). The proof rests on $\{\hat{\mu}_t\}$ and $\{\hat{s}_t^2\}$ being martingales whose conditional sampling noise adds an *undamped* constant increment per generation.

Table 2: Exact recursions underlying Propositions 1 and 2.

	Replacement	Accumulation
$\mathrm{Var}(\hat{\mu}_t)$ recursion	$+\sigma_0^2/n$ (constant increment)	$+\,n\,E[\hat{s}_{t-1}^2]/N_t^2$ (shrinking increment)
Long-run behavior	$\to \infty$ (linear in t)	$\to$ finite limit

Proposition 2 (Accumulation: damped innovations) *Under accumulation,* $\hat{\mu}_t = \hat{\mu}_{t-1} + \frac{n}{N_t}\varepsilon_t$ *with* $\varepsilon_t \mid \mathcal{F}_{t-1}$ *of mean* 0 *and variance* $\hat{s}_{t-1}^2/n$*; consequently*

$$\mathrm{Var}(\hat{\mu}_t) \;\longrightarrow\; \frac{\sigma_0^2}{n_0} + \sigma_0^2 \sum_{i=1}^{\infty} \frac{n}{(n_0+in)^2} \;<\; \infty.$$

The innovation is damped by n/N_t, so the variance increments shrink as $1/N_t^2$ and their sum converges (terms $\sim 1/i^2$) — in sharp contrast to replacement's divergent constant-per-term series (Table 2). With $n_0 = 2000$, $n = 500$, $\sigma_0^2 = 1$, the limit is ≈ 0.000943, reached to within 7% by $t = 25$. An exact coupled recursion for $E[\hat{s}_t^2]$ (Appendix B) shows $E[\hat{s}_t^2] \to \sigma_0^2$.

Proposition 3 (Exact product form for $\mathrm{Var}(\hat{s}_t^2)$ under accumulation) *Let* $SS_t = (N_t - 1)\hat{s}_t^2$*. Then* $SS_t = SS_{t-1}K_t$ *with*

$$K_t := 1 + \frac{X_t}{N_{t-1}-1} + \frac{A_t}{n(N_{t-1}-1)}\,W_t, \qquad A_t = \frac{N_{t-1}n}{N_t},$$

where $X_t \sim \chi^2_{n-1}$ *and* $W_t \sim \chi^2_1$ *are the classical independent pivotal quantities of Gaussian sampling theory, independent across generations and of* SS_0*. Hence* $E[SS_t] = E[SS_0]\prod_{i\le t} E[K_i]$ *and* $E[SS_t^2] = E[SS_0^2]\prod_{i\le t} E[K_i^2]$ *are exact finite products of closed-form constants, giving* $\mathrm{Var}(\hat{s}_t^2)$ *exactly at every* t*.*

Because X_t and W_t are pivotal — their distributions do not depend on $(\hat{\mu}_{t-1}, \hat{s}_{t-1}^2)$ — no moment-closure approximation is needed, unlike a direct second-moment extension of the mean analysis. Iterating the product shows $\mathrm{Var}(\hat{s}_t^2)$ converges to a finite limit (≈ 0.00188 at $n_0 = 2000, n = 500, \sigma_0^2 = 1$): *both* first and second moments of the accumulation-regime estimator are provably bounded.

Numerical verification. Figure 6 confirms the recursions against direct Monte Carlo simulation of the generational process (4,000 trials replacement, 500 accumulation): theory and simulation agree within Monte Carlo noise at every generation, and extending to $t = 300$ confirms accumulation's plateau at the predicted limit vs. replacement's unbounded growth. Proposition 3's product formula is verified likewise (800 trials; Figure A11, Appendix D).

Testing the theory against a neural estimator. Propositions 1–2 are exact for Gaussian MLE fitting, but do they predict anything quantitative about a real neural network? We check directly, using the VAE of Section 8's real-data bridge ($\alpha = 0$, matching the propositions' assumption of no further real-data injection exactly). At each generation we draw a reference sample from the model and project it onto the top principal component of the real reference data (a direction fixed once, never refit), giving a scalar whose across-trial variance, $\mathrm{Var}(\hat{\mu}_t)$, and within-trial variance, $\hat{s}_t^2$, are the direct neural analogues of the theory's quantities. Because the VAE is not an exact unbiased fit — it measurably underdisperses even at generation 0, a known VAE failure mode unrelated to the collapse mechanism — calibrating the theory to the *true* real-data variance would conflate two different sources of error. Instead we calibrate $\mathrm{Var}(\hat{\mu}_0)$ and $\hat{s}_0^2$ to what the VAE itself measurably produced at generation 0, and let Propositions 1–2 evolve from there using only n_0 and n (both known exactly): a test of whether the recursion's *shape*, not the model's absolute fit quality, transfers.

Under **accumulation** the match is close: the theory predicts a $1.20\times$ growth in $\mathrm{Var}(\hat{\mu}_t)$ over 10 generations; the VAE shows $1.17\times$ (Figure A12, right, Appendix D). Under **replacement**, the theory predicts continued linear growth (a $9.26\times$ increase by generation 10); the VAE instead shows $\mathrm{Var}(\hat{\mu}_t)$ growing only to $1.68\times$ before plateauing (Figure A12, middle). The reason is visible in the model's own variance: $E[\hat{s}_t^2]$ collapses from 0.47 at generation 0 to ≈ 0.00002 by generation 4 — the VAE's decoder degenerates to a near point-mass within 2–3 generations, far faster than

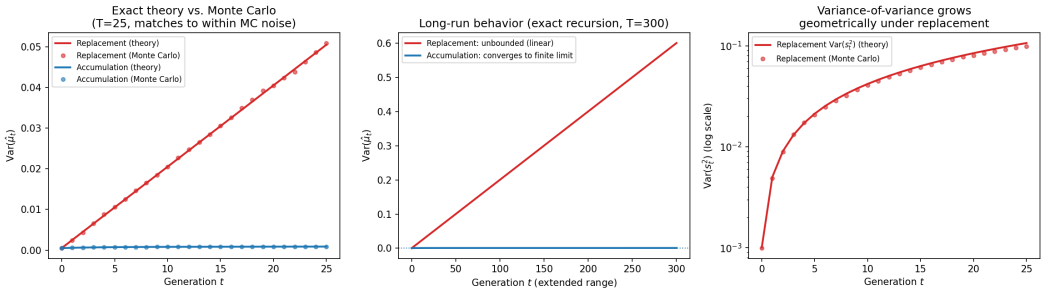

Figure 6: Left: exact theory (lines) vs. Monte Carlo (points) for $\mathrm{Var}(\hat{\mu}_t)$, both regimes, $t \leq 25$. Center: recursions extended to $t = 300$ — replacement grows linearly without bound, accumulation converges to the predicted limit (dotted). Right: $\mathrm{Var}(\hat{s}_t^2)$ under replacement grows geometrically (log scale), confirming Proposition 1.

