ForTIFAI: Fending Off Recursive Training Induced Failure for AI Models

Soheil Zibakhsh Shabgahi[1*†], Pedram Aghazadeh[1*†], Azalia Mirhoseini[2] and Farinaz Koushanfar[1]

[1*]Department of Electrical and Computer Engineering, UC San Diego, 9500 Gilman Dr., La Jolla, 92093, CA, USA.
[2]Department of Computer Science, Stanford University, 353 Jane Stanford Way, Stanford, 94305 , CA, USA.

*Corresponding author(s). E-mail(s): szibakhshshabgahi@ucsd.edu; paghazadeh@ucsd.edu;
Contributing authors: azalia@stanford.edu; farinaz@ucsd.edu;
†These authors contributed equally to this work.

Abstract

The increasing reliance on generative AI models has accelerated the generation rate of synthetic data, with some projections suggesting that most available new data for training could be machine-generated by 2030 [1]. This shift to a mainly synthetic content presents a critical challenge: repeated training in synthetic data leads to a phenomenon known as model collapse, where model performance degrades over generations of training, eventually rendering the models ineffective. Although prior studies have explored the causes and detection of model collapse, existing mitigation strategies remain limited.

In this paper, we identify model overconfidence in their self-generated data as a key driver of collapse. Building on this observation, we propose a confidence-aware loss function that downweights high-confidence predictions during training. We introduce a novel loss function we call Truncated Cross Entropy (TCE). We demonstrate that TCE significantly delays model collapse in recursive training. We provide a model-agnostic framework that links the loss function design to model collapse mitigation and validate our approach both theoretically and empirically, showing that it can extend the model's fidelity interval before collapse by more than **2.3**×. Finally, we show that our method generalizes across modalities. These findings suggest that the design of loss functions provides a simple yet powerful tool for preserving the quality of generative models in the era of increasing synthetic data.

1 Introduction

Generative models have become the foundation for modern AI applications in several modalities, including text, image, code, and audio. Large Language Models (LLMs) such as ChatGPT [2], LLaMA [3], Phi [4], and DeepSeek [5], as well as image generators DALL-E [6] and Imagen [7], all rely on large datasets scraped from the Web. As these models are continuously updated to reflect recent knowledge and linguistic patterns, the need for ever larger and frequently refreshed training corpora has grown substantially. However, this demand is colliding with a shift in the data landscape: synthetic content is increasingly populating the Internet, contaminating the very datasets used for model training.

This shift raises fundamental concerns. Generative models trained on outputs from earlier generations can degrade over time, a phenomenon known as *model collapse* [8–10]. Even moderate levels of synthetic data (as little as 1%) have been shown to cause collapse, and scaling the model or the size of the dataset fails to reliably prevent it [11]. Importantly, this is not a hypothetical scenario: recent analysis of the LAION-5B dataset [12], a cornerstone for many generative models, revealed measurable contamination from synthetic sources.[1] As synthetic data becomes dominant, safeguarding generative model stability becomes an urgent challenge.

Several approaches have been explored to mitigate model collapse. Some studies combine synthetic and real data to slow degradation [13]; others rely on post hoc supervision, such as SIMS [14], using labeled synthetic data as negative signal to improve generations or explore decoding strategies [15]. In practice, however, distinguishing real from synthetic data remains a major challenge. Watermarking methods [16, 17] have been proposed to help identify model-generated content, but they are not widely adopted. More fundamentally, recent critiques [18] highlight that many prior studies simulate collapse in unrealistic settings, such as using purely synthetic training data, limiting the applicability of their conclusions. Despite growing interest in this issue, the role of the loss function, central to how models learn, has received little attention in the context of model collapse.

In this work, we present ForTIFAI, showing that model collapse can be mitigated through principled modifications to training loss. We argue that generative models tend to be overconfident in their synthetic predictions (as depicted in Figure 1), causing feedback loops that degrade performance. This issue is further amplified by approximation errors and finite sampling effects during training, which cause models to concentrate probability mass on a shrinking subset of the true data distribution over generations [9, 10]. To counter this, we introduce Truncated Cross Entropy (TCE), a novel confidence-aware loss function that reduces the weight of high-confidence predictions during training by masking predictions above a confidence threshold. TCE is model-agnostic and easy to implement on top of Cross Entropy (CE), making it attractive for real-world deployment.

[1] Following prior works [8, 9], we use the term *contamination* or *poisoning* to describe the recursive use of synthetic data in training, which leads to degradation in performance over generations.

In brief, the contributions of ForTIFAI are as follows:

- We introduce a loss-based, model-agnostic method for mitigating model collapse, explicitly addressing sampling bias identified in prior studies.
- We propose Truncated Cross Entropy (TCE), a novel confidence-aware loss function that downweights high-confidence tokens during training, and demonstrate that it extends model fidelity by more than 2.3×.
- We design a comprehensive evaluation framework grounded in practical and evolving data regimes, including datasets where new real-world content continues to appear, some content is regenerated multiple times while other parts remain untouched, and model quality is measured by retained knowledge rather than token similarity.
- We demonstrate the generality of our approach by applying it to diverse generative models, including Gaussian Mixture Models (GMMs) and Variational Autoencoders (VAEs), confirming its effectiveness beyond LLMs.

2 Method

In this section, we first describe the *Model Collapse* phenomena. We then provide intuition about the correlation between model collapse and token prediction confidence[2] in Large Language Models (LLMs). Building upon this intuition, we derive a class of confidence-aware loss functions to thwart autophagy [12] in generative models.

2.1 Problem Definition

Model Collapse is the gradual degradation that occurs when a model's own outputs pollute the training data of its successors. Each new model then learns from these possibly flawed examples, causing a drift from the true data distribution [8].

During the generation phase, statistical approximations, limited functional expressiveness, and approximation errors introduce non-zero probabilities of distributional shifts in the model's outputs.

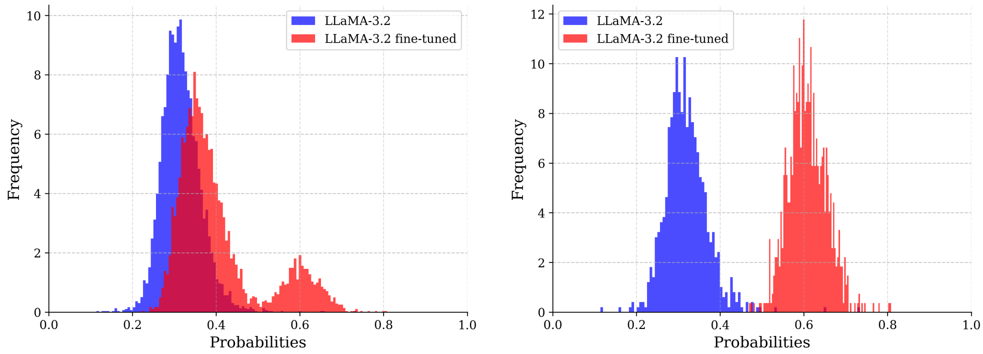
2.1.1 Key Observation

Figure 1 illustrates a consistent gap in model confidence between self-generated samples and unseen data. We exploit this difference as a signal to identify and discount synthetic samples during training, allowing the model to reduce their influence and mitigate collapse. This gap is a side effect of the approximation error and the greedy sampling performed for generating text [15].

