COLLAPSE OR THRIVE? PERILS AND PROMISES OF SYNTHETIC DATA IN A SELF-GENERATING WORLD

Anonymous authors
Paper under double-blind review

ABSTRACT

The increasing prevalence of AI-generated content on the internet raises a critical and timely question: What happens when generative machine learning models are pretrained on web-scale datasets containing data created by earlier models? Previous work prophesied *model collapse*, a phenomenon whereby model-generated synthetic data degrades performance with each additional model-fitting and sampling iteration, with newer models becoming more and more useless. In this work, we clarify and unify the fractured literature on the perils and promises of synthetic data in model-data feedback loops to better forecast likely futures of large-scale deep generative models. First, previous work claimed that model collapse is caused by replacing all past data with fresh synthetic data at each model-fitting iteration and that collapse is avoided by instead accumulating data across model-fitting iterations; we test this claim on three prominent generative modeling settings, and find both claims hold in all three settings. Second, we study a middle-ground in which the available data pool contains increasing amounts of synthetic data, but each model is fit using a fixed compute budget; we demonstrate that model test loss on real data increases more quickly, but still plateaus unlike when data are replaced en masse. Third, we investigate whether the cardinality or proportion of real data matters more for avoiding model collapse. Surprisingly, we find a non-trivial interaction between real and synthetic data, where the value of synthetic data for reducing model test loss depends on the absolute quantity of real data. Our insights are particularly important when forecasting whether future deep generative models will collapse or thrive, and our results open avenues for empirically and mathematically studying the context-dependent value of synthetic data.

1 INTRODUCTION: MODEL COLLAPSE & WHY IT MATTERS

With each passing day, the internet contains increasingly more AI-generated content (Altman, 2024). What is the impact of this for future of deep generative models pretrained on web-scale datasets containing data generated by their predecessors? Previous work forewarned that such model-data feedback loops can exhibit *model collapse*, a phenomenon whereby model performance degrades with each model-fitting iteration such that newer models trend towards useless (Shumailov et al., 2023). This prophecy is deeply concerning because society is increasingly relying on these deep generative models (Bommasani et al., 2022; Reuel et al., 2024; Perrault & Clark, 2024; Kapoor et al., 2024), and model collapse threatens that future models will be made useless as society's current data practices pollute the pretraining data supply.

However, the model collapse literature is replete with different experimental methodologies and different mathematical assumptions of different generative models, with different papers reaching different conclusions (Taori & Hashimoto, 2023; Hataya et al., 2023; Martínez et al., 2023a; Shumailov et al., 2023; Alemohammad et al., 2023; Martínez et al., 2023b; Bohacek & Farid, 2023; Guo et al., 2023; Bertrand et al., 2023; Briesch et al., 2023; Dohmatob et al., 2024a;b; Gerstgrasser et al., 2024; Seddik et al., 2024; Marchi et al., 2024; Padmakumar & He, 2024; Chen et al., 2024; Ferbach et al., 2024a; Veprikov et al., 2024). These mixed methods and findings make assessing the probability and harm of model collapse difficult.

In this work, we extend the *accumulate* workflow studied in Gerstgrasser et al. (2024) to cover several settings previously not studied in the literature to exhibit its effectiveness over the *replace* workflow.

We begin by testing the following hypothesis – that model collapse emerges in a scenario where models are trained on evolving datasets built by deleting past data en masse; and model collapse is avoided in a scenario where the training datasets instead accumulate all real and synthetic data. We consider whether these claims hold in three new generative model task settings pointed to by recent prominent work; we find the claims hold. We then compare three clear dataset evolution scenarios, focusing on a new middle ground where data accumulate over time but each model is trained under a fixed compute budget; in this middle ground, we find that losses on real test data climb faster than without a compute budget but plateau to lower values than if data are deleted en masse after each model-fitting iteration. These results are consistent across five different generative modeling settings. Lastly, we investigate, in a specific situation, whether the proportion or cardinality of initial real data matters more for preventing model collapse and discover a non-trivial interaction between real and synthetic data: when real data are scarce, an appropriate amount of synthetic data reduces the test loss on real data, whereas when real data are ample, synthetic data increases the test loss on real data. Altogether, our work provides valuable comprehensive insights for predicting likely futures of deep generative models pretrained on web-scale data.

2 Related Work

Since at least 2022, it has been known that the outputs of image generation models are largely unsuitable for training other image models Hataya et al. (2023). Shumailov et al. (2023) initially sounded alarms about more general autophageous training loops by showing that a model trained repeatedly on its own outputs exhibits severely denigrated quality. Subsequently, Shumailov's experiments, which involved VAE's, GMM's, and fine-tuning of LLM's, were extended across a host of different settings Alemohammad et al. (2023); Bertrand et al. (2023); Dohmatob et al. (2024b;a). While most of the theory concerning model collapse is limited to specific statistical models, recently Marchi et al. (2024) has provided theoretical conditions under which deep generative models collapse.

Until the beginning of 2024, those expressing concern over model collapse primarily operated in the replace setting, discarding all but the generations of the latest model when training the next model. More recently, evidence emerged that when one does not discard data from previous model iterations, collapse can be avoided Gerstgrasser et al. (2024); Seddik et al. (2024). These works have verified this claim in settings involving linear regression, VAE's, and pretraining of LLM's. Similarly, collapse does not occur in image models when one mixes synthetic and real data in equal amounts Martínez et al. (2023a).

Several gaps remain in the existing literature. Though Gerstgrasser et al. (2024) suggests that model collapse is restricted to the replace setting, the universality of this claim remains experimentally unverified. We help to close this gap by testing the hypothesis that replacing data drives model collapse, while accumulating unfiltered data avoids collapse in several remaining settings.

There is a current lack of understanding regarding the boundaries between harmful and helpful synthetic data. Jain et. al. Jain et al. (2024) claim that under certain circumstances, limited amounts of synthetic data can improve test-loss performance, and Ferbach et al. (2024b) suggest that with curation, self-consuming loops can improve alignment with human preferences. A growing literature on how to filter and harness synthetic data has achieved impressive results on a variety of benchmarks Zelikman et al. (2024); Li et al. (2024a); Yang et al. (2024), raising interesting questions about the limits of when unfiltered synthetic data can help. In this vein, Gerstgrasser et al. (2024) ponder, but do not provide an answer to, whether the proportion or the raw amount of real data in a mixed training set has a greater impact on test loss. We confirm that small amounts of unfiltered synthetic data can indeed improve test loss for SFT, and we provide statistics suggesting that both the proportion and the number of unfiltered synthetic datapoints have an impact on test loss.

3 Testing Two Model Collapse Claims in Three New Generative Modeling Settings

Gerstgrasser et al. (2024) recently made two claims about model collapse:

1. Many previous papers induced model collapse by deleting past data en masse and training largely (or solely) on synthetic data from the latest generative model, and

2. If new synthetic data are instead added to real data, i.e., data accumulate over time, then model collapse is avoided.

These two claims are important for forecasting the future of generative models because, if correct, model collapse is then less likely to pose a realistic threat since accumulating data over time is a more realistic modeling assumption; as a partner at Andreessen Horowitz elegantly explained, deleting data en masse is "not what is happening on the internet. We won't replace the Mona Lisa or Lord of the Rings with AI generated data, but the classics will continue to be part of the training data set" (Appenzeller, 2024).

However, these claims have not been tested in three new generative modeling settings recently introduced by prominent work (Shumailov et al., 2024) for studying model collapse:

1. **Multivariate Gaussian Modeling:** Multivariate Gaussians are repeatedly fit to data and then used to sample new synthetic data for future Gaussian fitting.
2. **Kernel Density Estimation:** Kernel density estimators are repeatedly fit to data and then used to sample new synthetic data for future kernel density estimators.
3. **Supervised Finetuning of Language Models:** Language models are finetuned in a supervised manner and then used to sample new synthetic text for future finetuning.

In this section, we ask and answer:

In these three new generative modeling settings, is model collapse caused by deleting data en masse and avoided by instead accumulating data?

In all three settings, we empirically find and (when possible) mathematically prove the answer is yes.

3.1 Model Collapse in Multivariate Gaussian Modeling

We consider repeatedly fitting multivariate Gaussians to data and sampling from the fit Gaussians. We begin with n *real* data drawn from a multivariate Gaussian with mean $\mu^{(0)}$ and covariance $\Sigma^{(0)}$:

$$X_1^{(0)}, ..., X_n^{(0)} \sim_{i.i.d.} \mathcal{N}(\mu^{(0)}, \Sigma^{(0)}).$$

For model fitting, we compute the unbiased mean and covariance of the most recent data:

$$\hat{\mu}_{\text{Replace}}^{(t+1)} \stackrel{\text{def}}{=} \frac{1}{n}\sum_{j=1}^{n} X_j^{(t)} \tag{1}$$

$$\hat{\Sigma}_{\text{Replace}}^{(t+1)} \stackrel{\text{def}}{=} \frac{1}{n-1}\sum_{j=1}^{n}(X_j^{(t)} - \hat{\mu}_{\text{Replace}}^{(t+1)})(X_j^{(t)} - \hat{\mu}_{\text{Replace}}^{(t+1)})^T \tag{2}$$

For model sampling, we sample m new synthetic data using the fit Gaussian parameters:

$$X_1^{(t)}, ..., X_n^{(t)} \,\Big|\, \hat{\mu}_{\text{Replace}}^{(t)}, \hat{\Sigma}_{\text{Replace}}^{(t)} \quad \sim_{i.i.d.} \quad \mathcal{N}(\hat{\mu}_{\text{Replace}}^{(t)}, \hat{\Sigma}_{\text{Replace}}^{(t)}). \tag{3}$$

Under the above data-model feedback loop, Shumailov et al. (2024) prove that

$$\hat{\Sigma}_{\text{Replace}}^{(t+1)} \stackrel{a.s.}{\to} 0 \quad ; \quad \mathbb{E}[\mathbb{W}_2^2(\mathcal{N}(\hat{\mu}_{\text{Replace}}^{(t+1)}, \hat{\Sigma}_{\text{Replace}}^{(t+1)}), \mathcal{N}(\mu^{(0)}, \Sigma^{(0)}))] \to \infty \text{ as } t \to \infty, \tag{4}$$

where $\mathbb{W}_2$ denotes the Wasserstein-2 distance. This result states that the fit covariance will collapse to 0 and that the Wasserstein-2 distance will diverge as this model-data feedback loop unfolds. Note that the Wasserstein-2 distance diverges not because the covariance collapses to 0 but because the distance between the t-th fit mean $\hat{\mu}_{\text{Replace}}^{(t)}$ and the true mean $\mu^{(0)}$ diverges.

However, *this result assumes that all data are deleted after each model-fitting iteration*. As discussed above, this assumption is likely unrealistic because society does not delete earlier content from the internet and replace it with new model-generated content after fitting each state-of-the-art model.