Proposition 1's idealization allows. This is not the theory failing so much as it is diagnostic: Proposition 1's divergence rests on $\{\hat{s}_t^2\}$ being an *unbiased* martingale (exactly true for Gaussian MLE, by construction), which keeps feeding the mean estimator fresh variance to drift with, generation after generation. A neural density estimator has no such unbiasedness guarantee; its variance estimate is systematically biased downward, and that bias compounds multiplicatively across generations far faster than sampling noise alone would. Once the model collapses to a near point-mass, there is no variance left for $\hat{\mu}_t$ to drift with, so it freezes rather than keeps diverging — the idealized theory's own worst case is, in this sense, optimistic relative to what a real neural estimator can do under pure replacement.

Scope. This is an idealized proxy, not a literal derivation for the KDE experiments: bandwidth dynamics differ from Gaussian MLE and the mixture ground truth is not a single Gaussian, so we do not claim the specific rates or constants transfer quantitatively to KDE. Tested directly against a real neural network, however, the theory's predictions do transfer quantitatively under accumulation, and its qualitative direction (replacement drifts further than accumulation) holds under replacement too, even though the neural estimator's own additional variance bias makes it collapse faster than the idealization's worst case. What the closed-form result establishes rigorously either way is the mechanism: replacement's fitting noise compounds through an *undamped* martingale increment (divergent sum), accumulation's through a *damped* one shrinking as $1/N_t^2$ (convergent sum) — exactly the asymmetry observed with KDE, GMM, real-image density estimation, and now a real neural network trained by gradient descent, throughout this paper, and the same kind of exact-guarantee-in-idealization as Gerstgrasser et al. (2024)'s linear-regression theorem.

10 Discussion and Conclusion

Taken together, these results support the paper's contributions: collapse needs no neural network and matches recursive-fitting theory; the replacement/accumulation divide (Shumailov et al., 2024; Gerstgrasser et al., 2024) and the thresholdless degradation of Dohmatob et al. (2025) both reproduce in one minimal setting; *retention*, not injection rate, is the dominant lever, acting as a smooth knob with no sharp safe threshold; and the effect generalizes across dimensionality, tail weight, seven estimator families (three neural), real images, and two sequence architectures, and is proven exactly in a Gaussian-MLE idealization matching both simulation and a real VAE. The practical implication is that pipelines should not aim to keep the synthetic fraction below some threshold but should never discard real data, since a fixed ratio enforced fresh each run offers no protection and partial retention only helps once its half-life spans many training cycles. Three caveats: metric choice matters (shared bin edges are needed for KL to track Wasserstein reliably); our neural models are small, so specific constants may not transfer to LLM scale; and the VAE collapses *faster* than the idealization predicts under replacement, since it lacks the exact unbiasedness a Gaussian MLE has by construction, showing real estimators can be worse than the tractable worst case. Overall, the qualitative distinctions currently debated — replace vs. accumulate, safe mixing fractions, retention vs. injection rate — are recovered and proven within one statistical framework: the dominant mechanism is resampling, and neural networks amplify it rather than create it.

AI USE STATEMENT

In this work, we used generative AI tools for editing the manuscript for readability, formatting references, and drafting/editing code. We have not used generative AI tools to generate synthetic data, formulate mathematical claims, or write proofs. We also used generative AI tools to draft parts of the text and summarize the related literature. All AI-generated code and text were reviewed and verified by the authors. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

All experiments are small-scale and run on a single CPU, most in under a minute each; four are the exception, all retraining a small neural network from scratch every generation: Section 8's neural-flow sweep (4 configurations $\times$ 2 regimes $\times$ 4 trials $\times$ 13 refits) takes about 4 minutes, its GRU sequence-model bridge (2 regimes $\times$ 4 trials $\times$ 13 refits) takes about 7 minutes, its Transformer sequence-model bridge (2 regimes $\times$ 4 trials $\times$ 13 refits, no KV cache so autoregressive sampling recomputes causal self-attention over the whole growing prefix at every position) takes about 50 minutes, and Section 9's VAE-vs-theory check (2 regimes $\times$ 20 trials $\times$ 11 refits of a VAE) takes about 16 minutes. Every result corresponds to one self-contained script in the accompanying code package, organized one folder per section (Appendix E lists the Section 3 code in full), with all settings — sample sizes, number of generations T, injection rate α, trial counts, KDE/GMM/ flow structure, and the MMD^2 kernel heuristic — stated inline where used. Seeds are fixed throughout, and shared across regimes in the retention-confound experiment (Section 6) so the forget and accumulate arms see identical batches. The digits experiment uses the public `sklearn` digits dataset. Complete proofs of Propositions 1–3 are given in Appendix B.

ETHICS STATEMENT

This is a controlled statistical study using synthetic distributions and the public `sklearn` digits dataset — no human subjects, personal data, or direct harmful application. If anything, the results are constructive: showing that *retention* of real data, not just its mixing fraction, is what stabilizes recursive training speaks to data-provenance and archival practice. We make no claim that the specific rates or constants measured here transfer to large-scale deep generative models or LLMs (Section 10).

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. In *International Conference on Learning Representations (ICLR)*, 2024.

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 *International Conference on Learning Representations (ICLR)*, 2024. Spotlight.

Martin Briesch, Dominik Sobania, and Franz Rothlauf. Large Language Models Suffer From Their Own Output: An Analysis of the Self-Consuming Training Loop. *arXiv preprint arXiv:2311.16822*, 2024.

Elvis Dohmatob, Yunzhen Feng, Arjun Subramonian, and Julia Kempe. Strong Model Collapse. In *International Conference on Learning Representations (ICLR)*, 2025.

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 (ICML)*, 2024.

Matthias Gerstgrasser, Rylan Schaeffer, Apratim Dey, Rafael Rafailov, Henry Sleight, John Hughes, Tomasz Korbak, Rajashree Agrawal, Dhruv Pai, Andrey Gromov, Daniel A. Roberts, Diyi Yang,

David L. Donoho, and Sanmi Koyejo. Is Model Collapse Inevitable? Breaking the Curse of Recursion by Accumulating Real and Synthetic Data. In *First Conference on Language Modeling (COLM)*, 2024.