2.2 Confidence-aware Loss Functions

Building on Sec. 2.1.1, we propose confidence-aware loss functions that reduce the influence of overconfident predictions, typically associated with synthetic tokens. By selectively down-weighting, or ignoring, model-generated samples during training using their statistical properties, we mitigate their harmful influence on the loss and reduce the risk of collapse.

[2] Throughout this paper, we define a model's confidence in tokens as the predicted probability assigned to them.



(a) Confidences over Wikitext

(b) Confidences over re-generated Wikitext

Fig. 1: Models display higher confidence in their own generated data when recursively trained. These charts illustrate the probabilities assigned by the LLaMA-3.2-1B model and a version fine-tuned on a fraction of Wikitext. The synthetic dataset in this case is generated by the fine-tuned model. The fine-tuned model shows higher confidence on the portion of the Wikitext it was trained on as well as the entire dataset generated by it. In contrast, the original model demonstrates similar confidence in both cases. This behavioral analysis was key to finding mitigations discussed in the following sections of ForTIFAI.

2.2.1 Truncated Cross Entropy (TCE)

Building on the findings from Section 2.1.1, we introduce a novel loss function, Truncated Cross Entropy (TCE). In TCE, confident predictions are explicitly masked to remove their influence on training. By selectively masking overly confident predictions, TCE forces the model to learn from less-certain datapoints, thereby preserving distribution tails compared to CE. We formulate TCE as:

$$TCE(p_t) = \chi_\gamma(p_t) \times CE(p_t)$$

$$\chi_\gamma(p_t) = \begin{cases} 1 \;\; if\; p_t \leq \gamma \\ 0 \;\; if\; p_t > \gamma \end{cases}$$

Here, p_t is the probability of the correct class, $\chi_\gamma(.)$ is an indicator function, $CE(.)$ is the cross-entropy loss, and $\gamma \in [0, 1]$ is the confidence threshold. Setting γ close to 1 recovers standard CE loss, while smaller values selectively suppress high-confidence predictions. In Appendix A we examine Focal Loss as another alternative confidence-aware loss function with similar properties.

2.3 Mathematical Intuition

The growing reliance on synthetic data in generative model training pipelines has given rise to a self-consuming feedback loop wherein models are recursively trained on their own outputs.

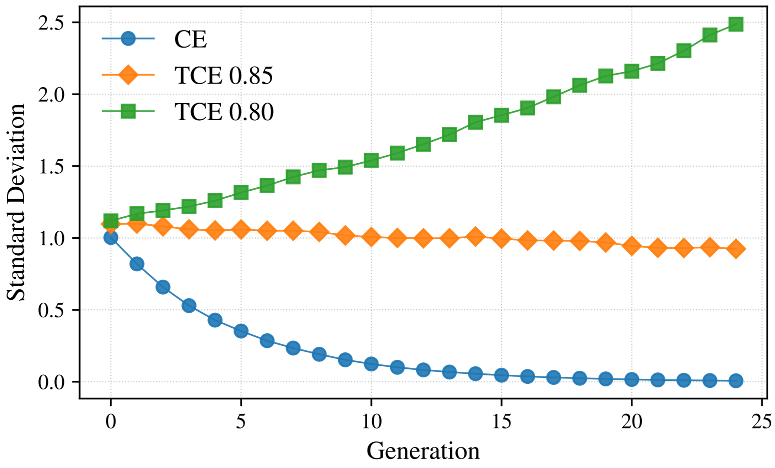


Fig. 2: Effect of TCE on a one-dimensional Gaussian estimator under a fully synthetic training loop. In each generation, $10,000$ samples are generated with sampling bias $\gamma = 0.9$. Standard estimation with Cross Entropy (CE) leads to rapid variance collapse. By contrast, TCE with a well-chosen threshold ($\gamma = 0.85$) delays collapse substantially. Conversely, if the threshold is set poorly ($\gamma = 0.80$), the variance may diverge or converge to zero prematurely.

In a fully synthetic training loop, where each model G_t is trained solely on samples from G_{t-1}, Alemohammad et al[12] prove that the model's estimated covariance σ_t collapses almost surely:

$$\mathbb{E}[\sigma_t \mid \sigma_{t-1}] = \lambda \sigma_{t-1}, \quad \text{with } \lambda \leq 1, \qquad \Rightarrow \quad \sigma_t \xrightarrow{a.s.} 0. \tag{1}$$

where λ represents sampling bias, contracting the variance at each iteration, leading to rapid loss of sample diversity, as illustrated in Figure 2. This decrease in variance is empirically observed across models such as DDPMs [19] and StyleGAN-2 [20].

To build intuition on how to counteract this collapse, we analyze a one-dimensional Gaussian model where data is drawn from $X_t \sim \mathcal{N}(0, \sigma_t^2)$. Our strategy is to modify the training objective to focus on low-probability samples. In our Gaussian case, this is equivalent to training only on samples from the tails of the distribution, i.e., where $|X_t| \geq a\sigma_t$ for a chosen threshold $a > 0$.

The conditional variance of this truncated distribution is given by:

$$\mathrm{Var}(X_t \mid |X_t| \geq a\sigma_t) = \eta(a) \cdot \sigma_t^2, \tag{2}$$

where $\eta(a)$ is a variance amplification factor. Using known properties of the truncated normal distribution, this factor is calculated as:

$$\eta(a) = 1 + \frac{a\,\phi(a)}{1 - \Phi(a)} \tag{3}$$

where $\phi(\cdot)$ and $\Phi(\cdot)$ are the standard normal PDF and CDF, respectively. Crucially, $\eta(a) > 1$ for all $a > 0$.

By incorporating this variance amplification into the self-consuming loop, the recurrence for the variance becomes:

$$\mathbb{E}[\sigma_{t+1}^2] = \lambda \cdot \eta(a) \cdot \mathbb{E}[\sigma_t^2]. \tag{4}$$

The collapse can now be slowed by choosing a threshold a such that the combined factor $\lambda \cdot \eta(a) \approx 1$, stabilizing the training process by preserving sample diversity.

We argue that overconfident predictions on common tokens can dominate the training signal, ultimately causing the model to under-represent less frequent but meaningful patterns. If not addressed, this effect progressively narrows the model's output distribution, highlighting the importance of suppressing overconfident learning.

By generalizing this principle from a Gaussian model to general generative models, we propose augmenting loss functions to address this issue. **Truncated Cross Entropy (TCE)** explicitly masks the contribution of high-confidence predictions. This shifts the training signal toward low-confidence, often underrepresented tokens, mitigating the tail-vanishing effect and reducing the recursive amplification of statistical errors highlighted above.

3 Model Collapse Evaluation Framework

To realistically assess the impact of *Model Collapse*, we adopt an experimental framework inspired by [13]. Initially, web data was predominantly human-generated and often curated for quality. Over time, however, machine-generated content, often indistinguishable from human-written text, has increasingly supplemented online data. Our framework simulates a realistic model collapse process by progressively adding synthetic data to the training set. We evaluate the methods on math, reasoning, and knowledge recall benchmark performance. Moreover, we analyze the number of synthetic training iterations it takes for the model to lose performance below a certain threshold, which we call the time to failure.