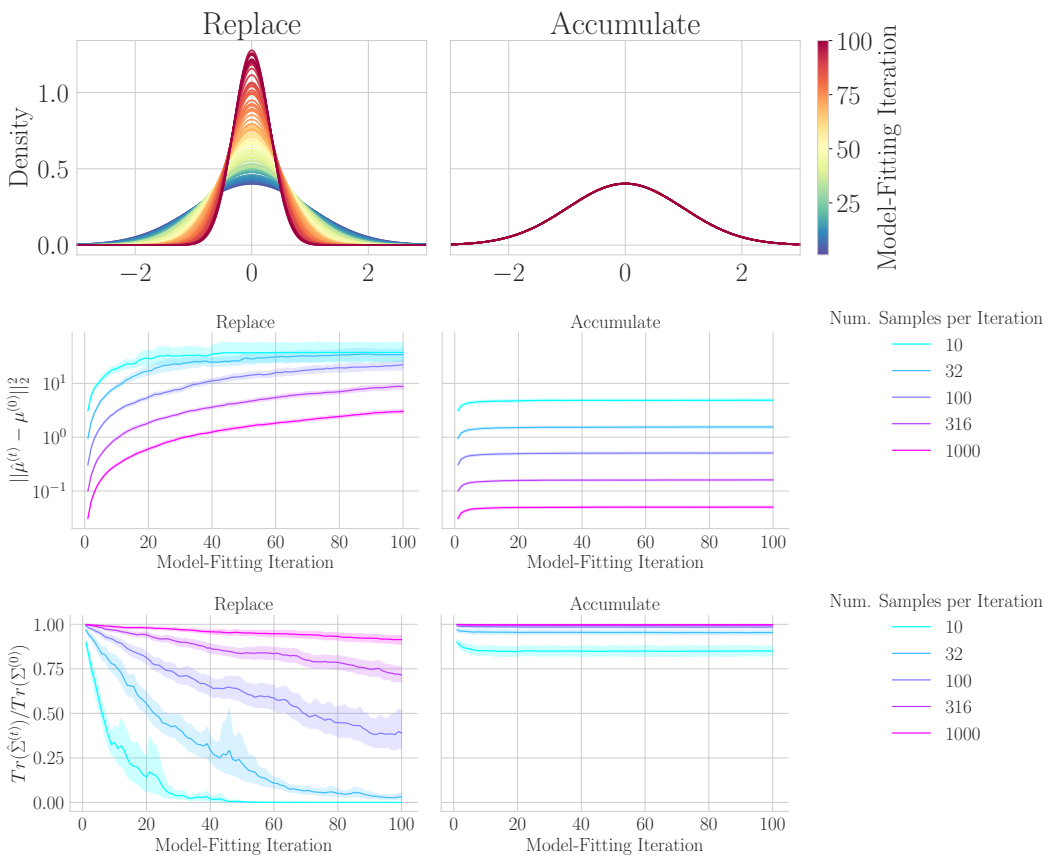

Figure 1: **Model Collapse in Multivariate Gaussian Modeling. Top:** Previous work (Shumailov et al., 2024) proves model collapse occurs if one iteratively fits means and covariances to data and then samples new data from a Gaussian with the fitted parameters (left). We demonstrate that if one doesn't delete all data after each model-fitting iteration - i.e., if data accumulate - then model collapse does not occur (right). Number of Samples Per Iteration: 316. Note: We visualize the fit Gaussians as zero-mean for easy comparison of the fit covariances across model-fitting iterations. **Middle:** If data are replaced, then the empirically fit means drift away from the original data's mean with increasing model-fitting iterations, but if data instead accumulate, then the empirically fit means stabilize. **Bottom:** If data are replaced, then the empirically fit covariances collapse compared to the original data's covariance, but if past data are not discarded, then the fit covariances stabilize quickly and collapse is avoided. Note: Rows 2 and 3 correspond to $d = 31$ dimensional data.

What happens if data instead *accumulate* across model-fitting iterations? To study this, we instead consider fitting to all previous real and synthetic data:

$$\hat{\mu}^{(t+1)}_{\text{Accumulate}} \stackrel{\text{def}}{=} \frac{1}{n(t+1)} \sum_{i=0}^{t} \sum_{j=1}^{n} X_j^{(i)} \tag{5}$$

$$\hat{\Sigma}^{(t+1)}_{\text{Accumulate}} \stackrel{\text{def}}{=} \frac{1}{n(t+1)-1} \sum_{i=0}^{t} \sum_{j=1}^{n} (X_j^{(i)} - \hat{\mu}^{(t+1)}_{\text{Accumulate}})(X_j^{(i)} - \hat{\mu}^{(t+1)}_{\text{Accumulate}})^T \tag{6}$$

Data are then sampled using these fit Accumulate parameters rather than the fit Replace parameters.

Empirically, we find that deleting all data after each model-fitting iteration causes model collapse (Fig. 1 Left), whereas accumulating data across model-fitting iterations prevents model collapse (Fig. 1 Right). More specifically, we find that if data are deleted, the squared error between the fit mean $\hat{\mu}^{(n)}_{\text{Replace}}$ and the initial mean $\mu^{(0)}$ diverges (Fig. 1, Middle Left), and the fit covariance $\hat{\Sigma}^{(n)}_{\text{Replace}}$ relative to the initial covariance $\Sigma^{(0)}$ collapses to 0 (Fig. 1, Bottom Left), as measured by the ratio

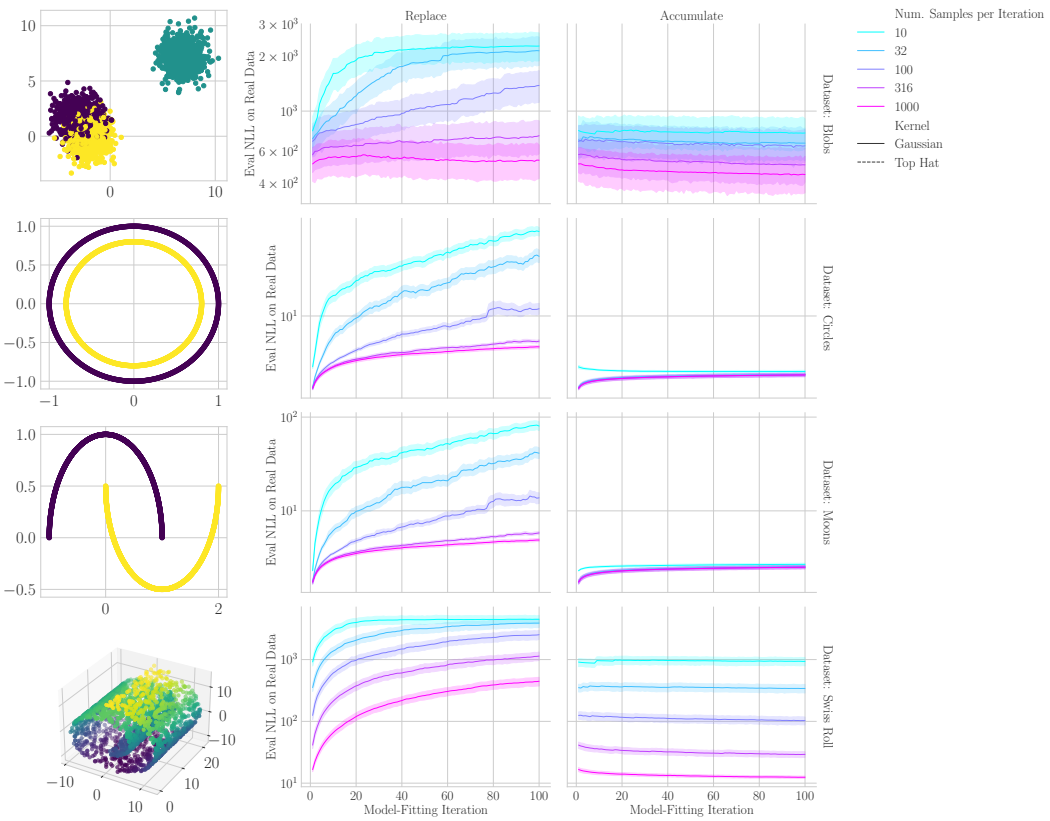

Figure 2: **Model Collapse in Kernel Density Estimation.** Left: We consider 4 standard datasets from `sklearn`: Blobs, Circles, Moons and Swiss Roll. Center: For all four datasets, deleting data en masse causes the negative log likelihoods (NLL) of real test data to increase with each model-fitting iteration. Right: For all four datasets, accumulating data avoids model collapse. Interestingly, for specific pairs of datasets and number of samples per iteration, training on real and accumulating synthetic data can yield *lower loss on real test data* than training on real data alone.

between the trace of $\hat{\Sigma}^{(t)}$ and the trace of $\Sigma^{(0)}$. In contrast, if data accumulate, the squared error between the fit mean and the initial mean plateaus quickly (Fig. 1, Middle Right), as does the fit covariance relative to the initial covariance (Fig. 1, Bottom Right).

Additionally, in the univariate case, we mathematically characterize the limit distribution:

Theorem 1. *For notational efficiency, for a univariate Gaussian, let $\hat{\mu}^{(t)}$ and $\hat{\sigma}^{(t)}$ denote $\hat{\mu}^{(t)}_{Accumulate}$ and $\hat{\Sigma}^{(t)}_{Accumulate}$. Suppose that the mean and covariance are updated as in Eqns. 5 and 6. Then*

$$\mathbb{E}\left(\sigma_t^2\right) = \sigma_0^2 \cdot \prod_{k=1}^{t}\left(1 - \frac{1}{k^2 n}\right) \xrightarrow{t\to\infty} \sigma_0^2 \cdot \left(\frac{\sin(\pi/\sqrt{n})}{\pi/\sqrt{n}}\right) \tag{7}$$

$$\mathbb{E}[(\mu_t - \mu_0)^2] = \sigma_0^2 \cdot \left(1 - \prod_{k=1}^{t}\left(1 - \frac{1}{k^2 n}\right)\right) \xrightarrow{t\to\infty} \sigma_0^2 \cdot \left(1 - \frac{\sin(\pi/\sqrt{n})}{\pi/\sqrt{n}}\right). \tag{8}$$

See Appendix Sec. A for the proof. This reveals two key differences when data accumulate: the covariance no longer collapses, and the mean no longer diverges, meaning model collapse is mitigated.

3.2 Model Collapse in Kernel Density Estimation

We next turn to the second generative modeling setting for studying model collapsed introduced by Shumailov et al. (2024): kernel density estimation (KDE). Similar to multivariate Gaussian

modeling, we begin with n real data points drawn from an initial probability distribution $p^{(0)}$: $X_1^{(0)}, ..., X_n^{(0)} \sim_{i.i.d.} p^{(0)}$. We then iteratively fit KDEs to the data and sample new synthetic data from these estimators. In the Replace setting, we fit the KDE to n data samples from the most recently fit model, whereas in the Accumulate setting, we fit the KDE to all data points from all previous iterations, with the number of points growing linearly as $n(t+1)$:

$$\hat{p}^{(t+1)}_{\text{Replace}}(x) \stackrel{\text{def}}{=} \frac{1}{nh} \sum_{j=1}^{n} K\Big(\frac{x - X_j^{(t)}}{h}\Big) \tag{9}$$

$$\hat{p}^{(t+1)}_{\text{Accumulate}}(x) \stackrel{\text{def}}{=} \frac{1}{nh(t+1)} \sum_{i=0}^{t} \sum_{j=1}^{n} K\Big(\frac{x - X_j^{(i)}}{h}\Big) \tag{10}$$

where K is the kernel function and h is the bandwidth parameter. We consider a standard Gaussian kernel. For sampling, at each iteration, we draw n new synthetic data points from the fitted kernel density estimators. We evaluate the performance using the negative log-likelihood (NLL) on real held-out test data; lower NLL indicates better performance. For data, we use four standard synthetic datasets from `sklearn` (Pedregosa et al., 2011): blobs, circles, moons, and swiss roll.