N. J. Gordon, D. J. Salmond, and A. F. M. Smith. Novel Approach to Nonlinear/Non-Gaussian Bayesian State Estimation. *IEE Proceedings F (Radar and Signal Processing)*, 140(2):107–113, 1993.

Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes. In *International Conference on Learning Representations (ICLR)*, 2014.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention Is All You Need. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Hengzhi He, Shirong Xu, and Guang Cheng. Recursive Learning Without Collapse: A Weighting-Based Stabilization Framework. *arXiv preprint arXiv:2502.18049*, 2025.

Mohamed El Amine Seddik, Suei-Wen Chen, Soufiane Hayou, Pierre Youssef, and Merouane Debbah. How Bad is Training on Synthetic Data? A Statistical Analysis of Language Model Collapse. *arXiv preprint arXiv:2404.05090*, 2024.

Lianghe Shi, Meng Wu, Huijie Zhang, Zekai Zhang, Molei Tao, and Qing Qu. A Closer Look at Model Collapse: From a Generalization-to-Memorization Perspective. *arXiv preprint arXiv:2509.16499*, 2025.

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

A WASSERSTEIN-2 DISTANCE UNDER RECURSIVE GAUSSIAN FITTING

For the recursive Gaussian fitting process of Section 3.2, Shumailov et al. (2024) give the expected Wasserstein-2 distance between the true distribution and the approximation at step $n+1$:

$$\mathbb{E}\left[\mathbb{W}_2^2\left(\mathcal{N}(\mu,\sigma^2),\mathcal{N}(\mu_{n+1},\sigma_{n+1}^2)\right)\right] = \frac{3}{2}\sigma^2\left(\frac{1}{M_0}+\frac{1}{M_1}+\cdots+\frac{1}{M_n}\right)+\mathcal{O}\left(M_i^{-2}\right), \tag{2}$$

$$\mathrm{Var}\left[\mathbb{W}_2^2\left(\mathcal{N}(\mu,\sigma^2),\mathcal{N}(\mu_{n+1},\sigma_{n+1}^2)\right)\right] = \frac{1}{2}\sigma^4\left(\frac{3}{M_0^2}+\frac{3}{M_1^2}+\cdots+\frac{3}{M_n^2}+\sum_{i\neq j}\frac{4}{M_iM_j}\right)+\mathcal{O}\left(M_i^{-3}\right) \tag{3}$$

which increases with each step n — a direct explanation of collapse in this simple model.[1]

B PROOFS

Proof of Proposition 1. Conditional on $(\hat{\mu}_{t-1}, \hat{s}_{t-1}^2)$, standard Gaussian sampling theory gives $\hat{\mu}_t \sim N(\hat{\mu}_{t-1}, \hat{s}_{t-1}^2/n)$ and $(n-1)\hat{s}_t^2/\hat{s}_{t-1}^2 \sim \chi_{n-1}^2$, independently. Both are therefore conditionally unbiased for the previous generation's parameter, so $\{\hat{\mu}_t\}$ and $\{\hat{s}_t^2\}$ are martingales with $E[\hat{\mu}_t] = \mu_0$ and $E[\hat{s}_t^2] = \sigma_0^2$ for all t (by induction). Applying the law of total variance to each gives the recursions $\mathrm{Var}(\hat{\mu}_t) = \mathrm{Var}(\hat{\mu}_{t-1}) + \sigma_0^2/n$ and $\mathrm{Var}(\hat{s}_t^2) = \frac{n+1}{n-1}\mathrm{Var}(\hat{s}_{t-1}^2) + \frac{2\sigma_0^4}{n-1}$ (using $\mathrm{Var}(\chi_{n-1}^2) = 2(n-1)$ and $E[\hat{s}_{t-1}^4] = \mathrm{Var}(\hat{s}_{t-1}^2) + \sigma_0^4$); solving both linear recursions in closed form gives the stated expressions. ■

Proof of Proposition 2. Write $\hat{\mu}_t = (n_0\bar{X}_{\mathrm{real}} + n\sum_{i=1}^t \bar{Y}_i)/N_t$, where $\bar{X}_{\mathrm{real}}$ is the fixed original real-data sample mean and $\bar{Y}_i$ is generation i's synthetic-batch mean. Then $N_t\hat{\mu}_t = N_{t-1}\hat{\mu}_{t-1} + n\bar{Y}_t$, and since $\bar{Y}_t = \hat{\mu}_{t-1} + \varepsilon_t$ with $\varepsilon_t \mid \mathcal{F}_{t-1} \sim N(0, \hat{s}_{t-1}^2/n)$, substituting and dividing by N_t gives the recursion. Taking variances,

$$\mathrm{Var}(\hat{\mu}_t) = \mathrm{Var}(\hat{\mu}_{t-1}) + \frac{n\,E[\hat{s}_{t-1}^2]}{N_t^2}.$$

Jointly with the exact identity $E[\hat{s}_t^2] = \big(E[Q_t] - N_t\mathrm{Var}(\hat{\mu}_t)\big)/(N_t-1)$, where $Q_t = \sum_{x\in\mathrm{pool}_t}(x-\mu_0)^2$ satisfies $E[Q_t] = E[Q_{t-1}] + n\big(E[\hat{s}_{t-1}^2] + \mathrm{Var}(\hat{\mu}_{t-1})\big)$, this gives a closed, exactly-solvable coupled system. Numerically iterating it shows $E[\hat{s}_t^2] \to \sigma_0^2$ and $\mathrm{Var}(\hat{\mu}_t)$ converges to the stated limit, a convergent series with terms $\sim 1/i^2$. With $n_0 = 2000$, $n = 500$, $\sigma_0^2 = 1$, the limit ≈ 0.000943 is reached to within 0.1% by generation $t \approx 300$. ■