As illustrated in Figure 3, the experiment proceeds in multiple stages. At each stage, the dataset simulates a large-scale pretraining corpus, composed of both authentic and self-generated data. At every step, we regenerate the previous stage's dataset using the latest model and concatenate the dataset with a fixed amount of new human-written text. This new dataset is then used to further train the model, as detailed further in Algorithm 1.

This setup also follows the conditions mentioned in [18]:

1. **Increasing Pre-training Data:** Unlike previous studies with fixed dataset sizes, our setup continuously increases the pre-training data at each generation. This growth mirrors real-world trends among state-of-the-art models; for example, LLaMA 1, 2, and 3 were trained on 1.4, 2, and 15 trillion tokens, respectively. Although dataset growth has eventual limits, these bounds remain extremely high (approximately one quadrillion tokens) [21], far exceeding current dataset sizes.

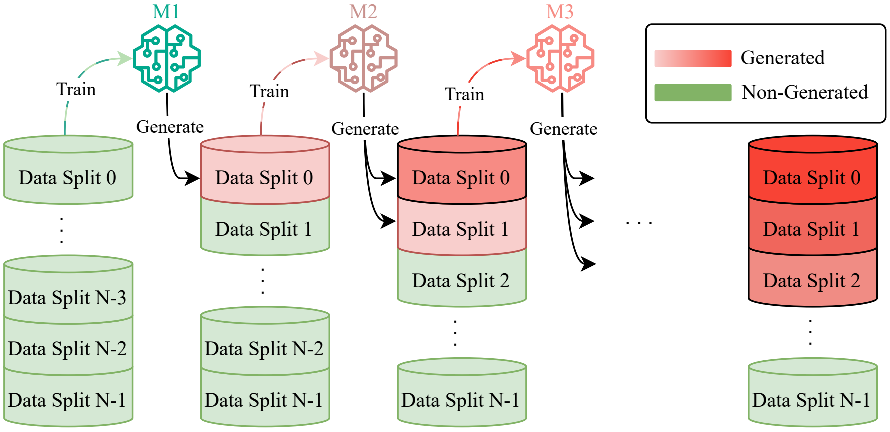


Fig. 3: Our experimental setup simulates model collapse, illustrating the transition from predominantly human-generated content to mostly synthetic datasets. M_i denotes the i-th generation of recursively trained models. The clean data is initially split into N equally sized splits. At each generation, the entire dataset from the previous iteration is regenerated and used as training data for the next iteration. One split of clean data is added to this training dataset to simulate the accumulation of non-generated data into the training set. The red color for each data split indicates the contamination from recursively training on the self-generated data, resulting in lower quality (darker red color)

2. **Synthetic Data Accumulation Alongside Real Data:** Another distinctive feature of our framework is its simultaneous training on both real and synthetic data. Unlike previous work, we do not discount the effect of real unseen data on model collapse. We argue that the real unseen data is the main reason behind training the new generation of models in the first place.
3. **Decreasing Proportion of Real Data:** A primary concern highlighted in recent literature is the accelerating generation rate of synthetic data and its impact on future generative models. Our experimental setup realistically emulates this crucial aspect (e.g. in our setup with 6 stages, we have 100%, 50%, 33%, 25%, 20%, and 15% real data at each stage, respectively).

4 Results

To demonstrate the effectiveness of `Truncated Cross Entropy`, we evaluate it across three settings: Transformers, Variational Autoencoders, and Gaussian Mixture Models. For Transformers, we focus on language modeling, testing LLaMA 3 [3] and Gemma 3 [22] on benchmarks for mathematical reasoning, logical inference, and factual recall. For image generation, we use Variational Autoencoders, and for mixture modeling, we

Algorithm 1 Recursive data generation with an LLM

```
Inputs:
  $N \geq 2$
  $N+1$ splits of authentic dataset $D_0^{(0)}, \dots, D_N^{(0)}$, each of equal size
                                  ▷ $D_i^{(j)}$ is the $j$-th generation of the $i$-th split.
  $N+1$ evaluation sets $Eval_0, \dots, Eval_N$, each of equal size
                                  ▷ $Eval_i$ is based on the content of $D_i^{(0)}$.
$M_0 \Leftarrow$ Pretrained model
for $i = 0 \to N$ do
    Train $M_i$ on $[D_0^{(i)}, D_1^{(i-1)}, \dots, D_i^{(0)}]$
    for $k = 0 \to i$ do
        $D_k^{(i+1-k)} \Leftarrow M_i(D_k^{(i-k)})$
        Evaluate $M_i$ on $Eval_k$
    end for
    $M_{i+1} \Leftarrow M_i$
end for
```

evaluate standard GMMs. Across all experiments, our loss function mitigates model collapse without degrading overall training performance.

4.1 Experiment Details

We apply the recursive training setup illustrated in Figure 3 on two datasets: Wikitext-2-raw-v1 [23] and a newly curated dataset called *Imagination of Web*.

4.1.1 Datasets

Wikitext-2-raw-v1 is a language modeling dataset derived from high-quality Wikipedia articles. Unlike its tokenized counterpart, it preserves raw formatting—including punctuation, casing, and whitespace—making it well-suited for evaluating a model's ability to handle long-range dependencies and natural language structure.

The second dataset, *Imagination of Web*, is a composite dataset constructed from subsets of Wikitext, HellaSwag [24], and GSM8k [25]. It is designed to test the generalization of our loss functions across multiple tasks, such as commonsense, math, and factual recall.

4.1.2 Framework Parameters

We used the framework described in Section 3 to simulate a progressive model collapse scenario.

For our experiments, we set the number of stages to $N = 6$. We found that model collapse was not reliably observable for fewer than six stages, while increasing N beyond six introduced significant computational overhead with diminishing returns under our resource constraints. For simplicity, we used a fixed amount of data at each stage from the original dataset.

4.1.3 Hyperparameters

For both LLaMA and Gemma model families, we tuned and used the best-performing hyperparameters. All models were fine-tuned using a learning rate of 2×10^{-5}, with a batch size of 64. The training was conducted over 10 epochs per stage with the best performing checkpoints saved.

4.2 Evaluation Benchmarks

To comprehensively assess the impact of our loss functions on model performance and resilience to model collapse, we perform evaluations on a mix of standard and custom benchmarks. These benchmarks test various capabilities including factual recall, linguistic coherence, commonsense, and mathematical reasoning. In addition to widely used benchmarks like BLiMP, Hellaswag, and GSM8K, we introduce a new question-answering benchmark—*Knowledge Retention test* (KR-test) to measure how well models retain training-set facts.

4.2.1 Knowledge Retention Test

Validation perplexity is a common metric for evaluating LLMs, but it primarily reflects predictive fluency; how well a model guesses the next token in unseen text. While useful for assessing grammar and style, it does not quantify factual knowledge retention. Moreover, low perplexity does not guarantee meaningful or human-like output, as humans do not always select the most likely next word [26, 27].

To address this gap, we introduce the *Knowledge Retention Test* (KR-test), which measures factual *retention* from the training data rather than general fluency.

Given a body of training text, we divide it into complete paragraphs and generate five questions per paragraph. Each question consists of a context, a true continuation, and a false continuation.

We compute the log probability of the model for two sequences:

- *Context + Factually correct completion*
- *Context + Factually incorrect completion*

When the log probability is higher for the true continuation, the question is considered correctly answered. The KR-test score is the model's total accuracy over all the questions.