We again observe the same general difference between replacing data and accumulating data (Fig. 2): replacing data causes a rapid increase in NLL as the number of model-fitting iterations increases, indicating that the KDEs are becoming increasingly poor at modeling the true underlying distribution. In contrast, when data accumulate across model-fitting iterations, we observe that the NLL remains relatively stable, suggesting that accumulating data helps maintain the quality of the KDEs.

Despite the apparent empirical similarities to the iterative Gaussian fitting shown in Figure 1, Gaussian KDEs are theoretically distinct. Unlike Gaussian fitting, regardless of whether we accumulate or replace, the test NLL for a Gaussian KDE asymptotically diverges in theory. There are two interesting caveats: (1) when one begins with a small bandwidth, iteratively fitting KDEs can cause the NLL to initially decrease before it diverges due to the effective kernel bandwidth increasing with model-fitting iterations; (2) although accumulating data causes the NLL to diverge asymptotically, this occurs at a rate so glacial that it doesn't pose a practical concern. If one wishes to prevent the eventual divergence, one can do so by fitting at each iteration with the optimal bandwidth for the number of data, which should be of the form $c(in)^{-1/5}$ in the ith model-fitting iteration as long as data accumulates at a constant rate. Practically speaking, one chooses the bandwidth for KDEs based on the number and characteristics of the data, implying that conscientious practitioners should never witness severe model collapse for KDEs in the accumulate case. For details, see Appendix Sec. B.

We also note a surprising discovery: accumulating data can yield NLLs that decrease with additional model-fitting iterations, meaning that training on real and synthetic data yields lower test loss on real data than training on real data alone. While synthetic data has been shown valuable elsewhere, e.g., Jain et al. (2024), we were surprised to discover this behavior in such a simple setting. This behavior is analogous to Mobahi et al. (2020), which demonstrated how self-distillation of linear models can initially improve model performance by acting as increasing regularization in Hilbert space, but if too many iterations take place, the predictor is regularized towards 0 and performance deteriorates.

3.3 Model Collapse in Supervised Finetuning of Language Models

We now turn to the third setting for studying model collapse introduced by Shumailov et al. (2024): supervised finetuning of language models. We begin with an instruction following dataset – Nvidia's HelpSteer2 (Wang et al., 2024) – and finetune a language model before sampling new text data from it. We choose Google's Gemma2 2B model (Team et al., 2024) because it is high performing and relatively small. For Replace, we fine-tune the n-th language model only on data generated by the $(n-1)$ language model. For Accumulate, we instead fine-tune the n-th language model on the starting real data plus all the synthetic data sampled from all previous models; thus, the amount of data for Replace is constant $\sim 12.5k$, whereas the amount of data for Accumulate grows linearly $\sim 12.5k * t$. Consistent with our results and with Gerstgrasser et al. (2024), we find that deleting data after each iteration leads to collapse whereas accumulating data avoids collapse (Fig. 3).

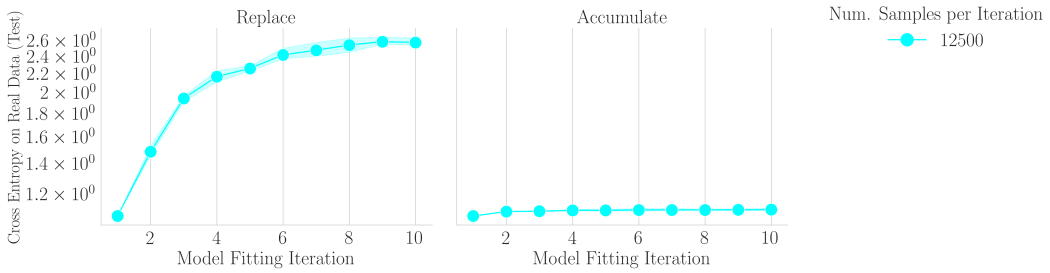

Figure 3: **Model Collapse in Supervised Finetuning of Language Models.** Finetuning Google's Gemma2 2B on Nvidia's HelpSteer 2 dataset demonstrates that model collapse occurs if previous data are replaced after each model-fitting iteration (left), whereas model collapse is avoided if new synthetic data instead accumulate with previous real and synthetic data (right).

4 Model Collapse Under a Fixed Compute Budget

Thus far, we have focused on two data paradigms: Replace and Accumulate. As discussed in Sec. 3, Replace is unlikely to be an faithful model of reality because we do not delete the internet after pretraining each model, but one might also argue that Accumulate is similarly unfaithful because Accumulate requires that every new model is trained on (linearly) more data and thus requires more compute than its predecessor. Whether this criticism is valid in practice is unclear, since newer models *are* trained on increasing data (e.g., 1.4T tokens for Llama 1, 2T tokens for Llama 2, 15T tokens from Llama 3) and increasing GPUs (e.g., 2k GPUs for Llama1, 4K for Llama2, 16k for Llama3 (Goyal, 2024)). Nevertheless, for the sake of understanding the space of possible outcomes and predicting likely outcomes for future generative models, we ask and answer:

Does model collapse occur when data accumulate but models are trained under a fixed compute budget?

We call this data paradigm *Accumulate-Subsample* because data accumulate but are then subsampled to ensure constant data and thus constant compute at each model-fitting iteration. To study whether model collapse occurs in Accumulate-Subsample, we use the same three generative modeling settings we've studied (multivariate Gaussian modeling, supervised finetuning of language models and kernel density estimation) plus two new generative modeling settings studied by prior work (Mobahi et al., 2020; Dohmatob et al., 2024a; Gerstgrasser et al., 2024): linear regression and pretraining language models on a GPT3.5/GPT4-generated dataset of kindergarten-level text (Eldan & Li, 2023).

To explain how linear regression can be used as a generative model, we briefly here and direct the reader to prior work (Mobahi et al., 2020; Dohmatob et al., 2024a; Gerstgrasser et al., 2024) for a more thorough description. We begin with our real covariates $X \in \mathbb{R}^{n \times d}$ and true linear relationship $w^{(0)}$. Initializing $\hat{w}^{(0)} = w^{(0)}$, we sample the regression targets as:

$$y^{(t)} \stackrel{\text{def}}{=} X\hat{w}^{(t)} + E^{(t)} \quad ; \quad E^{(t)} \sim \mathcal{N}(0, \sigma^2 I_d) \tag{11}$$

Assuming $X^T X$ is full rank, e.g., $n \gg d$, we fit the next linear model using ordinary least squares:

$$\hat{w}^{(t+1)} \stackrel{\text{def}}{=} (X^T X)^{-1} X^T y^{(t)} \tag{12}$$

Following Gerstgrasser et al. (2024), we additionally pretrain sequences of small variants of common large language models – GPT (Radford et al., 2019; Brown et al., 2020) and Llama (Touvron et al., 2023a;b) – on TinyStories (Eldan & Li, 2023), a synthetic dataset of simple short stories; this combination of models, parameters and data was chosen to faithfully study model collapse in as realistic a setting as possible, subject to our limited computational budget.

Across all five generative modeling settings, we find that Accumulate-Subsample's test loss on real data lies between the test losses of Replace and Accumulate (Fig. 4 Center). Specifically, Accumulate-Subsample (Fig. 4 center) exhibits higher test loss than Accumulate (Fig. 4, Right) but lower test loss than Replace (Fig. 4 Left), showing that the fixed compute budget imposes some cost. In a

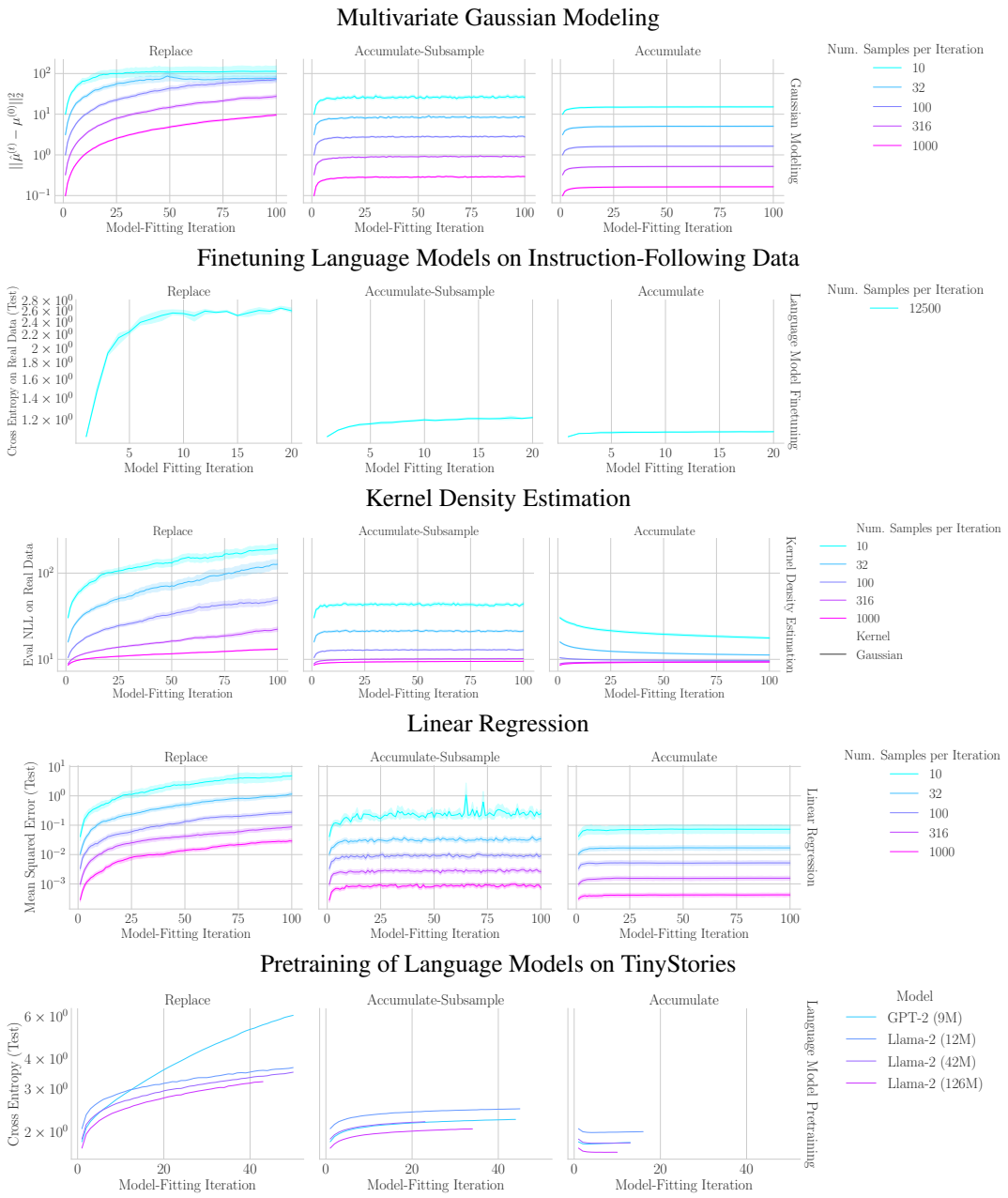

Figure 4: **Model Collapse Under a Fixed Compute Budget.** We compare deleting data after each model-fitting iteration (Replace) and accumulating data after each iteration (Accumulate) with a new fixed-compute data paradigm Accumulate-Subsample. In Accumulate-Subsample, real and synthetic data accumulate but are then subsampled so that each model is trained on a constant number of data. Accumulate-Subsample's test loss on real data deteriorates more quickly than Accumulate's loss but more slowly than Replace's loss, and frequently converges, albeit to a higher plateau than Accumulate. These results hold across five settings: multivariate Gaussian modeling, language model instruction finetuning, kernel density estimation, linear regression and language model pretraining.

qualitative difference, test losses on real data typically plateaus for both Accumulate-Subsample and Accumulate, whereas test losses for Replace typically diverge in an apparently unbounded manner. These results collectively tell a consistent story: under more realistic conditions, where data accumulate and compute is bounded, model performance on real test data is unlikely to diverge.