Proof of Proposition 3. Directly extending the Q_t-based approach to second moments requires an ever-growing hierarchy of coupled moments ($E[\hat{\mu}_t^4]$, $\mathrm{Cov}(Q_t, (\hat{\mu}_t-\mu_0)^2)$, …) that does not close at any finite order; the following decomposition avoids this entirely. The pooled-variance combination identity gives $SS_t = SS_{t-1} + \hat{s}_{t-1}^2 X_t + A_t\varepsilon_t^2$, where $X_t = (n-1)\times$(new batch's own sample variance)$/\hat{s}_{t-1}^2 \sim \chi_{n-1}^2$ governs the new batch's within-batch spread and $W_t := n\varepsilon_t^2/\hat{s}_{t-1}^2 \sim \chi_1^2$ its mean's deviation from $\hat{\mu}_{t-1}$ (Cochran's theorem). Because X_t and W_t are pivotal — their distributions do not depend on the conditioning values $(\hat{\mu}_{t-1}, \hat{s}_{t-1}^2)$ at all — they are independent not just of $\mathcal{F}_{t-1}$ conditionally, but unconditionally, and independent across generations (each generation draws entirely fresh synthetic samples). Substituting $\hat{s}_{t-1}^2 = SS_{t-1}/(N_{t-1}-1)$ turns the additive identity into the multiplicative one. Since $\{K_t\}$ are independent (not identically distributed — each depends on t through N_{t-1}) and independent of SS_0, the expectation products follow, with $E[K_i]$ and $E[K_i^2]$ available in closed form from χ^2 moments, and $\mathrm{Var}(\hat{s}_t^2) = \big(E[SS_t^2] - E[SS_t]^2\big)/(N_t-1)^2$ exactly at every t. ■

[1] See also `https://en.wikipedia.org/wiki/Model_collapse`.

C Extended Results Deferred from the Main Text

Two results are summarized briefly in the main text (Sections 5, 8) with full detail here, since both repeat the qualitative conclusion of an earlier result with a different setup rather than adding a new qualitative finding.

C.1 Is There a Safe Injection Rate? (Full Sweep)

Setup. Does Dohmatob et al. (2025)'s "strong model collapse" — persistent degradation from even a small synthetic fraction — hold in the KDE setting, and is there a critical injection rate α? Each generation's batch of $n = 500$ has αn fresh real samples from p_0 and $(1 - \alpha)n$ synthetic samples from the current KDE; no data is retained across generations, matching Dohmatob et al.'s forgetting-style framing. We sweep $\alpha \in \{0, 0.01, 0.02, 0.05, 0.1, 0.2, 0.3, 0.5, 0.7, 1.0\}$ over $T = 40$ generations, 12 trials each.

Result. Figure A1 shows no sharp threshold: final KL decreases smoothly and monotonically in α. Even at $\alpha = 0.05$, variance is still climbing linearly at generation 40 (slope over the last five generations: 16.6, vs. 125.2 at $\alpha = 0$) — still diverging, merely more slowly. Under a forgetting regime, no fixed nonzero $\alpha < 1$ appears fully safe; smaller α only buys a slower rate of divergence.

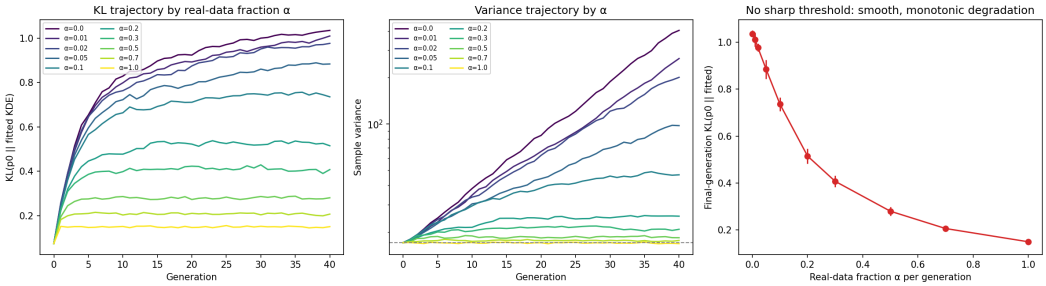

Figure A1: Left: KL trajectories over 40 generations for injection rates $\alpha \in [0, 1]$, forgetting regime. Center: variance trajectories (log scale) — none plateau for $\alpha < 1$. Right: final KL vs. α — smooth and monotonic, no visible threshold.

C.2 A Real-Data Bridge with a Classical Estimator (Full Setup)

Setup. We repeat the forget-vs-accumulate comparison on `sklearn`'s digits dataset (1,797 real 8×8 handwritten-digit images, lightly dequantized), reduced to 30 principal components (95.7% of variance) with a 15-component diagonal GMM as the density model ($\alpha = 0.05$, $T = 15$, 5 trials; 400 held-out images as the MMD^2 reference).

Result. Final MMD^2 is 0.0176 under forgetting vs. 0.0017 under accumulation — a $10.4\times$ gap. Decoded samples (Figure A2) and a pixel-gradient sharpness proxy ($7.16 \to 7.17$ forget, $7.16 \to 7.04$ accumulate) indicate the divergence manifests as a shift in which digit shapes are represented rather than as blurring. Though a small dataset and a shallow density model, this is a real-image-statistics setting, and the retention effect reproduces there as well.