A high KR-test score indicates that the model has internalized the factual content of the paragraph. This benchmark specifically tests factual retention from the training data. More details on KR-test can be found in Appendix B. In Fig. A1 superiority of TCE compared to the baseline is shown after six iterations in correctly answering questions about Wikitext.

4.2.2 Math, Reasoning, and Coherence

In addition to factual retention, we evaluate model's performance across three reasoning and language understanding benchmarks. These evaluations assess whether our loss functions preserve or enhance generalization and linguistic structure, both of which are vulnerable under recursive training regimes.

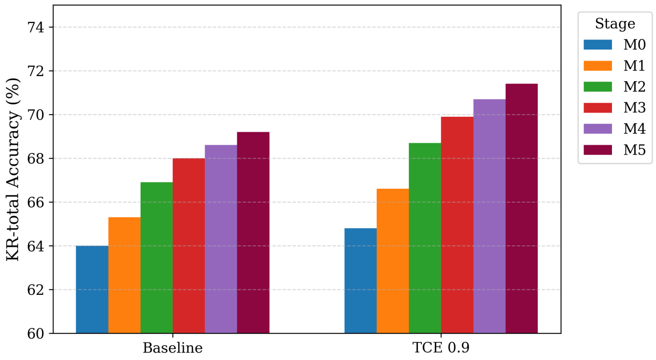


Fig. 4: Total knowledge retention (KR-total) accuracy with LLaMA-3.2-1B model trained on Wikitext. TCE retains learning capability and even exceed the baseline throughout different stages.

To evaluate the impact of model collapse on reasoning and linguistic capability, we use three standard benchmarks: GSM8K [25], Hellaswag [24], and BLiMP [28].

- **GSM8K** measures mathematical reasoning by testing the model's ability to solve grade-school level word problems. Success on this benchmark reflects the model's ability to perform multi-step arithmetic reasoning.
- **Hellaswag** evaluates commonsense reasoning through multiple-choice sentence completion tasks. It requires the model to infer plausible continuations based on everyday scenarios.
- **BLiMP** tests grammatical coherence and syntactic structure. It consists of minimal pairs where the model must choose the grammatically correct option. High performance indicates strong control over syntax and language structure.

Together, these benchmarks assess the model's ability to generalize, reason, and maintain coherence under different forms of linguistic stress. For GSM8K and Hellaswag, we use the Language Model Evaluation Harness [29] to ensure consistent evaluation across models.

4.3 Wikitext dataset

To assess performance under idealized, non-collapsing conditions, we also evaluate models trained exclusively on original, non-synthetic data. The results, shown in Table 1, confirm that `FL` and `TCE 0.9` maintain strong performance in streamline training settings.

Table 2 presents the results after 5 stages of synthetic data accumulation as discussed in Section 3. The evaluation spans across the full benchmark suite, excluding GSM, since the Wikitext training dataset does not include any math or reasoning. On average, `FL` outperforms `CE` by 3.8% for LLaMA-3.2-1B and 1.3% for Gemma-3-1b-pt. Overall, `TCE 0.9` outperforms `CE`, especially in preserving factual knowledge.

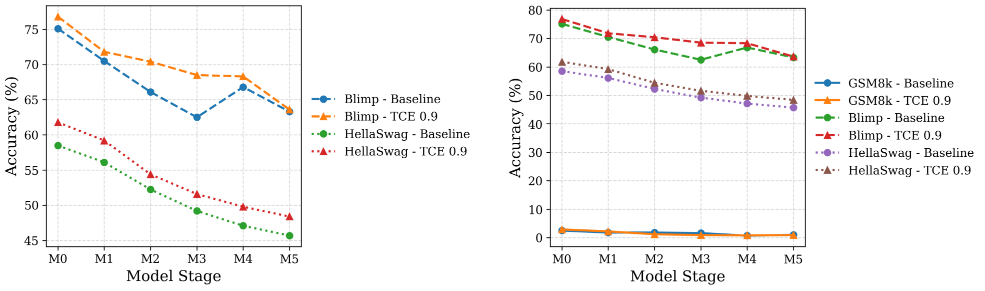
Table 1: Comparison of LLaMA and Gemma models on M0 (clean and non-generated data) across Blimp, Hellaswag, GSM8k, and KR-test on *Wikitext dataset*. For LLaMA and Gemma, TCE outperforms baseline (CE) when trained on real non-generated data. The highest value per block is **bolded**.

Model	Method	Blimp	Hellaswag	KR-test	Average
LLaMA	Baseline	75.1%	58.5%	64.0%	65.86%
	TCE 0.9	**76.8%**	**61.8%**	**64.8%**	**67.8%**
Gemma	Baseline	71.2%	63.9%	67.7%	67.6%
	TCE 0.9	**73.9%**	**64.6%**	**68.2%**	**68.9%**

Table 2: Comparison of M5 benchmark results across LLaMA and Gemma models for Baseline, and TCE 0.9 methods on *Wikitext dataset*. The highest value per block is **bolded**.

Model	Method	Blimp	Hellaswag	KR-test	Average
LLaMA	Baseline	63.3%	45.7%	69.2%	59.4%
	TCE 0.9	**63.6%**	**48.4%**	**71.4%**	**61.13%**
Gemma	Baseline	61.2%	52.4%	68.1%	60.56%
	TCE 0.9	**61.8%**	**53%**	**69.7%**	**61.5%**

Performance of both models when experiment is conducted on Wikitext dataset is shown in Fig. 5 where TCE consistently outperforms **CE** during different stages of the experiment.



(a) Wikitext

(b) Imagination

Fig. 5: Our proposed solution outperforms baselines (CE) across all benchmarks consistently when experiment is done on both datasets. Figures (a) and (b) show the results of LLaMA-3.2-1B across different benchmarks.

4.4 Imagination-of-Web dataset

To evaluate in a setting closer to real-world pre-training, we adopt the *Imagination-of-Web* (IoW) corpus—a composite of factual prose (Wikitext), commonsense reasoning (HellaSwag), and mathematical problem solving (GSM8K).

Table 3 reports clean-data results. Here, TCE surpasses CE by 1.8% on LLaMA-3.2-1B and 1.3% on Gemma-3-1b-pt, confirming the strong learning capability of performance under non-collapsing conditions.

Table 4 shows outcomes after five rounds of synthetic data accumulation. TCE 0.9 outperforms CE by 3.6% on LLaMA-3.2-1B and 0.4% on Gemma-3-1b-pt. As on Wikitext, TCE preserves factual and reasoning accuracy, reinforcing its robustness against model collapse in heterogeneous training scenarios.

Table 3: Comparison of LLaMA and Gemma models on M0 (clean and non-generated data) across Blimp, Hellaswag, GSM8k, and KR-total on *Imagination dataset*. The highest value per block is **bolded**, and the second highest is underlined.

Model	Method	Blimp	Hellaswag	GSM8k	KR-total	Average
LLaMA	Baseline	76.3%	63.1%	8.1%	63.6%	58.78%
	Focal ($\gamma = 2$)	**79.6%**	**65.2%**	**9.5%**	**64.8%**	**60.62%**
	TCE 0.9	76.8%	64.6%	9.1%	64.8%	59.52%
Gemma	Baseline	73.1%	65%	**6.2%**	64.1%	52.1%
	Focal ($\gamma = 2$)	**77.2%**	**65.7%**	4.5%	64.2%	**52.9%**
	TCE 0.9	76%	65%	5.2%	64%	52.55%

Table 4: Comparison of M5 benchmark results across LLaMA and Gemma models for Baseline and TCE 0.9 method on *Imagination dataset*. The highest value per block is **bolded**.