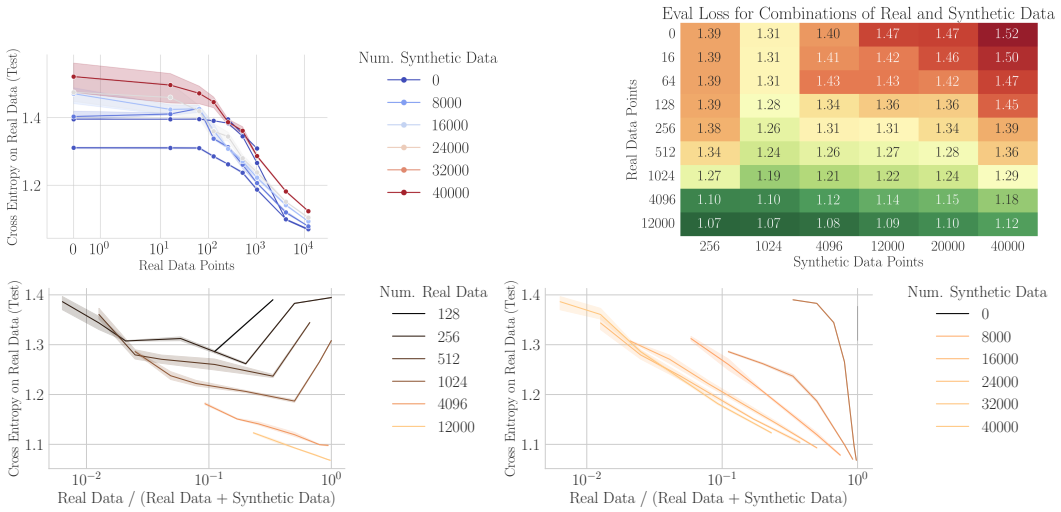

Figure 5: **The Value of Synthetic Data in Supervised Finetuning of Language Models.** Finetuning Google's Gemma 2 2B on Nvidia's HelpSteer 2 dataset on different combinations of real and synthetic data demonstrates that loss grows with the number of synthetic data. Our results suggest that the test loss depends on both the proportion (p-value$= 3.84 \times 10^{-16}$) and the cardinality (p-value$= 3.54 \times 10^{-8}$) of real data. We plot the test loss against the number of real datapoints in the training set (top left). The hue represents the number of synthetic datapoints. Additionally, we display a heatmap demonstrating the effect of the number of real and synthetic datapoints on test loss (top right). We provide a graph of the test loss versus the fraction of real data, where the hue represents the cardinality of the real data (bottom left). Finally, we plot the test loss against the fraction of real datapoints, where the hue represents the number of synthetic datapoints (bottom right).

5 Cardinality of Real Data vs Proportion of Real Data in Mitigating Model Collapse

We conclude by turning to a question asked by Gerstgrasser et al. (2024) that, to the best of our knowledge, remains open:

Which matters more for avoiding model collapse: the cardinality of real data or the proportion of real data? Relatedly, how does the value of synthetic data for reducing test loss on real data depend on the amount of real data?

These questions are highly pertinent to researchers sampling from web-scale data in order to pretrain or finetune language models. We conduct our investigation of this question as follows: First, we perform SFT on the HelpSteer2 dataset for Google's Gemma 2 2B model. We sample 100k completions from the finetuned model and filter for those that are fewer than 512 tokens in length. This leaves us with over 55,000 remaining completions. We aggregate datasets containing various numbers of real and synthetic synthetic data, which are given in Figure 5, and perform SFT on these datasets starting from the original Gemma 2B model. We record and display the final test loss from this process.

This experiment provides several insights. First, both the number and proportion of real data have an impact on the test loss following SFT. To assess this, we first transformed the number of real datapoints n as $\frac{1}{n^{1/2}}$, in keeping with intuitions from classical statistics on how the log likelihood scales with the number of data points. Then, based on observation of the data, we computed

$$\log\left(\frac{\text{real data}}{\text{real data} + \text{synthetic data}}\right)$$

to best capture the relationship between the fraction of real data and the log likelihood. We measured R^2 values of 0.59 for the transformed number of real data and 0.34 for the proportion of real data. We then computed F-statistics for the one-term versus two term models involving each of these

covariates, which gave us p-values of 6.9×10^{-25} and 4.6×10^{-25}. These statistics suggest that both the proportion and the cardinality of real data have a statistically significant effect on the test loss, and explain a sizable fraction of the variance in the test loss.

Second, we find a difference in the effect that synthetic data has on test loss in high versus low real data regimes. In our experiments specifically, when the number of real data is 1024 or lower, we find that there is an *small but non-zero amount of synthetic data that improves the test loss when it is included*. This suggests that practitioners fine-tuning with insufficient amounts of real data should consider supplementing with synthetic data to improve model quality.

On the other hand, when real data are plentiful, we find that more synthetic data almost always harms final model quality when the number of real data is held constant. In some cases, datasets containing only real data prove to be more valuable than datasets that contain ten times more real data mixed with synthetic data.

Although these results are preliminary, they raise some interesting questions about the role of synthetic data in SFT that deserve further exploration. In some of our experiments, we achieve better results by removing all synthetic data from the training set than by doubling the amount of real training data. More generally, when constructing datasets subject to cost constraints, these results suggest that the value of removing synthetic or low-quality data can sometimes exceed that of collecting greater volumes high-quality data.

6 Discussion

Our work sought to clarify and unify understanding of model-data feedback loops. We demonstrated in three new generative modeling settings that accumulating data over time avoids model collapse, whereas replacing data over time induces model collapse. We then demonstrated in five generative modeling settings that even when each model is trained on a fixed compute budget with a mixture of real and synthetic data, model performance does deteriorate more, but still tends to plateau. The consistency of these results across different model types and datasets suggests that *this distinction is a general phenomenon, and is not specific to any particular model, dataset, or learning algorithm.* Lastly, we explored the value of synthetic data for reducing the test loss on real data and found two different regimes: when real data are plentiful, synthetic data is harmful, but when real data are scarce, there exists an optimal amount of synthetic data that are helpful.

In our view, the data paradigm in which synthetic data accumulates from a host of models in conjunction with a constant influx of real-world data is more realistic. Under such dynamics, where new synthetic data are added to existing real and synthetic data, model collapse appears unlikely. Our experiments take a pessimistic viewpoint, in the sense that our experiments pay no attention to the quality of data, whereas in practice, engineers heavily filter data based on various indicators of data quality, e.g., (Brown et al., 2020; Lee et al., 2023; Wettig et al., 2024; Penedo et al., 2024; Li et al., 2024b; Sachdeva et al., 2024); for a recent review, see Albalak et al. (2024).

7 Future Directions

An especially interesting future direction is how to combine synthetic data generation with filtering techniques to enable performant and efficient pretraining at scale using synthetic data. As we saw in kernel density estimation (Fig. 2) and in language model pretraining on TinyStories (Fig. 4), training on accumulating real and synthetic data can yield lower loss on real test data than training on real data alone. Identifying under what conditions, and why, this is possible is a tantalizing prospect.

Our results in Section 5 suggest that removing low-quality synthetic data from model training sets *can improve test loss more than gathering additional high-quality data*. Developing efficient removal techniques for detrimental data would streamline the model fine-tuning process and produce better alignment. Future work that further studies and quantifies the harm done by lower-quality synthetic data across broader settings could greatly improve the efficiency of model training.

REFERENCES

Alon Albalak, Yanai Elazar, Sang Michael Xie, Shayne Longpre, Nathan Lambert, Xinyi Wang, Niklas Muennighoff, Bairu Hou, Liangming Pan, Haewon Jeong, Colin Raffel, Shiyu Chang, Tatsunori Hashimoto, and William Yang Wang. A survey on data selection for language models, 2024. URL https://arxiv.org/abs/2402.16827.

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

Sam Altman. openai now generates about 100 billion words per day., Feb 2024. URL https://twitter.com/sama/status/1756089361609981993. [Online; accessed 13-October-2024].

Guido Appenzeller. The internet contains an increasing amount of ai generated data... LinkedIn, Jul 2024. URL https://www.linkedin.com/posts/appenz_the-internet-contains-an-increasing-amount-activity-7223028230444785664-wg86. [Online; accessed 13-October-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. *arXiv preprint arXiv:2310.00429*, 2023.

Matyas Bohacek and Hany Farid. Nepotistically trained generative-ai models collapse. *arXiv preprint arXiv:2311.12202*, 2023.

Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, Shyamal Buch, Dallas Card, Rodrigo Castellon, Niladri Chatterji, Annie Chen, Kathleen Creel, Jared Quincy Davis, Dora Demszky, Chris Donahue, Moussa Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren Gillespie, Karan Goel, Noah Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling, Fereshte Khani, Omar Khattab, Pang Wei Koh, Mark Krass, Ranjay Krishna, Rohith Kuditipudi, Ananya Kumar, Faisal Ladhak, Mina Lee, Tony Lee, Jure Leskovec, Isabelle Levent, Xiang Lisa Li, Xuechen Li, Tengyu Ma, Ali Malik, Christopher D. Manning, Suvir Mirchandani, Eric Mitchell, Zanele Munyikwa, Suraj Nair, Avanika Narayan, Deepak Narayanan, Ben Newman, Allen Nie, Juan Carlos Niebles, Hamed Nilforoshan, Julian Nyarko, Giray Ogut, Laurel Orr, Isabel Papadimitriou, Joon Sung Park, Chris Piech, Eva Portelance, Christopher Potts, Aditi Raghunathan, Rob Reich, Hongyu Ren, Frieda Rong, Yusuf Roohani, Camilo Ruiz, Jack Ryan, Christopher Ré, Dorsa Sadigh, Shiori Sagawa, Keshav Santhanam, Andy Shih, Krishnan Srinivasan, Alex Tamkin, Rohan Taori, Armin W. Thomas, Florian Tramèr, Rose E. Wang, William Wang, Bohan Wu, Jiajun Wu, Yuhuai Wu, Sang Michael Xie, Michihiro Yasunaga, Jiaxuan You, Matei Zaharia, Michael Zhang, Tianyi Zhang, Xikun Zhang, Yuhui Zhang, Lucia Zheng, Kaitlyn Zhou, and Percy Liang. On the opportunities and risks of foundation models, 2022. URL https://arxiv.org/abs/2108.07258.