D Additional Figures

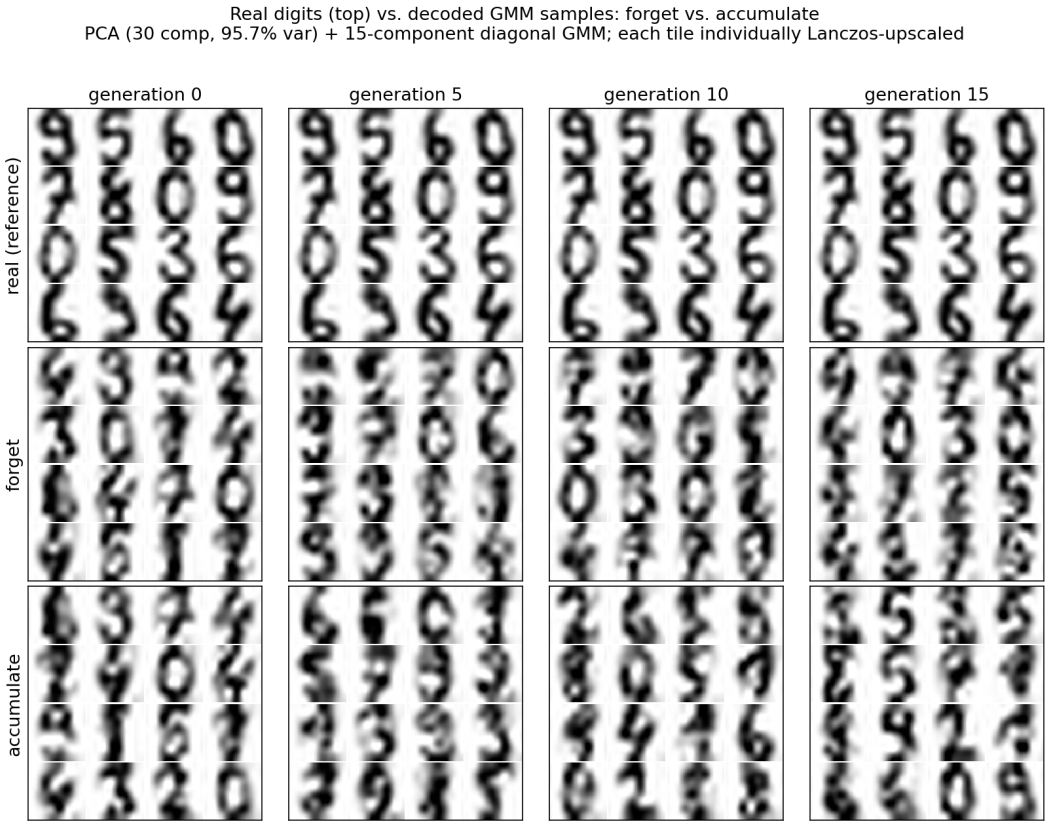

Figure A2: Digits experiment (Section 8): samples decoded via inverse PCA from the fitted GMM (30 PCs, 95.7% variance; 15-component diagonal GMM) at generations 0, 5, 10, and 15. Top: a fixed set of real reference digits (identical across columns) as an anchor; middle: forgetting; bottom: accumulation. Each 8×8 tile is upscaled individually with Lanczos interpolation before compositing, avoiding the cross-tile bleeding of a naively interpolated stitched grid. The divergence manifests as a drift in which digit shapes are represented rather than as blurring (Section 8).

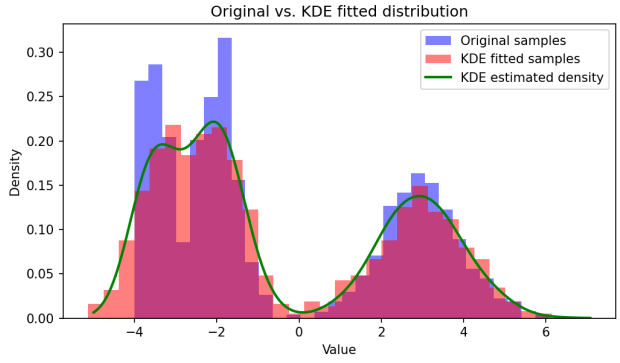

Figure A3: A synthetic distribution consisting of two Gaussian components and a uniform component: original samples, fitted KDE density, and samples generated from the KDE (Section 3.1).

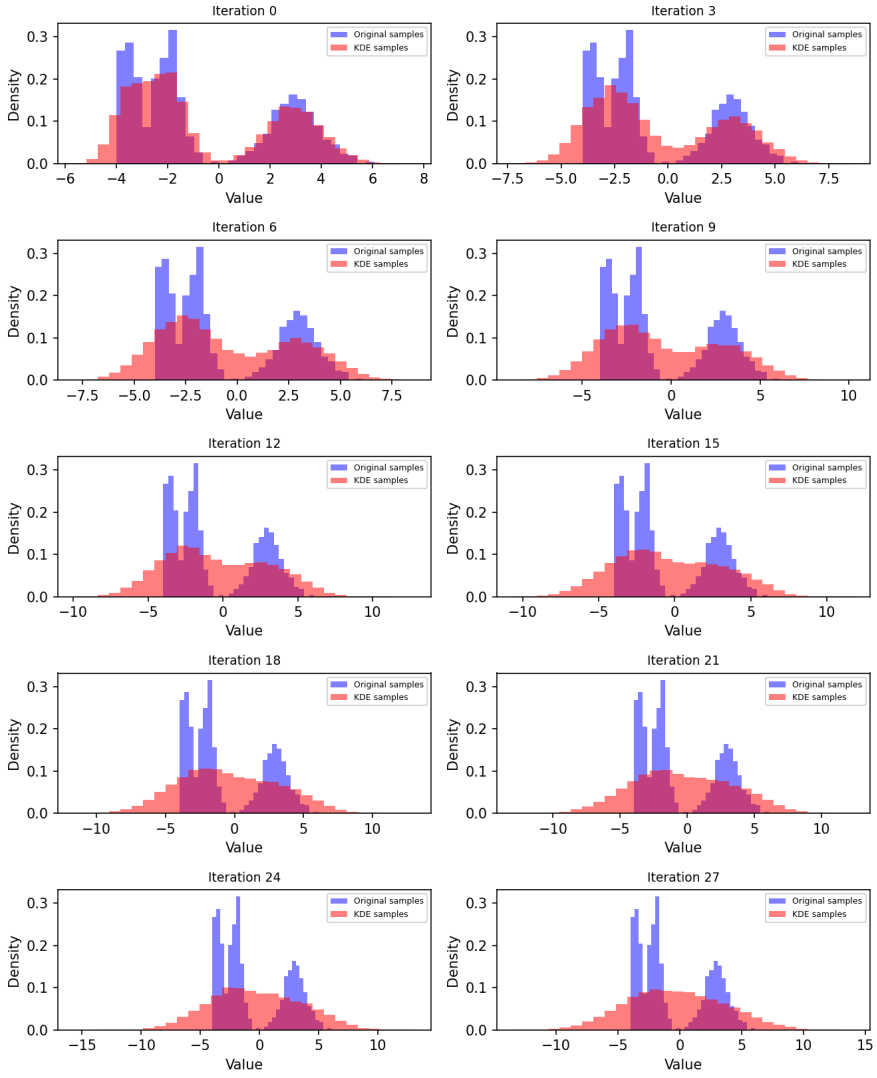

Figure A4: Recursive KDE fitting and sampling over 30 iterations, shown in steps of 3, for the distribution of Figure A3. The original structure (two modes plus a uniform block) is progressively smoothed into a single broad unimodal shape.