Model	Method	Blimp	Hellaswag	GSM8k	KR-total	Average
LLaMA	Baseline	72.1%	52.0%	5.9%	68.7%	52.94%
	TCE 0.9	**77.6%**	**54.2%**	**6.2%**	**71.0%**	**55.52%**
Gemma	Baseline	71%	**53.4%**	3.9%	71.6%	49.97%
	TCE 0.9	**71.7%**	53.1%	**4.2%**	**74.1%**	**50.77%**

Performance of both models when experiment is conducted on Imagination dataset is shown in Fig. 5, where TCE consistently outperform CE during different stages of the experiment.

4.5 Time to failure

We define *time to failure* as the number of re-training tokens it takes a model in the context of this framework to degrade significantly compared to the initial model.

Specifically, we analyze the performance of the model on the first partition of the dataset, which is re-generated at each stage. Moreover, since the KR-test on average has a baseline of 50% and a maximum of 100% is achievable, we define failure as performance dropping below 75%, which is exactly half the maximum gain one can expect from training the model. As illustrated in Figure 6, Cross Entropy collapses (reaches 75% accuracy) between the first and the second generation and both TCE reaches this point after three stages of self-consuming generation, increasing the time to failure by 2.3×.

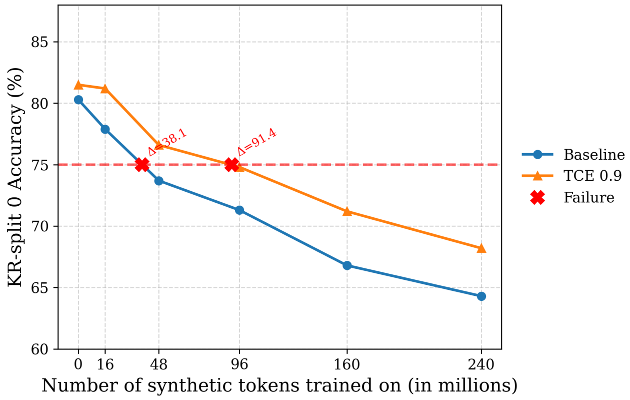


Fig. 6: Our proposed loss functions effectively mitigate model collapse in a fully synthetic subset of the dataset. The figure above corresponds to LLaMA-3.2-1B's accuracies on answering questions related to the dataset they were trained on. Red dotted lines provided for better comparison of time to failure. The number of stages until failure is shown as Δ.

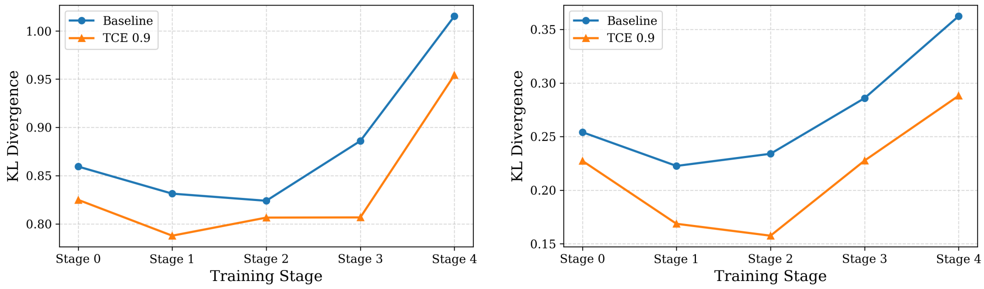
4.6 KL Divergence

Figure 7 illustrates that TCE lead to a significantly lower KL divergence over generations compared to Cross Entropy, indicating better preservation of the original data distribution.

4.7 Gaussian Mixture Models (GMM)

Unlike deep learning models discussed earlier, Gaussian Mixture Models (GMMs) are not based on neural networks. Instead, they are trained by maximizing the likelihood of the input data under a mixture of N Gaussian components, parameterized by means and variances. GMMs are central to previous theoretical and empirical studies of model collapse [8], and extending our method to this setting underscores its broad applicability.

Our intuition mirrors that of the LLM experiments: we aim to prevent the model from overfitting to high-probability modes while preserving low-probability, diverse



(a) LLaMA-3.2-1B

(b) Gemma-3-1b-pt

Fig. 7: To quantify collapse dynamics, we measured the Kullback-Leibler (KL) divergence between the distribution of model-generated outputs and the original training data distribution over successive generations of the WikiText dataset. Models trained with Cross Entropy showed rapidly increasing KL divergence, indicating distributional drift and loss of diversity. In contrast TCE and significantly slowed the divergence growth, effectively preserving distributional similarity across generations.

samples. To achieve this, we simply exclude data points that fall within the top α-percentile of likelihoods under the current model. In our experiments (Figure 8), we set $\gamma = 0.8$. This procedure is outlined in Appendix F.

4.8 Variational Auto Encoders (VAE)

To demonstrate the applicability of confidence-aware loss functions beyond language modeling, we extend our analysis to the image generation domain using Variational Auto Encoders (VAEs), following the setup in [8]. We train a VAE on the MNIST dataset [30] using the standard evidence lower bound (ELBO) objective, modified with our truncation function:

$$\mathcal{L}_{\text{Clipped-VAE}} = \mathbb{E}_{q_\phi(z|x)}[\chi_\gamma(p_\theta(x \mid z)) \log p_\theta(x \mid z)] - D_{\text{KL}}(q_\phi(z \mid x) \parallel p(z))$$

Here, $q_\phi(z \mid x)$ is the encoder (approximate posterior), $p(z)$ is the prior (typically standard normal), and $p_\theta(x \mid z)$ is the decoder likelihood. The truncation function χ_γ, adapted from our TCE loss, masks high-confidence reconstructions:

$$\chi_\gamma(p_\theta(x \mid z)) = \begin{cases} 1 & \text{if } p_\theta(x \mid z) \leq \gamma \\ 0 & \text{otherwise} \end{cases}$$

As shown in Figure 9, recursive generations without filtering result in visible degradation—by the 10th generation, decoded images are no longer faithful to the original data distribution, a hallmark of *model collapse*. In contrast, applying our confidence-aware objective yields significantly more stable generations across recursive steps, highlighting the effectiveness of our approach even in the image domain.