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*, 2023.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020. URL https://arxiv.org/abs/2005.14165.

Tianwei Chen, Yusuke Hirota, Mayu Otani, Noa Garcia, and Yuta Nakashima. Would deep generative models amplify bias in future models? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10833–10843, 2024.

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

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

Ronen Eldan and Yuanzhi Li. Tinystories: How small can language models be and still speak coherent english? *arXiv preprint arXiv:2305.07759*, 2023.

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

Damien Ferbach, Quentin Bertrand, Avishek Joey Bose, and Gauthier Gidel. Self-consuming generative models with curated data provably optimize human preferences, 2024b. URL https://arxiv.org/abs/2407.09499.

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, 2024. URL https://arxiv.org/abs/2404.01413.

Naman Goyal. llama1: 2048 gpus llama2: 4096 gpus llama3: 16384 gpus llama4: ....., Jul 2024. URL https://twitter.com/NamanGoyal21/status/1815819622525870223. [Online; accessed 13-October-2024].

Yanzhu Guo, Guokan Shang, Michalis Vazirgiannis, and Chloé Clavel. The curious decline of linguistic diversity: Training language models on synthetic text. *arXiv preprint arXiv:2311.09807*, 2023.

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

Ayush Jain, Andrea Montanari, and Eren Sasoglu. Scaling laws for learning with real and surrogate data, 2024. URL https://arxiv.org/abs/2402.04376.

Sayash Kapoor, Rishi Bommasani, Kevin Klyman, Shayne Longpre, Ashwin Ramaswami, Peter Cihon, Aspen Hopkins, Kevin Bankston, Stella Biderman, Miranda Bogen, et al. Position: On the societal impact of open foundation models. In *International Conference on Machine Learning*, pp. 23082–23104. PMLR, 2024.

Alycia Lee, Brando Miranda, Sudharsan Sundar, and Sanmi Koyejo. Beyond scale: the diversity coefficient as a data quality metric demonstrates llms are pre-trained on formally diverse data. *arXiv preprint arXiv:2306.13840*, 2023.

Chengpeng Li, Zheng Yuan, Hongyi Yuan, Guanting Dong, Keming Lu, Jiancan Wu, Chuanqi Tan, Xiang Wang, and Chang Zhou. Mugglemath: Assessing the impact of query and response augmentation on math reasoning, 2024a. URL https://arxiv.org/abs/2310.05506.

Jeffrey Li, Alex Fang, Georgios Smyrnis, Maor Ivgi, Matt Jordan, Samir Gadre, Hritik Bansal, Etash Guha, Sedrick Keh, Kushal Arora, et al. Datacomp-lm: In search of the next generation of training sets for language models. *arXiv preprint arXiv:2406.11794*, 2024b.

Matteo Marchi, Stefano Soatto, Pratik Chaudhari, and Paulo Tabuada. Heat death of generative models in closed-loop learning, 2024. URL https://arxiv.org/abs/2404.02325.

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

Gonzalo Martínez, Lauren Watson, Pedro Reviriego, José Alberto Hernández, Marc Juarez, and Rik Sarkar. Towards understanding the interplay of generative artificial intelligence and the internet. *arXiv preprint arXiv:2306.06130*, 2023b.

Hossein Mobahi, Mehrdad Farajtabar, and Peter Bartlett. Self-distillation amplifies regularization in hilbert space. *Advances in Neural Information Processing Systems*, 33:3351–3361, 2020.

Vishakh Padmakumar and He He. Does writing with language models reduce content diversity?, 2024. URL https://arxiv.org/abs/2309.05196.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Guilherme Penedo, Hynek Kydlíček, Loubna Ben allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, and Thomas Wolf. The fineweb datasets: Decanting the web for the finest text data at scale, 2024. URL https://arxiv.org/abs/2406.17557.

Ray Perrault and Jack Clark. Artificial intelligence index report 2024. 2024.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Anka Reuel, Ben Bucknall, Stephen Casper, Tim Fist, Lisa Soder, Onni Aarne, Lewis Hammond, Lujain Ibrahim, Alan Chan, Peter Wills, Markus Anderljung, Ben Garfinkel, Lennart Heim, Andrew Trask, Gabriel Mukobi, Rylan Schaeffer, Mauricio Baker, Sara Hooker, Irene Solaiman, Alexandra Sasha Luccioni, Nitarshan Rajkumar, Nicolas Moës, Jeffrey Ladish, Neel Guha, Jessica Newman, Yoshua Bengio, Tobin South, Alex Pentland, Sanmi Koyejo, Mykel J. Kochenderfer, and Robert Trager. Open problems in technical ai governance, 2024. URL https://arxiv.org/abs/2407.14981.

Noveen Sachdeva, Benjamin Coleman, Wang-Cheng Kang, Jianmo Ni, Lichan Hong, Ed H Chi, James Caverlee, Julian McAuley, and Derek Zhiyuan Cheng. How to train data-efficient llms. *arXiv preprint arXiv:2402.09668*, 2024.

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, 2024.

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

Ilia Shumailov, Zakhar Shumaylov, Yiren Zhao, Nicolas Papernot, Ross Anderson, and Yarin Gal. Ai models collapse when trained on recursively generated data. *Nature*, 631(8022):755–759, 2024. ISSN 1476-4687. doi: 10.1038/s41586-024-07566-y. URL https://doi.org/10.1038/s41586-024-07566-y.

Rohan Taori and Tatsunori Hashimoto. Data feedback loops: Model-driven amplification of dataset biases. In *International Conference on Machine Learning*, pp. 33883–33920. PMLR, 2023.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, Johan Ferret, Peter Liu, Pouya Tafti, Abe Friesen, Michelle Casbon, Sabela Ramos, Ravin Kumar, Charline Le Lan, Sammy Jerome, Anton Tsitsulin, Nino Vieillard, Piotr Stanczyk, Sertan Girgin, Nikola Momchev, Matt Hoffman, Shantanu Thakoor, Jean-Bastien Grill, Behnam Neyshabur, Olivier Bachem, Alanna Walton, Aliaksei Severyn, Alicia Parrish, Aliya Ahmad, Allen Hutchison, Alvin Abdagic, Amanda Carl, Amy Shen, Andy Brock, Andy Coenen, Anthony Laforge, Antonia Paterson, Ben Bastian, Bilal Piot, Bo Wu, Brandon Royal, Charlie Chen, Chintu Kumar, Chris Perry, Chris Welty, Christopher A. Choquette-Choo, Danila Sinopalnikov, David Weinberger, Dimple Vijaykumar, Dominika Rogozińska, Dustin Herbison, Elisa Bandy, Emma Wang, Eric Noland, Erica Moreira, Evan Senter, Evgenii Eltyshev, Francesco Visin, Gabriel Rasskin, Gary Wei, Glenn Cameron, Gus Martins, Hadi Hashemi, Hanna Klimczak-Plucińska, Harleen Batra,

Harsh Dhand, Ivan Nardini, Jacinda Mein, Jack Zhou, James Svensson, Jeff Stanway, Jetha Chan, Jin Peng Zhou, Joana Carrasqueira, Joana Iljazi, Jocelyn Becker, Joe Fernandez, Joost van Amersfoort, Josh Gordon, Josh Lipschultz, Josh Newlan, Ju yeong Ji, Kareem Mohamed, Kartikeya Badola, Kat Black, Katie Millican, Keelin McDonell, Kelvin Nguyen, Kiranbir Sodhia, Kish Greene, Lars Lowe Sjoesund, Lauren Usui, Laurent Sifre, Lena Heuermann, Leticia Lago, Lilly McNealus, Livio Baldini Soares, Logan Kilpatrick, Lucas Dixon, Luciano Martins, Machel Reid, Manvinder Singh, Mark Iverson, Martin Görner, Mat Velloso, Mateo Wirth, Matt Davidow, Matt Miller, Matthew Rahtz, Matthew Watson, Meg Risdal, Mehran Kazemi, Michael Moynihan, Ming Zhang, Minsuk Kahng, Minwoo Park, Mofi Rahman, Mohit Khatwani, Natalie Dao, Nenshad Bardoliwalla, Nesh Devanathan, Neta Dumai, Nilay Chauhan, Oscar Wahltinez, Pankil Botarda, Parker Barnes, Paul Barham, Paul Michel, Pengchong Jin, Petko Georgiev, Phil Culliton, Pradeep Kuppala, Ramona Comanescu, Ramona Merhej, Reena Jana, Reza Ardeshir Rokni, Rishabh Agarwal, Ryan Mullins, Samaneh Saadat, Sara Mc Carthy, Sarah Perrin, Sébastien M. R. Arnold, Sebastian Krause, Shengyang Dai, Shruti Garg, Shruti Sheth, Sue Ronstrom, Susan Chan, Timothy Jordan, Ting Yu, Tom Eccles, Tom Hennigan, Tomas Kocisky, Tulsee Doshi, Vihan Jain, Vikas Yadav, Vilobh Meshram, Vishal Dharmadhikari, Warren Barkley, Wei Wei, Wenming Ye, Woohyun Han, Woosuk Kwon, Xiang Xu, Zhe Shen, Zhitao Gong, Zichuan Wei, Victor Cotruta, Phoebe Kirk, Anand Rao, Minh Giang, Ludovic Peran, Tris Warkentin, Eli Collins, Joelle Barral, Zoubin Ghahramani, Raia Hadsell, D. Sculley, Jeanine Banks, Anca Dragan, Slav Petrov, Oriol Vinyals, Jeff Dean, Demis Hassabis, Koray Kavukcuoglu, Clement Farabet, Elena Buchatskaya, Sebastian Borgeaud, Noah Fiedel, Armand Joulin, Kathleen Kenealy, Robert Dadashi, and Alek Andreev. Gemma 2: Improving open language models at a practical size, 2024. URL https://arxiv.org/abs/2408.00118.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023a.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023b.

Andrey Veprikov, Alexander Afanasiev, and Anton Khritankov. A mathematical model of the hidden feedback loop effect in machine learning systems. *arXiv preprint arXiv:2405.02726*, 2024.

Zhilin Wang, Yi Dong, Olivier Delalleau, Jiaqi Zeng, Gerald Shen, Daniel Egert, Jimmy J. Zhang, Makesh Narsimhan Sreedhar, and Oleksii Kuchaiev. Helpsteer2: Open-source dataset for training top-performing reward models, 2024.

Alexander Wettig, Aatmik Gupta, Saumya Malik, and Danqi Chen. Qurating: Selecting high-quality data for training language models, 2024. URL https://arxiv.org/abs/2402.09739.

Zitong Yang, Neil Band, Shuangping Li, Emmanuel Candès, and Tatsunori Hashimoto. Synthetic continued pretraining, 2024. URL https://arxiv.org/abs/2409.07431.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah D. Goodman. Star: self-taught reasoner bootstrapping reasoning with reasoning. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2024. Curran Associates Inc. ISBN 9781713871088.