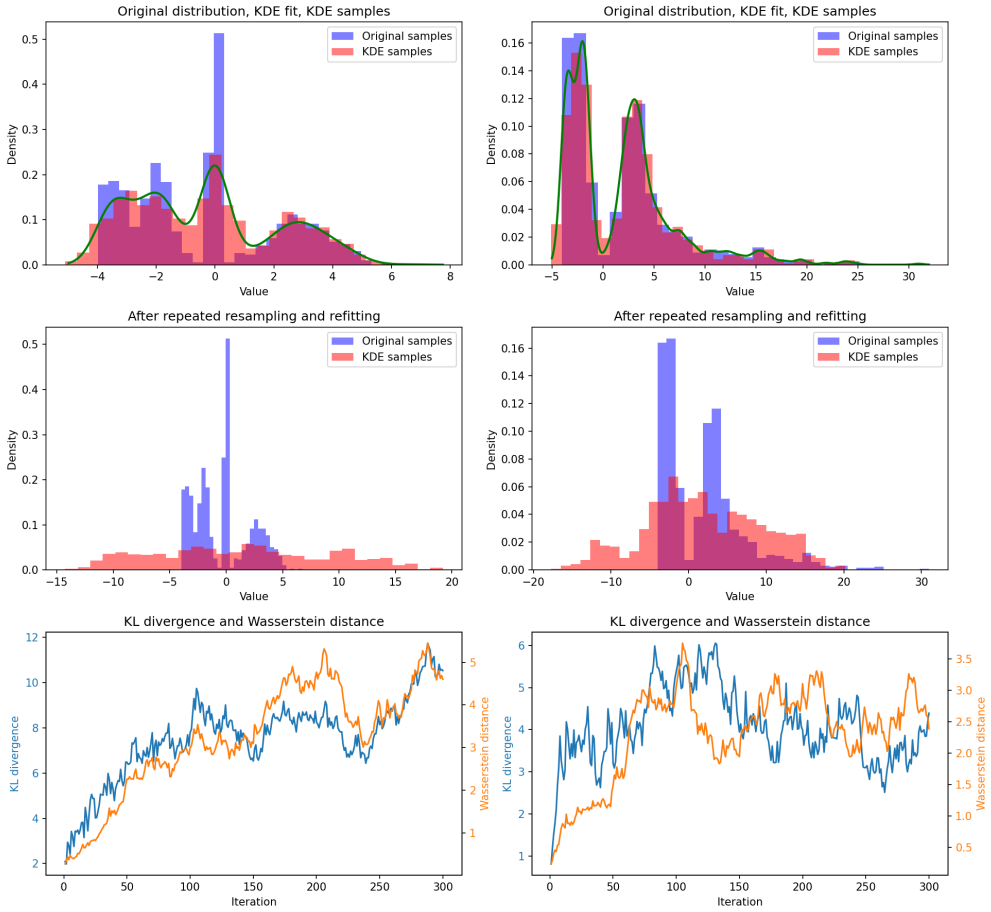

Figure A5: Two additional composite distributions (one per column; see Section 3.1). Top row: original distribution, fitted KDE, and samples generated from the KDE. Middle row: after repeated resampling and refitting, the samples exhibit a single mode and resemble a Gaussian. Bottom row: KL divergence and Wasserstein distance over 300 iterations.

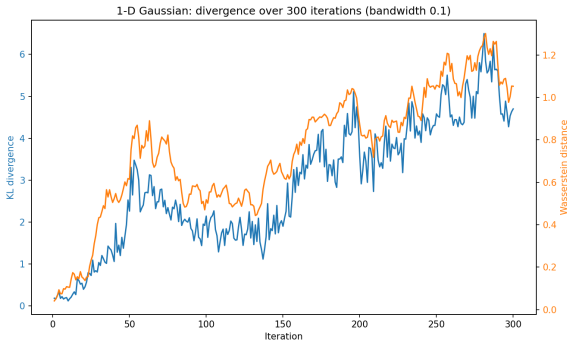

Figure A6: Recursive KDE fitting on a 1-D Gaussian (bandwidth 0.1), corresponding to Section 3.2: KL divergence and Wasserstein distance over 300 iterations. The sample mean stays consistent with the original while the variance inflates (Figure A7, below).

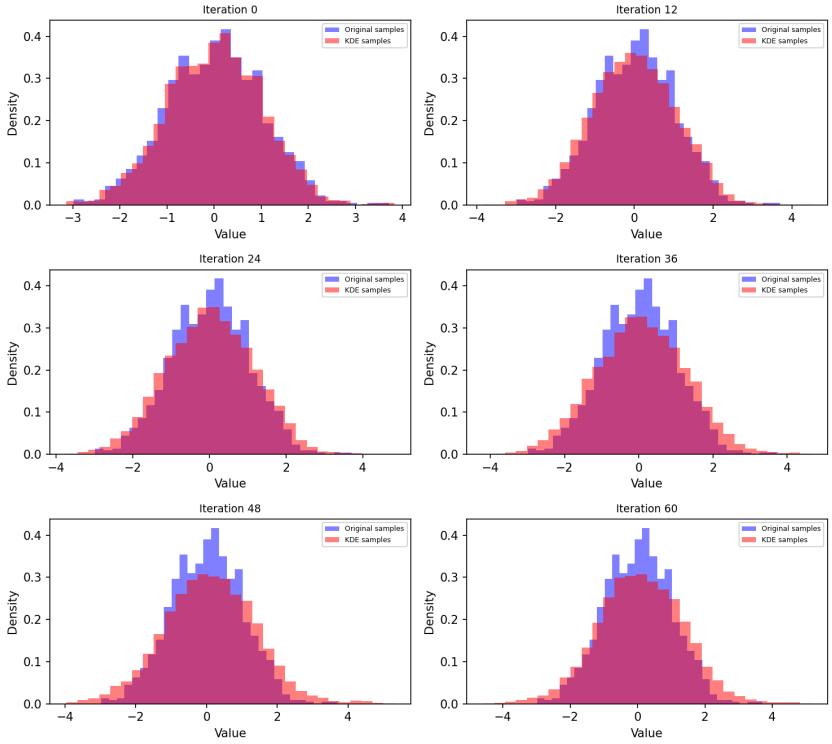

Figure A7: A 1-D Gaussian distribution and samples from the recursively KDE-fitted distribution (bandwidth 0.1) over 64 iterations, in steps of 12, corresponding to Figure A6.