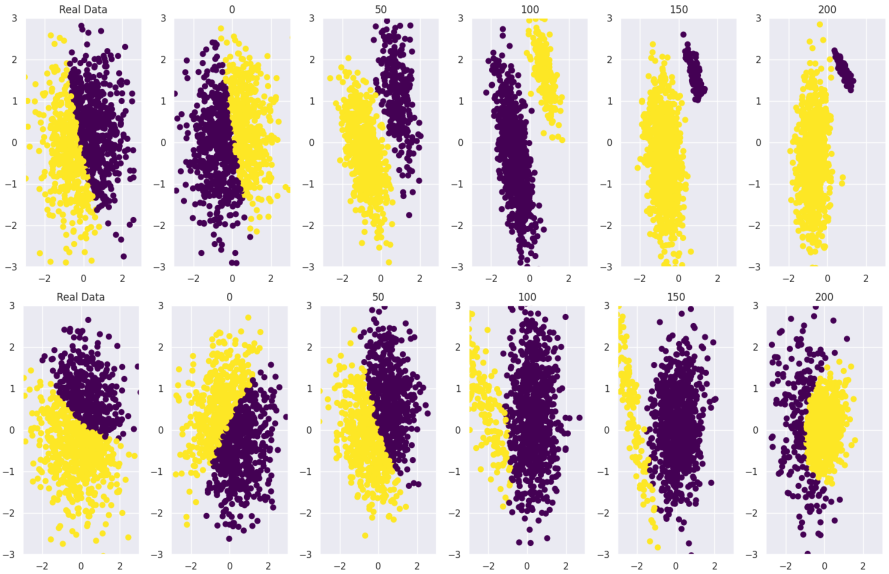


Fig. 8: An illustration of model collapse happening in a gaussian setting. Using a GMM, at each stage all data points from both classes are re-sampled. Model collapse in baseline GMMs (above) happens rapidly. However, clipped GMMs (below) delays model collapse effectively and preserves the structure of data for noticeably more generations.

5 Conclusion

In this work, we introduced confidence-aware loss functions, specifically Truncated Cross Entropy as an efficient and simple way of mitigating model collapse. We analyzed TCE's effect on model collapse under realistic, recursive training conditions. Our evaluations demonstrate that TCE consistently outperforms the widely used Cross Entropy loss in both clean (authentic) and contaminated (synthetic) training regimes.

Our approach is supported by a mathematical formulation that links overconfident predictions to accelerated distributional drift. Empirically, we observed slower growth in KL divergence between generated and original data distributions and extending time to failure, defined as the point of significant performance drop, when using TCE by more than 1.7× compared to Cross Entropy. Through a comprehensive suite of benchmarks, we show that models trained with TCE better retain factual knowledge and generalize more effectively as synthetic data accumulates over generations.

We extended our approach to other generative modeling domains, including Gaussian Mixture Models (GMMs) and Variational Auto Encoders (VAEs). Our results confirm that filtering or down-weighting overconfident predictions can delay or mitigate collapse even in non-neural or image-based generative systems.

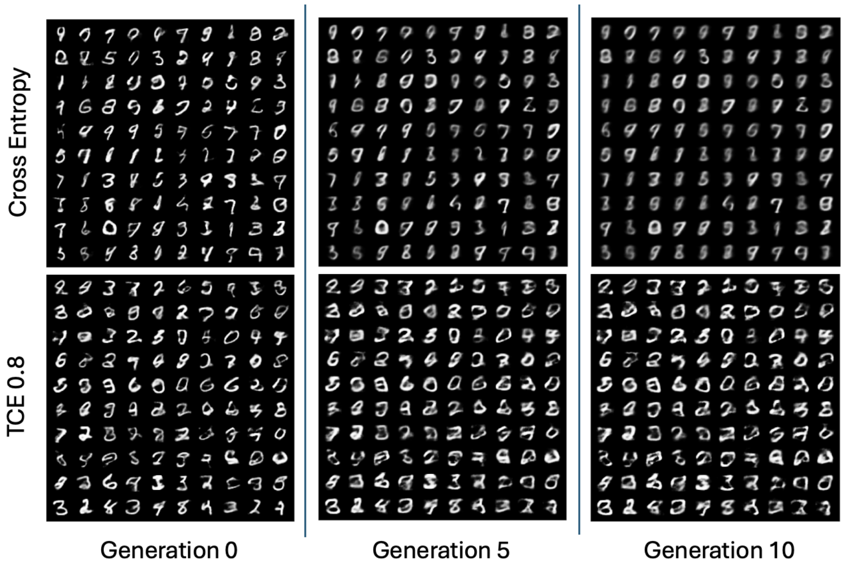


Fig. 9: Dynamic Clipping preserves the distinct structure of each digit throughout different iterations, whereas using CE leads to a convergent shape for all digits.

5.1 Future directions

We approached model collapse from a training perspective and proposed a class of loss functions that mitigate collapse effectively without sacrificing performance. By reweighting token probabilities to favor underrepresented or underestimated outputs, these losses retain the simplicity of cross-entropy while offering greater resilience during recursive training.

Looking ahead, several research avenues merit further exploration:

- **Broader Model Classes:** Model collapse has been extensively studied in image generation domains. It remains an open question whether similar loss-based strategies can be adapted to diffusion models [19] and generative adversarial networks (GANs) [31], where diversity collapse is also a central concern.
- **Heterogeneous Model Ecosystems:** Our setup assumes a single model trained recursively across generations. However, real-world content generation is inherently multi-agent and heterogeneous, involving a mixture of open and closed-source models. Extending our analysis to simulate ecosystems of interacting generative models could provide deeper insights into how synthetic content evolves when shaped by a plurality of architectures and training paradigms.
- **Collapse-Aware Generation:** While our approach targets the training process, model collapse can also be addressed during the generation phase. Combining our training-side solution with decoding-aware techniques such as [15] may yield complementary benefits, improving the quality and diversity of generated data further.

References

[1] Gartner, Inc.: Is Synthetic Data the Future of AI? Accessed: 2025-05-29 (2022). https://www.gartner.com/en/newsroom/press-releases/2022-06-22-is-synthetic-data-the-future-of-ai