A Iterative Gaussian Model Fitting: Mathematical Results and Proofs

A.1 Setup

Lemma 2. *Using the notation of Theorem 1, we can express* $\mu_t = \sum_{r=1}^{t} \sigma_{r-1} \frac{\overline{z_r}}{r} + \mu_0$.

Proof. Note that $X_{i,t} = \mu_{t-1} + \sigma_{t-1} z_{i,t}$, where $z_{i,t} \sim \mathcal{N}(0,1)$. Therefore,

$$\begin{aligned}\mu_t &= \frac{1}{nt}\sum_{r=1}^{t}\sum_{i=1}^{n} X_{i,r} \\ &= \frac{t-1}{t}\mu_{t-1} + \frac{\mu_{t-1}}{t} + \sigma_{t-1}\frac{\overline{z_t}}{t} \\ &= \mu_{t-1} + \sigma_{t-1}\frac{\overline{z_t}}{t}.\end{aligned}$$

Therefore, $\mu_t = \sum_{r=1}^{t} \sigma_{r-1} \cdot \frac{\overline{z_r}}{r} + \mu_0$. □

Lemma 3. *Under the setup described in Theorem 1,* $\mathbb{E}[\frac{\sigma_t^2}{\sigma_0^2}] = \prod_{k=1}^{t}\left(1 - \frac{1}{nk^2}\right) \xrightarrow{t\to\infty} \frac{\sin(\pi/\sqrt{n})}{\pi/\sqrt{n}}$.

Proof. Using the recursive expression for μ_t in Lemma 2, we can rewrite

$$\begin{aligned}\sigma_t^2 &= \frac{1}{nt}\sum_{r=1}^{t}\sum_{i=1}^{n}\left(X_{i,r} - \mu_t\right)^2 \\ &= \frac{1}{nt}\sum_{r=1}^{t}\sum_{i=1}^{n}\left(X_{i,r} - \overline{X_r} + \overline{X}_r - \mu_t\right)^2 \\ &= \frac{1}{nt}\sum_{r=1}^{t}\left(\sum_{i=1}^{n}\left(X_{i,r} - \overline{X_r}\right)^2 + n(\overline{X_r} - \mu_t)^2\right) \\ &= \frac{1}{t}\sum_{r=1}^{t}\left(\sigma_{r-1}^2 S_r^2 + (\mu_{r-1} + \sigma_{r-1}\overline{z_r} - \mu_t)^2\right).\end{aligned}$$

In the last line, we define $S_r^2 = \sum_{i=1}^{n}(z_{i,r} - \overline{z_r})^2$. The term

$$(\mu_{r-1} + \sigma_{r-1}\overline{z_r} - \mu_t)^2 = \left(\sigma_{r-1}\overline{z_r} - \sum_{k=r}^{t}\sigma_{k-1}\cdot\frac{\overline{z_k}}{k}\right)^2,$$

so

$$\begin{aligned}\sigma_t^2 &= \frac{1}{t}\sum_{r=1}^{t}\left(\sigma_{r-1}^2 S_r^2 + \left(\sigma_{r-1}\overline{z_r} - \sum_{k=r}^{t}\sigma_{k-1}\frac{\overline{z_k}}{k}\right)^2\right) \\ \Rightarrow t\sigma_t^2 &= \sum_{r=1}^{t}\left(\sigma_{r-1}^2 S_r^2 + \left(\sigma_{r-1}\overline{z_r}\left(1 - \frac{1}{r}\right) - \sum_{k=r+1}^{t}\sigma_{k-1}\frac{\overline{z_k}}{k}\right)^2\right).\end{aligned}$$

We now compute the conditional expectations of the terms in this sum. Where $\mathcal{F}_i$ denotes the ith filtration,

$$\mathbb{E}[\sigma_{r-1}^2 S_r^2 | \mathcal{F}_{t-1}] = \begin{cases}\sigma_{r-1}^2 S_r^2 & r < t \\ \sigma_{t-1}^2 \cdot \left(\frac{n-1}{n}\right) & r = t.\end{cases}$$

For $r = t$, we find that

$$\mathbb{E}\left[\left(\sigma_{r-1}\overline{z_r}\cdot\left(1 - \frac{1}{r}\right) - \sum_{k=r+1}^{t}\sigma_{k-1}\cdot\frac{\overline{z_k}}{k}\right)^2 | \mathcal{F}_{t-1}\right] = \sigma_{t-1}^2\left(1 - \frac{1}{t}\right)\cdot\frac{1}{n}.$$

On the other hand, when $r < t$,

$$\mathbb{E}\left[\left(\sigma_{r-1}\overline{z_r}\cdot\left(1-\frac{1}{r}\right)-\sum_{k=r+1}^{t-1}\sigma_{k-1}\cdot\frac{\overline{z_k}}{k}-\sigma_{t-1}\cdot\frac{\overline{z}_t}{t}\right)^2|\mathcal{F}_{t-1}\right]$$
$$=\sigma_{t-1}^2\cdot\frac{1}{t^2}\cdot\frac{1}{n}+\left(\sigma_{r-1}\overline{z_r}\cdot\left(1-\frac{1}{r}\right)-\sum_{k=r+1}^{t-1}\sigma_{k-1}\cdot\frac{\overline{z_k}}{k}\right)^2.$$

Therefore,

$$\begin{aligned}\mathbb{E}[t\sigma_t^2|\mathcal{F}_{t-1}] &= (t-1)\sigma_{t-1}^2+\sigma_{t-1}^2\cdot\left(1-\frac{1}{n}\right)+\sigma_{t-1}^2\cdot\left(\frac{t-1}{t}\right)\cdot\left(\frac{1}{n}\right)+\sigma_{t-1}^2\cdot\left(1-\frac{1}{t}\right)^2\cdot\left(\frac{1}{n}\right)\\ &= \sigma_{t-1}^2\left(t-1+1-\frac{1}{n}+\frac{1}{tn}-\frac{1}{t^2n}+\frac{1}{n}-\frac{2}{tn}+\frac{1}{t^2n}\right)\\ &= \sigma_{t-1}^2\left(t-\frac{1}{tn}\right).\end{aligned}$$

It follows that

$$\mathbb{E}[\sigma_t^2|\mathcal{F}_{t-1}] = \sigma_{t-1}^2\left(1-\frac{1}{t^2n}\right) < \sigma_{t-1}^2$$

for all t. Thus, $\{\sigma_t^2\}_t$ is a supermartingale, and

$$\sigma_t^2 \xrightarrow{a.s.} \sigma_\infty^2$$

because σ_t^2 is bounded below by 0. Therefore, we still have convergence. Next, letting $m_t = \mathbb{E}[\sigma_t^2]$, we have

$$m_t = m_{t-1}\left(1-\frac{1}{t^2n}\right) = \cdots = \sigma_0^2\prod_{k=1}^{t}\left(1-\frac{1}{k^2n}\right),$$

so

$$\mathbb{E}[\sigma_t^2] = \sigma_0^2\prod_{k=1}^{\infty}\left(1-\frac{1}{k^2n}\right). \tag{13}$$

By a theorem of Euler, this is equal to

$$\sigma_0^2\frac{\sin(\pi/\sqrt{n})}{\pi/\sqrt{n}}. \tag{14}$$

□

Observe that by performing a variable replacement and using L'Hospital's rule, it is clear that $\lim_{n\to\infty}\mathbb{E}[\sigma_t^2] = \sigma_0^2$.

Finally, we are able to compute $\mathbb{E}[(\mu_t-\mu_0)^2]$.

Corollary 4. *The expected error in the mean*

$$\mathbb{E}[(\mu_t-\mu_0)^2] = \sigma_0^2\left(1-\prod_{k=1}^{t}\left(1-\frac{1}{k^2n}\right)\right). \tag{15}$$

Proof. Using the recursion from Lemma 2 and the expression for the variance in Lemma 6, we can rewrite

$$\begin{aligned}\mathbb{E}[(\mu_t - \mu_0)^2] &= \sum_{k=1}^{t} \frac{\mathbb{E}[\sigma_{k-1}^2]}{nk^2} \\ &= \sigma_0^2 \sum_{k=1}^{t} \frac{1}{k^2 n} \prod_{\ell=1}^{k-1} \left(1 - \frac{1}{\ell^2 n}\right) \\ &= \sigma_0^2 \sum_{k=1}^{t} \left(\prod_{\ell=1}^{k-1} \left((1 - \frac{1}{\ell^2 n}\right) - \prod_{\ell=1}^{k} \left(1 - \frac{1}{\ell^2 n}\right) \right) \\ &= \sigma_0^2 \left(1 - \prod_{k=1}^{t} \left(1 - \frac{1}{k^2 n}\right)\right).\end{aligned}$$

□

Therefore,

$$\lim_{t\to\infty} \mathbb{E}[(\mu_t - \mu_0)^2] = \sigma_0^2 \left(1 - \frac{\sin(\pi/\sqrt{n})}{\pi/\sqrt{n}}\right).$$

B Iterative KDE Fitting: Mathematical Results and Proofs

In this section, we prove that the NLL diverges when iteratively fitting KDE's regardless of whether one accumulates or replaces data from previous iterations.

Theorem 5. *In the replace setting described in Section 3.2, as long as one holds the bandwidth constant, the NLL asymptotically diverges.*

Proof. Define f_0 as the density function for the data distribution from which the original data $x_1, ..., x_n$ are sampled. Define K_h to be the Gaussian kernel function with fixed bandwidth h. One can rewrite the fitted distribution at iteration t as

$$D_t = K_h * D_{t-1}$$

where $*$ denotes the standard convolution of densities.

By a simple recursion, it is clear that $D_t = K^{*t} * D_0$. When two Gaussian kernels with bandwidths a and b are convolved, a basic calculation shows that the resulting effective bandwidth is $\sqrt{a^2 + b^2}$. Consequently, by an inductive argument, the effective bandwidth of K^{*t} is $h\sqrt{t}$. Therefore,

$$\lim_{t\to\infty} K^{*t} * D_0 = \lim_{t\to\infty} K_{h\sqrt{t}} * D_0 = 0$$

because as the bandwidth goes to ∞, the likelihood of any point goes to 0. Hence, regardless of the choice of test data, the negative log likelihood diverges to $-\infty$. □

The same conclusion holds when one accumulates rather than subsampling data:

Theorem 6. *For any non-trivial kernel (i.e. a kernel whose Fourier transform is not* 1*), 3.2, the NLL diverges.*

Proof. We adopt the same notation as in Theorem 5, except this time K denotes a general kernel K that doesn't necessarily need to be Gaussian. In this instance, it is more convenient to work in frequency space, where convolution in probability space corresponds to multiplication.

Define φ_0 as the Fourier transform (FT) of f_0, also called the characteristic function. Let κ denote the FT of K. Then

$$\varphi_t = \kappa \cdot \varphi_{t-1}$$

where $\cdot$ denotes standard complex multiplication. Define $\delta_t = \frac{\phi_t}{\phi_0}$ so that $\varphi_t = \delta_t \cdot \varphi_0$. Define $d_t = \varphi_t/\varphi_0$, and let $a_t = \frac{1}{t}\sum_{i=0}^{t} d_i$. Using this notation,

$$d_t = \kappa \cdot a_{t-1} \tag{16}$$
$$a_t = \left((t-1)a_{t-1} + d_t\right)/t. \tag{17}$$

We see that $a_t = L_{t,K}(a_{t-1})$ is an affine map with slope $((t-1)+\kappa)/t$ and intercept 0. Suppose that the characteristic function of the density converges to φ_∞. Then the map a_t has a fixed point. As long as $\kappa \neq 1$, this fixed point must satisfy the equation

$$\begin{aligned} \varphi &= ((t-1)+\kappa)\varphi \\ \Rightarrow 0 &= ((t-1)+\kappa)/g - 1)\,\varphi \\ \Rightarrow 0 &= (-1+\kappa)\,\varphi \Rightarrow \varphi = 0. \end{aligned}$$

Note that if $\varphi_\infty = 0$, its inverse FT is a function that has 0 probability density everywhere in probability space. Equivalently, the variance of f_t diverges to ∞.