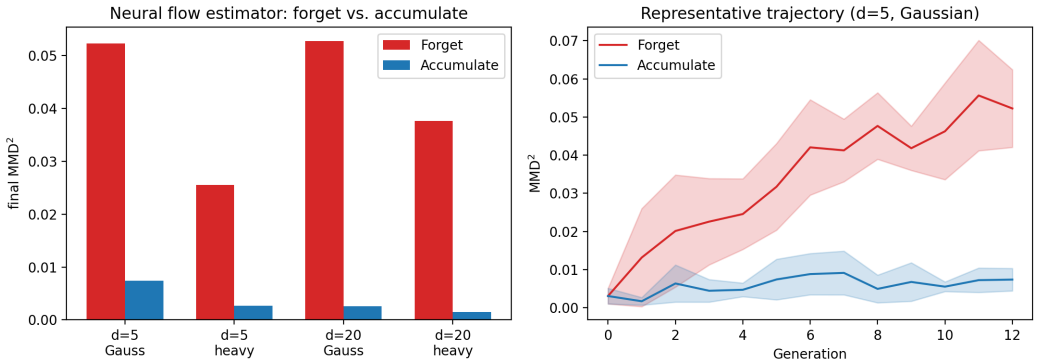

Figure A8: Neural density estimator (Section 8): 4-layer RealNVP flow, trained by Adam on NLL each generation. Final MMD^2, forget vs. accumulate, across all 4 configurations (left), and a representative trajectory ($d = 5$, Gaussian, right; shading ± 1 std over 4 trials). Accumulate wins in every configuration, matching the KDE/GMM pattern of Figure 5.

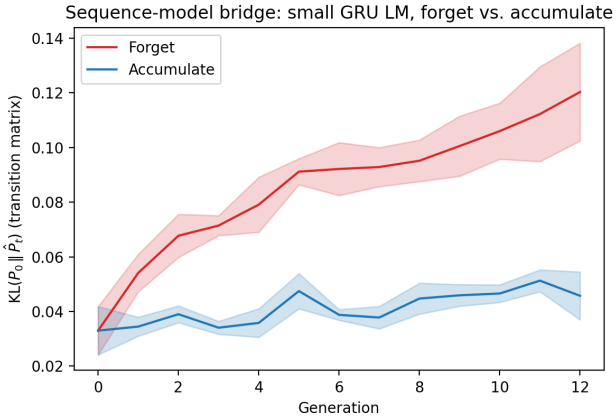

Figure A9: Sequence-model bridge (Section 8): a small GRU language model, retrained from scratch each generation, forget vs. accumulate. y-axis is exact KL between the true order-1 Markov transition matrix and the transition matrix estimated from the model's own generated sequences. Shading: ± 1 std over 4 trials.

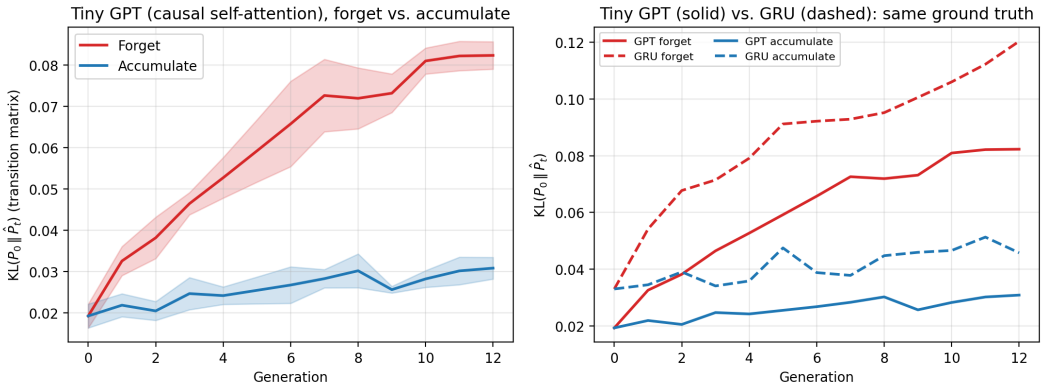

Figure A10: Attention-based sequence-model bridge (Section 8): a tiny decoder-only Transformer ($\approx$19k parameters, causal self-attention, no recurrence), retrained from scratch each generation, forget vs. accumulate (left; shading ± 1 std over 4 trials). Right: the same curves overlaid against Figure A9's GRU result (identical ground truth and sweep parameters) — forget-to-accumulate ratio $2.7\times$ (Transformer) vs. $2.6\times$ (GRU), essentially unchanged by swapping recurrence for self-attention.

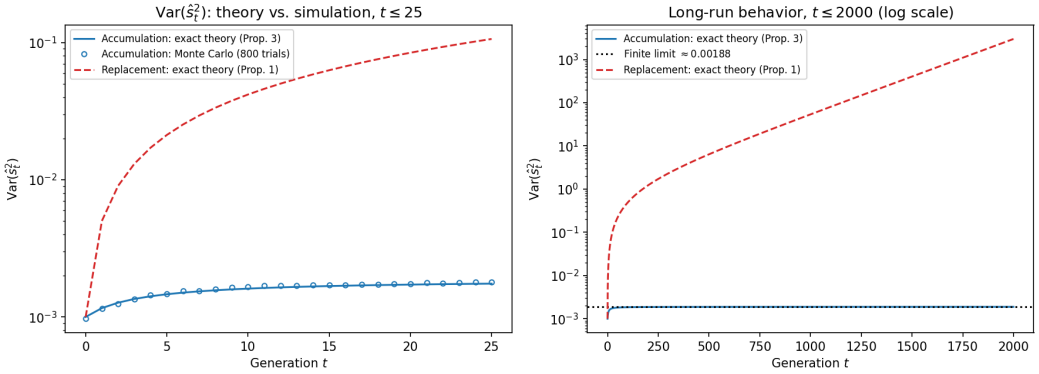

Figure A11: Left: exact theory (Proposition 3, line) vs. Monte Carlo (800 trials, points) for $\mathrm{Var}(\hat{s}_t^2)$ under accumulation, $t \le 25$, with replacement's theoretical curve (Proposition 1) for scale. Right: extending to $t = 2000$ (log scale) shows accumulation converges to the exact finite limit (dotted) while replacement's variance-of-variance continues its geometric blow-up.