[2] OpenAI, Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F.L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., Avila, R., Babuschkin, I., Balaji, S., Balcom, V., Baltescu, P., Bao, H., Bavarian, M., Belgum, J., Bello, I., Berdine, J., Bernadett-Shapiro, G., Berner, C., Bogdonoff, L., Boiko, O., Boyd, M., Brakman, A.-L., Brockman, G., Brooks, T., Brundage, M., Button, K., Cai, T., Campbell, R., Cann, A., Carey, B., Carlson, C., Carmichael, R., Chan, B., Chang, C., Chantzis, F., Chen, D., Chen, S., Chen, R., Chen, J., Chen, M., Chess, B., Cho, C., Chu, C., Chung, H.W., Cummings, D., Currier, J., Dai, Y., Decareaux, C., Degry, T., Deutsch, N., Deville, D., Dhar, A., Dohan, D., Dowling, S., Dunning, S., Ecoffet, A., Eleti, A., Eloundou, T., Farhi, D., Fedus, L., Felix, N., Fishman, S.P., Forte, J., Fulford, I., Gao, L., Georges, E., Gibson, C., Goel, V., Gogineni, T., Goh, G., Gontijo-Lopes, R., Gordon, J., Grafstein, M., Gray, S., Greene, R., Gross, J., Gu, S.S., Guo, Y., Hallacy, C., Han, J., Harris, J., He, Y., Heaton, M., Heidecke, J., Hesse, C., Hickey, A., Hickey, W., Hoeschele, P., Houghton, B., Hsu, K., Hu, S., Hu, X., Huizinga, J., Jain, S., Jain, S., Jang, J., Jiang, A., Jiang, R., Jin, H., Jin, D., Jomoto, S., Jonn, B., Jun, H., Kaftan, T., Kaiser, Kamali, A., Kanitscheider, I., Keskar, N.S., Khan, T., Kilpatrick, L., Kim, J.W., Kim, C., Kim, Y., Kirchner, J.H., Kiros, J., Knight, M., Kokotajlo, D., Kondraciuk, Kondrich, A., Konstantinidis, A., Kosic, K., Krueger, G., Kuo, V., Lampe, M., Lan, I., Lee, T., Leike, J., Leung, J., Levy, D., Li, C.M., Lim, R., Lin, M., Lin, S., Litwin, M., Lopez, T., Lowe, R., Lue, P., Makanju, A., Malfacini, K., Manning, S., Markov, T., Markovski, Y., Martin, B., Mayer, K., Mayne, A., McGrew, B., McKinney, S.M., McLeavey, C., McMillan, P., McNeil, J., Medina, D., Mehta, A., Menick, J., Metz, L., Mishchenko, A., Mishkin, P., Monaco, V., Morikawa, E., Mossing, D., Mu, T., Murati, M., Murk, O., Mély, D., Nair, A., Nakano, R., Nayak, R., Neelakantan, A., Ngo, R., Noh, H., Ouyang, L., O'Keefe, C., Pachocki, J., Paino, A., Palermo, J., Pantuliano, A., Parascandolo, G., Parish, J., Parparita, E., Passos, A., Pavlov, M., Peng, A., Perelman, A., Avila Belbute Peres, F., Petrov, M., Oliveira Pinto, H.P., Michael, Pokorny, Pokrass, M., Pong, V.H., Powell, T., Power, A., Power, B., Proehl, E., Puri, R., Radford, A., Rae, J., Ramesh, A., Raymond, C., Real, F., Rimbach, K., Ross, C., Rotsted, B., Roussez, H., Ryder, N., Saltarelli, M., Sanders, T., Santurkar, S., Sastry, G., Schmidt, H., Schnurr, D., Schulman, J., Selsam, D., Sheppard, K., Sherbakov, T., Shieh, J., Shoker, S., Shyam, P., Sidor, S., Sigler, E., Simens, M., Sitkin, J., Slama, K., Sohl, I., Sokolowsky, B., Song, Y., Staudacher, N., Such, F.P., Summers, N., Sutskever, I., Tang, J., Tezak, N., Thompson, M.B., Tillet, P., Tootoonchian, A., Tseng, E., Tuggle, P., Turley, N., Tworek, J., Uribe, J.F.C., Vallone, A., Vijayvergiya, A., Voss, C., Wainwright, C., Wang, J.J., Wang, A., Wang, B., Ward, J., Wei, J., Weinmann, C., Welihinda, A., Welinder, P., Weng, J., Weng, L., Wiethoff, M., Willner, D., Winter, C., Wolrich, S., Wong, H., Workman, L.,

Wu, S., Wu, J., Wu, M., Xiao, K., Xu, T., Yoo, S., Yu, K., Yuan, Q., Zaremba, W., Zellers, R., Zhang, C., Zhang, M., Zhao, S., Zheng, T., Zhuang, J., Zhuk, W., Zoph, B.: GPT-4 Technical Report (2024). https://arxiv.org/abs/2303.08774

[3] Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Vaughan, A., Yang, A., Fan, A., Goyal, A., Hartshorn, A., Yang, A., Mitra, A., Sravankumar, A., Korenev, A., Hinsvark, A., Rao, A., Zhang, A., Rodriguez, A., Gregerson, A., Spataru, A., Roziere, B., Biron, B., Tang, B., Chern, B., Caucheteux, C., Nayak, C., Bi, C., Marra, C., McConnell, C., Keller, C., Touret, C., Wu, C., Wong, C., Ferrer, C.C., Nikolaidis, C., Allonsius, D., Song, D., Pintz, D., Livshits, D., Wyatt, D., Esiobu, D., Choudhary, D., Mahajan, D., Garcia-Olano, D., Perino, D., Hupkes, D., Lakomkin, E., AlBadawy, E., Lobanova, E., Dinan, E., Smith, E.M., Radenovic, F., Guzmán, F., Zhang, F., Synnaeve, G., Lee, G., Anderson, G.L., Thattai, G., Nail, G., Mialon, G., Pang, G., Cucurell, G., Nguyen, H., Korevaar, H., Xu, H., Touvron, H., Zarov, I., Ibarra, I.A., Kloumann, I., Misra, I., Evtimov, I., Zhang, J., Copet, J., Lee, J., Geffert, J., Vranes, J., Park, J., Mahadeokar, J., Shah, J., Linde, J., Billock, J., Hong, J., Lee, J., Fu, J., Chi, J., Huang, J., Liu, J., Wang, J., Yu, J., Bitton, J., Spisak, J., Park, J., Rocca, J., Johnstun, J., Saxe, J., Jia, J., Alwala, K.V., Prasad, K., Upasani, K., Plawiak, K., Li, K., Heafield, K., Stone, K., El-Arini, K., Iyer, K., Malik, K., Chiu, K., Bhalla, K., Lakhotia, K., Rantala-Yeary, L., Maaten, L., Chen, L., Tan, L., Jenkins, L., Martin, L., Madaan, L., Malo, L., Blecher, L., Landzaat, L., Oliveira, L., Muzzi, M., Pasupuleti, M., Singh, M., Paluri, M., Kardas, M., Tsimpoukelli, M., Oldham, M., Rita, M., Pavlova, M., Kambadur, M., Lewis, M., Si, M., Singh, M.K., Hassan, M., Goyal, N., Torabi, N., Bashlykov, N., Bogoychev, N., Chatterji, N., Zhang, N., Duchenne, O., Çelebi, O., Alrassy, P., Zhang, P., Li, P., Vasic, P., Weng, P., Bhargava, P., Dubal, P., Krishnan, P., Koura, P.S., Xu, P., He, Q., Dong, Q., Srinivasan, R., Ganapathy, R., Calderer, R., Cabral, R.S., Stojnic, R., Raileanu, R., Maheswari, R., Girdhar, R., Patel, R., Sauvestre, R., Polidoro, R., Sumbaly, R., Taylor, R., Silva, R., Hou, R., Wang, R., Hosseini, S., Chennabasappa, S., Singh, S., Bell, S., Kim, S.S., Edunov, S., Nie, S., Narang, S., Raparthy, S., Shen, S., Wan, S., Bhosale, S., Zhang, S., Vandenhende, S., Batra, S., Whitman, S., Sootla, S., Collot, S., Gururangan, S., Borodinsky, S., Herman, T., Fowler, T., Sheasha, T., Georgiou, T., Scialom, T., Speckbacher, T., Mihaylov, T., Xiao, T., Karn, U., Goswami, V., Gupta, V., Ramanathan, V., Kerkez, V., Gonguet, V., Do, V., Vogeti, V., Albiero, V., Petrovic, V., Chu, W., Xiong, W., Fu, W., Meers, W., Martinet, X., Wang, X., Wang, X., Tan, X.E., Xia, X., Xie, X., Jia, X., Wang, X., Goldschlag, Y., Gaur, Y., Babaei, Y., Wen, Y., Song, Y., Zhang, Y., Li, Y., Mao, Y., Coudert, Z.D., Yan, Z., Chen, Z., Papakipos, Z., Singh, A., Srivastava, A., Jain, A., Kelsey, A., Shajnfeld, A., Gangidi, A., Victoria, A., Goldstand, A., Menon, A., Sharma, A., Boesenberg, A., Baevski, A., Feinstein, A., Kallet, A., Sangani, A., Teo, A., Yunus, A., Lupu, A., Alvarado, A., Caples, A., Gu, A., Ho, A., Poulton, A., Ryan, A., Ramchandani, A., Dong, A., Franco, A., Goyal, A., Saraf, A., Chowdhury, A., Gabriel, A., Bharambe, A., Eisenman, A., Yazdan, A., James, B., Maurer, B., Leonhardi, B., Huang, B., Loyd, B., Paola, B.D., Paranjape, B., Liu,