□

Although the NLL eventually diverges in the accumulate case, it is clear from the expression for a_t that this divergence occurs very slowly.

For a Gaussian kernel, both the replace and accumulate case offer an interesting shared insight. Throughout the iterative fitting process, regardless of whether we accumulate or replace, the bandwidth monotonically grows. Therefore, when one starts this process with a very small bandwidth smaller than the optimal bandwidth for the density being fit, one could initially observe a decrease in the negative log likelihood as the bandwidth approaches its optimum.

Finally, model collapse, while inevitable with a fixed bandwidth, can be avoided in all cases by shrinking the bandwidth at a sufficiently fast rate. Since practitioners typically optimize their bandwidth according to the amount of the data that they have, the bandwidth should have the form $c(tn)^{1/5}$ where c is a constant. In this setting, model collapse is avoided entirely.

Theorem 7. *Suppose that data accumulates as in Section 3.2 for a Gaussian kernel. Let the bandwidth at the nth model-fitting iteration be $c(tn)^{-1/5}$ for a constant c. Then the asymptotic variance of the limiting KDE is finite.*

Proof. Let $K_{c(tn)^{-1/5}}$ denote the kernel at the tth model-fitting iteration. Let f_0 denote the original distribution, and define f_t to be the distribution of the KDE at the tth iteration.

We can write

$$\begin{aligned} f_t &= \frac{1}{t}\sum_{i=1}^{t} f_{i-1} * K_{c(in)^{-1/5}} \\ &= \left(1 - \frac{1}{t}\right) \cdot \left(\frac{1}{t-1}\sum_{i=1}^{t-1} f_{i-1} * K_{c(in)^{-1/5}}\right) + \frac{1}{t} f_{t-1} * K_{c(tn)^{-1/5}} \\ &= \left(1 - \frac{1}{t}\right) f_{t-1} + \frac{1}{t} f_{t-1} * K_{c(tn)^{-1/5}} \\ &= \left(\left(1 - \frac{1}{t}\right) K_0 + \frac{1}{t} K_{c(tn)^{-1/5}}\right) \end{aligned}$$

where K_0 is the identity kernel, or equivalently the Gaussian kernel with 0 bandwidth.

Therefore, we find that

$$f_t = f_0 * \circledast_{i=1}^{t} \left(\left(1 - \frac{1}{i}\right) K_0 + \frac{1}{i} K_{c(in)^{-1/5}}\right).$$

Define W_i to be a random variable that is $K_{c(in)^{-1/5}}$ with probability $\frac{1}{i}$ and K_0 with probability $1 - \frac{1}{i}$. We can rewrite X_t, a random variable drawn at the tth fitting iteration as

$$X_t = X_0 + \sum_{i=1}^{t} W_i.$$

All of $X_0, W_1, ..., W_t$ are independent. The variance is given by

$$\begin{aligned}
\text{Var}(X_t) &= \text{Var}(X_0) + \sum_{i=1}^{t} \text{Var}(W_i) \\
&= \text{Var}(X_0) + \sum_{i=1}^{t} \frac{1}{i} \times \frac{c}{(in)^{2/5}} \\
&= \text{Var}(X_0) + \frac{c}{n^{2/5}} \sum_{i=1}^{t} \frac{1}{i^{7/5}}.
\end{aligned}$$

As $t \to \infty$,

$$\text{Var}(X_t) \to \text{Var}(X_0) + \frac{c}{n^{2/5}} \sum_{i=1}^{\infty} \frac{1}{i^4} < \infty.$$

Therefore, when the kernel size is appropriately adjusted, the variance of the KDE under accumulate converges. □

C EXPERIMENTAL RESULTS: SWEEP CONFIGURATIONS

C.1 MODEL COLLAPSE IN MULTIVARIATE GAUSSIAN MODELING

To study model collapse in multivariate Gaussian modeling, we ran the following YAML sweep:

```
program: src/fit_gaussians/fit_gaussians.py
entity: rylan
project: rerevisiting-model-collapse-fit-gaussians
method: grid
parameters:
  data_dim:
    values: [ 1, 3, 10, 31, 100 ]
  num_samples_per_iteration:
    values: [10, 32, 100, 316, 1000]
  num_iterations:
    values: [ 100 ]
  seed:
    values: [ 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19,
  setting:
    values: [
      "Accumulate",
      "Accumulate-Subsample",
      "Replace",
    ]
  sigma_squared:
    values: [
      1.0,
    ]
```

Seeds were swept from 0 to 99, inclusive.

C.2 MODEL COLLAPSE IN KERNEL DENSITY ESTIMATION

To study model collapse in multivariate Gaussian modeling, we ran the following YAML sweep:

Seeds were swept from 0 to 99, inclusive.

C.3 MODEL COLLAPSE IN KERNEL DENSITY ESTIMATION

To study model collapse in kernel density estimation, we ran the following YAML sweep:

```
program: src/fit_kdes/fit_kdes.py
entity: rylan
project: rerevisiting-model-collapse-fit-kdes
method: grid
parameters:
  data_config:
    parameters:
      dataset_name:
        values: ["blobs"]
      dataset_kwargs:
        parameters:
          n_features:
            values: [2]
  kernel:
    values: ["gaussian"]
  kernel_bandwidth:
    values: [0.1, 0.5, 1.0]
```

```
    num_samples_per_iteration:
      values: [10, 32, 100, 316, 1000]
    num_iterations:
      values: [ 100 ]
    seed:
      values: [ 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19,
    setting:
      values: [
        "Accumulate",
        "Accumulate-Subsample",
        "Replace",
      ]

  program: src/fit_kdes/fit_kdes.py
  entity: rylan
  project: rerevisiting-model-collapse-fit-kdes
  method: grid
  parameters:
    data_config:
      parameters:
        dataset_name:
          values: ["circles"]
        dataset_kwargs:
          parameters:
            noise:
              values: [0.05]
    kernel:
      values: ["gaussian"]
    kernel_bandwidth:
      values: [0.1, 0.5, 1.0]
    num_samples_per_iteration:
      values: [10, 32, 100, 316, 1000]
    num_iterations:
      values: [ 100 ]
    seed:
      values: [ 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19,
    setting:
      values: [
        "Accumulate",
        "Accumulate-Subsample",
        "Replace",
      ]

  program: src/fit_kdes/fit_kdes.py
  entity: rylan
  project: rerevisiting-model-collapse-fit-kdes
  method: grid
  parameters:
    data_config:
      parameters:
        dataset_name:
          values: ["moons"]
        dataset_kwargs:
          parameters:
            noise:
              values: [0.05]
    kernel:
      values: ["gaussian"]
    kernel_bandwidth:
```

```
      values: [0.1, 0.5, 1.0]
    num_samples_per_iteration:
      values: [10, 32, 100, 316, 1000]
    num_iterations:
      values: [ 100 ]
    seed:
      values: [ 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19,
    setting:
      values: [
        "Accumulate",
        "Accumulate-Subsample",
        "Replace",
      ]

program: src/fit_kdes/fit_kdes.py
entity: rylan
project: rerevisiting-model-collapse-fit-kdes
method: grid
parameters:
  data_config:
    parameters:
      dataset_name:
        values: ["swiss_roll"]
      dataset_kwargs:
        parameters:
          noise:
            values: [0.05]
  kernel:
    values: ["gaussian"]
  kernel_bandwidth:
    values: [0.1, 0.5, 1.0]
  num_samples_per_iteration:
    values: [10, 32, 100, 316, 1000]
  num_iterations:
    values: [ 100 ]
  seed:
    values: [ 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19,
  setting:
    values: [
      "Accumulate",
      "Accumulate-Subsample",
      "Replace",
    ]
```

Seeds were swept from 0 to 99, inclusive.

C.4 Model Collapse in Linear Regression

To study model collapse in linear regression, we ran the following YAML sweep:

```
program: src/fit_linear_regressions/fit_linear_regressions.py
entity: rylan
project: rerevisiting-model-collapse-fit-lin-regr
method: grid
parameters:
  data_dim:
    values: [ 100, 10, 31, 3, 1 ]
  num_samples_per_iteration:
    values: [10, 32, 100, 316, 1000]
```

```
  num_iterations:
    values: [ 100 ]
  seed:
    values: [ 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19,
  setting:
    values: [
      "Accumulate",
      "Accumulate-Subsample",
      "Replace",
    ]
  sigma_squared:
    values: [
      0.1, 1.0, 10.
    ]
```

Seeds were swept from 0 to 99, inclusive. Note: We ran this sweep as 9 separate sweeps; to understand why, see this GitHub issue.

D Additional Experimental Results for Model Collapse Hyperparameters

Due to space limitations in the main text, we can oftentimes only present a subset of runs corresponding to a subset of hyperparameters. We present additional figures with a wide range of hyperparameters here for completeness.

D.1 Additional Results for Model Collapse in Linear Regression

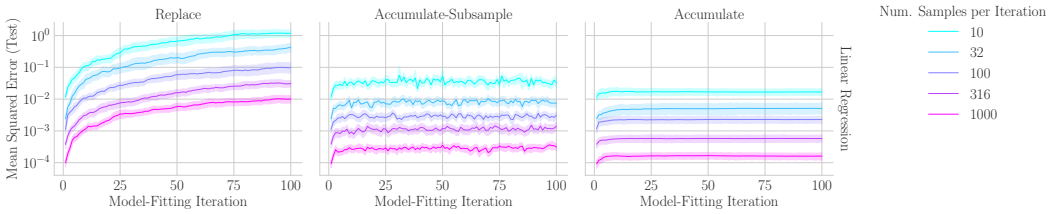

Figure 6: Linear Regression for Data Dimension $d = 1$ and variance $\sigma^2 = 0.10$.

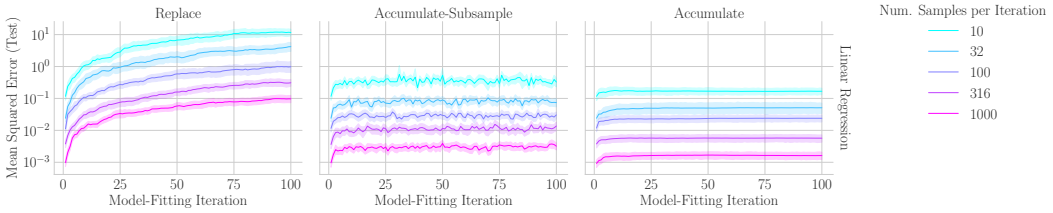

Figure 7: Linear Regression for Data Dimension $d = 1$ and variance $\sigma^2 = 1.00$.