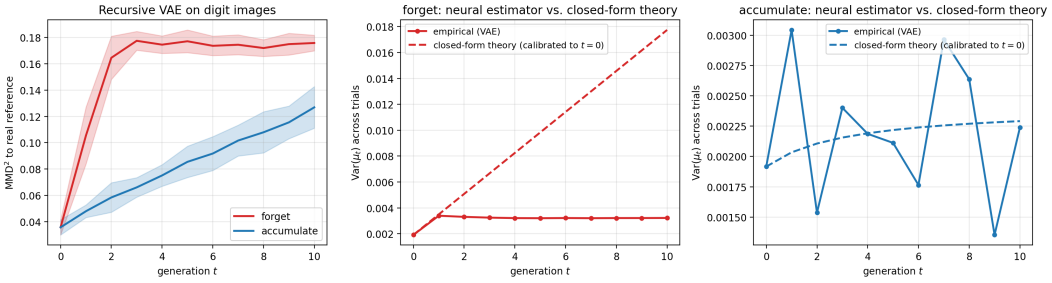

Figure A12: VAE-vs-theory check (Sections 8, 9): left, the usual MMD^2 collapse curve for the VAE (forget vs. accumulate, shading ± 1 std over 20 trials); middle/right, the closed-form Propositions 1–2 (dashed), calibrated to the VAE's own measured generation-0 variance, overlaid on $\mathrm{Var}(\hat{\mu}_t)$ measured across the same 20 trials (solid). Accumulation matches closely; under replacement, the VAE's own variance collapses to a near point-mass within 2–3 generations, so $\mathrm{Var}(\hat{\mu}_t)$ plateaus rather than keeps growing as the idealized theory predicts.

E CODE FOR THE KDE RESAMPLING EXPERIMENTS

```python
import numpy as np
import matplotlib.pyplot as plt
from sklearn.neighbors import KernelDensity
from scipy.stats import entropy, wasserstein_distance

def compute_kl_divergence(p_samples, q_samples, bins=100):
    # Estimate the PDF using histograms on ONE shared set of bin edges
    # spanning both samples' range (independent per-array bin edges
    # bias/inflate-noise the estimate as the KDE support drifts)
    lo = min(p_samples.min(), q_samples.min())
    hi = max(p_samples.max(), q_samples.max())
    edges = np.linspace(lo, hi, bins + 1)
    p_hist, _ = np.histogram(p_samples, bins=edges, density=True)
    q_hist, _ = np.histogram(q_samples, bins=edges, density=True)

    # Ensure no zero values for stability
    p_hist = np.where(p_hist == 0, 1e-10, p_hist)
    q_hist = np.where(q_hist == 0, 1e-10, q_hist)

    # Calculate KL divergence
    return entropy(p_hist, q_hist)

# Generate random samples from an arbitrary distribution
# (a mixture of Gaussians and a uniform)
samples1 = np.random.normal(loc=-2, scale=0.5, size=300)
samples2 = np.random.normal(loc=3, scale=1.0, size=300)
samples3 = np.random.uniform(low=-4, high=-3, size=200)
samples = np.concatenate([samples1, samples2, samples3])

# Shuffle the samples
np.random.shuffle(samples)

# Fit a Kernel Density Estimation model to the samples
kde = KernelDensity(kernel='gaussian', bandwidth=0.5).fit(samples.reshape
    (-1, 1))

# Generate a range of values for plotting the KDE
x = np.linspace(min(samples) - 1, max(samples) + 1, 1000)
log_density = kde.score_samples(x.reshape(-1, 1))
density = np.exp(log_density)

# Generate new samples from the fitted KDE model
fitted_samples = kde.sample(1000).flatten()

# Plot the original samples and the fitted samples
plt.figure(figsize=(10, 6))

plt.hist(samples, bins=30, density=True, alpha=0.5, color='blue', label='
    Original Samples')
plt.hist(fitted_samples, bins=30, density=True, alpha=0.5, color='red',
    label='KDE Fitted Samples')

# Plot the KDE-estimated density
plt.plot(x, density, 'green', lw=2, label='KDE Estimated Density')

plt.title('Original vs KDE Fitted Distribution')
plt.xlabel('Value')
plt.ylabel('Density')
plt.legend(loc='best')

plt.show()
```

Listing 1: Generating and plotting the first mixture of distributions (Figure A3)

```
# Parameters
num_iterations = 30  # Number of iterations
sample_size = 30000  # Number of samples to draw in each iteration

kl_divergences = []   # To store KL divergence values
wsd_divergences = []  # To store Wasserstein distance values

# Fit the initial KDE to the original data
kde = KernelDensity(kernel='gaussian', bandwidth=0.5).fit(samples.reshape(-1, 1))

for i in range(num_iterations):
    # Sample from the current KDE model
    sampled_data = kde.sample(sample_size).flatten()

    kl_div = compute_kl_divergence(sampled_data, samples)
    kl_divergences.append(kl_div)

    wsd = wasserstein_distance(sampled_data, samples)
    wsd_divergences.append(wsd)

    # Print the current iteration and divergences
    print(f"Iteration {i + 1}: KLD = {kl_div:.4f}, WSD = {wsd:.4f}")

    # Refit the KDE model to the new sampled data
    kde = KernelDensity(kernel='gaussian', bandwidth=0.5).fit(sampled_data.reshape(-1, 1))

    if i % 3 == 0:
      # Plot the original samples and the fitted samples
      plt.figure(figsize=(5, 3))

      plt.hist(samples, bins=30, density=True, alpha=0.5, color='blue', label='Original Samples')
      plt.hist(sampled_data, bins=30, density=True, alpha=0.5, color='red', label='KDE Fitted Samples')

      plt.title(f'Itr {i}: Original samples vs. KDE samples')
      plt.xlabel('Value')
      plt.ylabel('Density')
      plt.legend(loc='best')

      plt.show()
```

Listing 2: Repeated sample generation and KDE refitting (Figures A4 and 1)

```
# Mixture of three Gaussians and a uniform
samples0 = np.random.normal(loc=0, scale=0.1, size=300)
samples1 = np.random.normal(loc=-2, scale=0.5, size=300)
samples2 = np.random.normal(loc=3, scale=1.0, size=300)
samples3 = np.random.uniform(low=-4, high=-3, size=200)
samples = np.concatenate([samples0, samples1, samples2, samples3])
```

Listing 3: Mixture of distributions corresponding to the left column of Figure A5

```
# Mixture of a Gamma, two Gaussians, and a uniform
samples0 = np.random.gamma(shape=2, scale=4, size=300)
samples1 = np.random.normal(loc=-2, scale=0.5, size=300)
samples2 = np.random.normal(loc=3, scale=1.0, size=300)
samples3 = np.random.uniform(low=-4, high=-3, size=200)
samples = np.concatenate([samples0, samples1, samples2, samples3])
```

Listing 4: Mixture of distributions corresponding to the right column of Figure A5