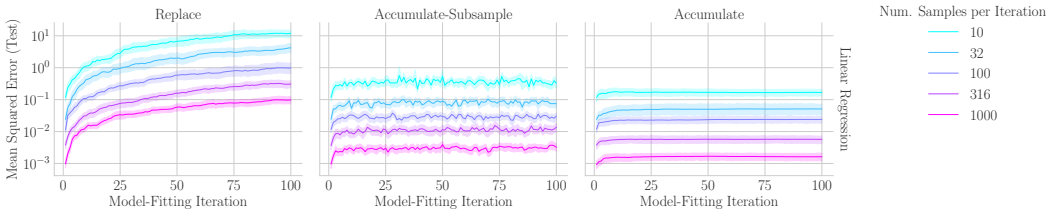

Figure 8: Linear Regression for Data Dimension $d = 1$ and variance $\sigma^2 = 10.0$.

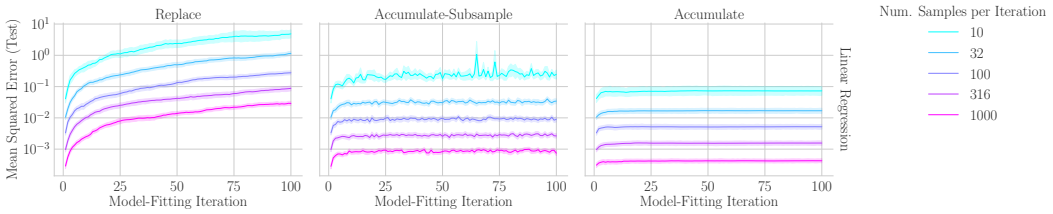

Figure 9: Linear Regression for Data Dimension $d = 3$ and variance $\sigma^2 = 0.10$.

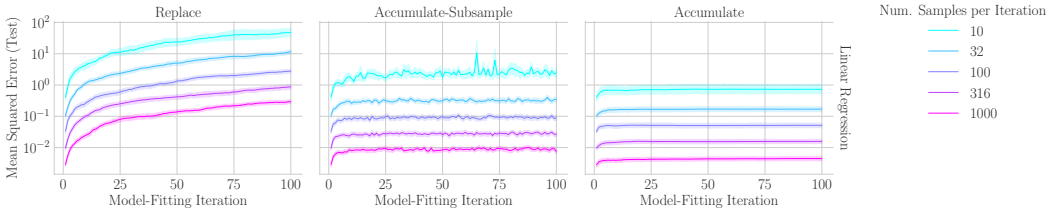

Figure 10: Linear Regression for Data Dimension $d = 3$ and variance $\sigma^2 = 1.00$.

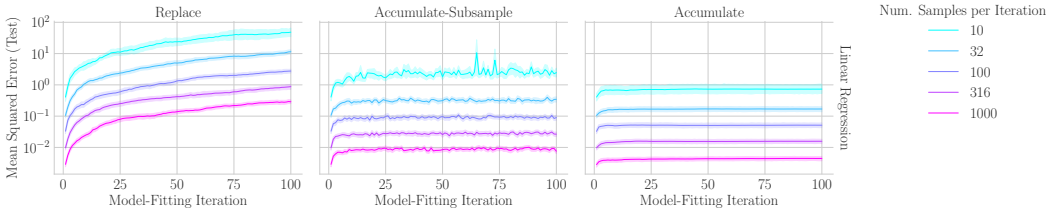

Figure 11: Linear Regression for Data Dimension $d = 3$ and variance $\sigma^2 = 10.0$.

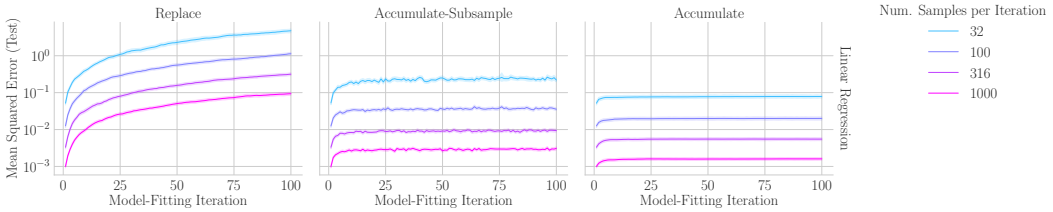

Figure 12: Linear Regression for Data Dimension $d = 10$ and variance $\sigma^2 = 0.10$.

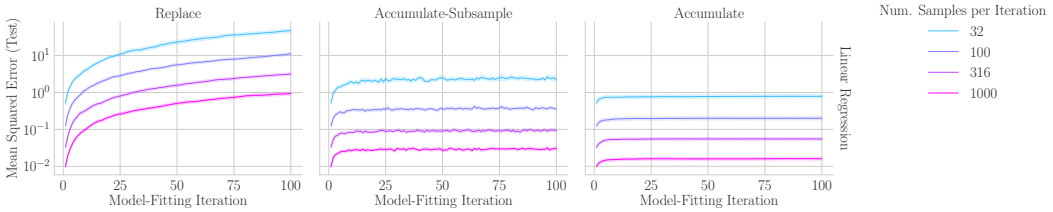

Figure 13: Linear Regression for Data Dimension $d = 10$ and variance $\sigma^2 = 1.00$.

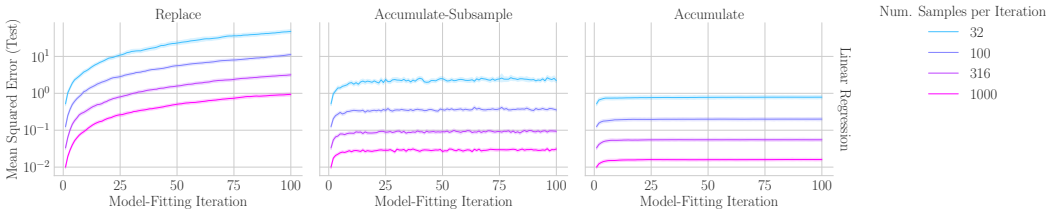

Figure 14: Linear Regression for Data Dimension $d = 10$ and variance $\sigma^2 = 10.0$.

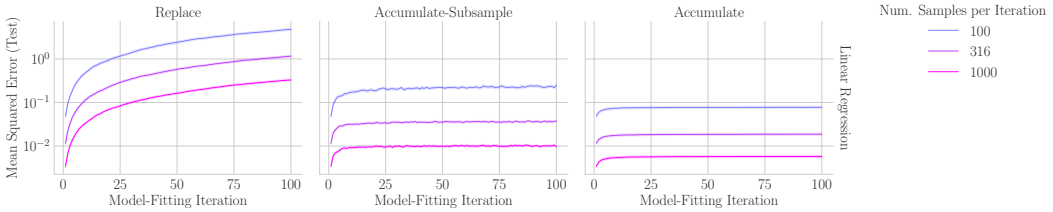

Figure 15: Linear Regression for Data Dimension $d = 32$ and variance $\sigma^2 = 0.10$.

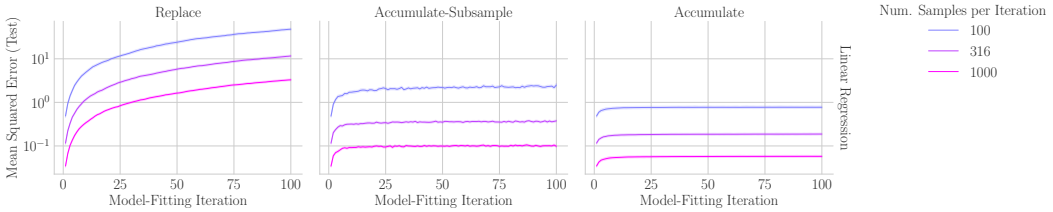

Figure 16: Linear Regression for Data Dimension $d = 32$ and variance $\sigma^2 = 1.00$.

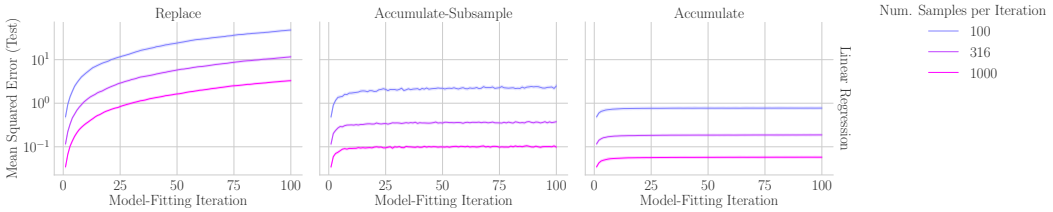

Figure 17: Linear Regression for Data Dimension $d = 32$ and variance $\sigma^2 = 10.0$.

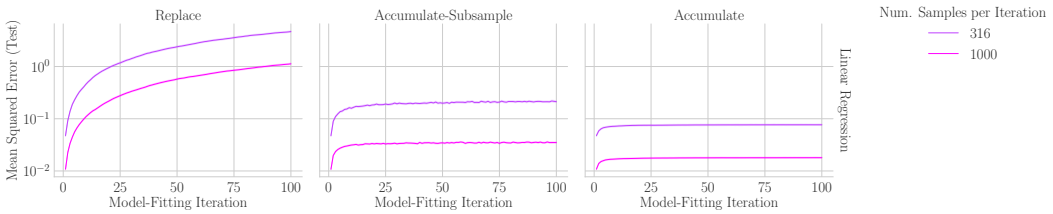

Figure 18: Linear Regression for Data Dimension $d = 100$ and variance $\sigma^2 = 0.10$.

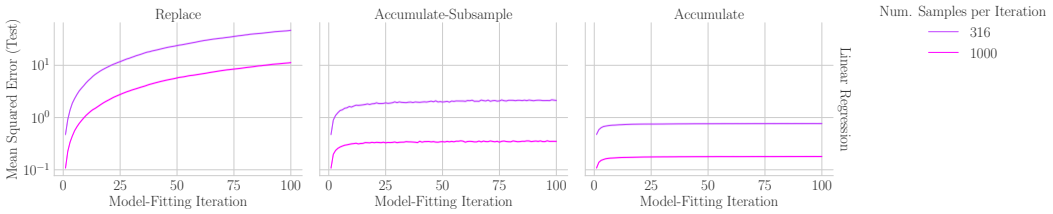

Figure 19: Linear Regression for Data Dimension $d = 100$ and variance $\sigma^2 = 1.00$.

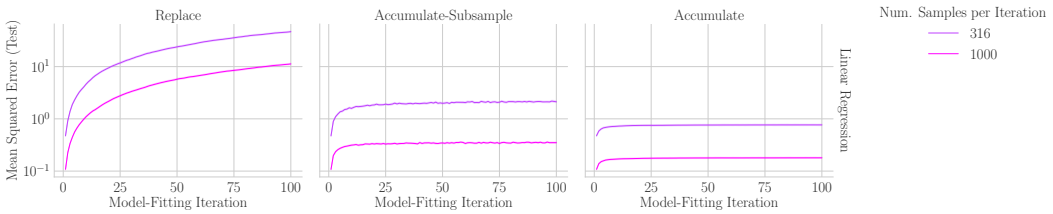

Figure 20: Linear Regression for Data Dimension $d = 100$ and variance $\sigma^2 = 10.0$.