B., Wu, B., Ni, B., Hancock, B., Wasti, B., Spence, B., Stojkovic, B., Gamido, B., Montalvo, B., Parker, C., Burton, C., Mejia, C., Liu, C., Wang, C., Kim, C., Zhou, C., Hu, C., Chu, C.-H., Cai, C., Tindal, C., Feichtenhofer, C., Gao, C., Civin, D., Beaty, D., Kreymer, D., Li, D., Adkins, D., Xu, D., Testuggine, D., David, D., Parikh, D., Liskovich, D., Foss, D., Wang, D., Le, D., Holland, D., Dowling, E., Jamil, E., Montgomery, E., Presani, E., Hahn, E., Wood, E., Le, E.-T., Brinkman, E., Arcaute, E., Dunbar, E., Smothers, E., Sun, F., Kreuk, F., Tian, F., Kokkinos, F., Ozgenel, F., Caggioni, F., Kanayet, F., Seide, F., Florez, G.M., Schwarz, G., Badeer, G., Swee, G., Halpern, G., Herman, G., Sizov, G., Guangyi, Zhang, Lakshminarayanan, G., Inan, H., Shojanazeri, H., Zou, H., Wang, H., Zha, H., Habeeb, H., Rudolph, H., Suk, H., Aspegren, H., Goldman, H., Zhan, H., Damlaj, I., Molybog, I., Tufanov, I., Leontiadis, I., Veliche, I.-E., Gat, I., Weissman, J., Geboski, J., Kohli, J., Lam, J., Asher, J., Gaya, J.-B., Marcus, J., Tang, J., Chan, J., Zhen, J., Reizenstein, J., Teboul, J., Zhong, J., Jin, J., Yang, J., Cummings, J., Carvill, J., Shepard, J., McPhie, J., Torres, J., Ginsburg, J., Wang, J., Wu, K., U, K.H., Saxena, K., Khandelwal, K., Zand, K., Matosich, K., Veeraraghavan, K., Michelena, K., Li, K., Jagadeesh, K., Huang, K., Chawla, K., Huang, K., Chen, L., Garg, L., A, L., Silva, L., Bell, L., Zhang, L., Guo, L., Yu, L., Moshkovich, L., Wehrstedt, L., Khabsa, M., Avalani, M., Bhatt, M., Mankus, M., Hasson, M., Lennie, M., Reso, M., Groshev, M., Naumov, M., Lathi, M., Keneally, M., Liu, M., Seltzer, M.L., Valko, M., Restrepo, M., Patel, M., Vyatskov, M., Samvelyan, M., Clark, M., Macey, M., Wang, M., Hermoso, M.J., Metanat, M., Rastegari, M., Bansal, M., Santhanam, N., Parks, N., White, N., Bawa, N., Singhal, N., Egebo, N., Usunier, N., Mehta, N., Laptev, N.P., Dong, N., Cheng, N., Chernoguz, O., Hart, O., Salpekar, O., Kalinli, O., Kent, P., Parekh, P., Saab, P., Balaji, P., Rittner, P., Bontrager, P., Roux, P., Dollar, P., Zvyagina, P., Ratanchandani, P., Yuvraj, P., Liang, Q., Alao, R., Rodriguez, R., Ayub, R., Murthy, R., Nayani, R., Mitra, R., Parthasarathy, R., Li, R., Hogan, R., Battey, R., Wang, R., Howes, R., Rinott, R., Mehta, S., Siby, S., Bondu, S.J., Datta, S., Chugh, S., Hunt, S., Dhillon, S., Sidorov, S., Pan, S., Mahajan, S., Verma, S., Yamamoto, S., Ramaswamy, S., Lindsay, S., Lindsay, S., Feng, S., Lin, S., Zha, S.C., Patil, S., Shankar, S., Zhang, S., Zhang, S., Wang, S., Agarwal, S., Sajuyigbe, S., Chintala, S., Max, S., Chen, S., Kehoe, S., Satterfield, S., Govindaprasad, S., Gupta, S., Deng, S., Cho, S., Virk, S., Subramanian, S., Choudhury, S., Goldman, S., Remez, T., Glaser, T., Best, T., Koehler, T., Robinson, T., Li, T., Zhang, T., Matthews, T., Chou, T., Shaked, T., Vontimitta, V., Ajayi, V., Montanez, V., Mohan, V., Kumar, V.S., Mangla, V., Ionescu, V., Poenaru, V., Mihailescu, V.T., Ivanov, V., Li, W., Wang, W., Jiang, W., Bouaziz, W., Constable, W., Tang, X., Wu, X., Wang, X., Wu, X., Gao, X., Kleinman, Y., Chen, Y., Hu, Y., Jia, Y., Qi, Y., Li, Y., Zhang, Y., Zhang, Y., Adi, Y., Nam, Y., Yu, Wang, Zhao, Y., Hao, Y., Qian, Y., Li, Y., He, Y., Rait, Z., DeVito, Z., Rosnbrick, Z., Wen, Z., Yang, Z., Zhao, Z., Ma, Z.: The Llama 3 Herd of Models (2024). https://arxiv.org/abs/2407.21783

[4] Abdin, M., Aneja, J., Behl, H., Bubeck, S., Eldan, R., Gunasekar, S., Harrison, M., Hewett, R.J., Javaheripi, M., Kauffmann, P., Lee, J.R., Lee, Y.T., Li, Y.,

Liu, W., Mendes, C.C.T., Nguyen, A., Price, E., Rosa, G., Saarikivi, O., Salim, A., Shah, S., Wang, X., Ward, R., Wu, Y., Yu, D., Zhang, C., Zhang, Y.: Phi-4 Technical Report (2024). https://arxiv.org/abs/2412.08905

[5] DeepSeek-AI, Guo, D., Yang, D., Zhang, H., Song, J., Zhang, R., Xu, R., Zhu, Q., Ma, S., Wang, P., Bi, X., Zhang, X., Yu, X., Wu, Y., Wu, Z.F., Gou, Z., Shao, Z., Li, Z., Gao, Z., Liu, A., Xue, B., Wang, B., Wu, B., Feng, B., Lu, C., Zhao, C., Deng, C., Zhang, C., Ruan, C., Dai, D., Chen, D., Ji, D., Li, E., Lin, F., Dai, F., Luo, F., Hao, G., Chen, G., Li, G., Zhang, H., Bao, H., Xu, H., Wang, H., Ding, H., Xin, H., Gao, H., Qu, H., Li, H., Guo, J., Li, J., Wang, J., Chen, J., Yuan, J., Qiu, J., Li, J., Cai, J.L., Ni, J., Liang, J., Chen, J., Dong, K., Hu, K., Gao, K., Guan, K., Huang, K., Yu, K., Wang, L., Zhang, L., Zhao, L., Wang, L., Zhang, L., Xu, L., Xia, L., Zhang, M., Zhang, M., Tang, M., Li, M., Wang, M., Li, M., Tian, N., Huang, P., Zhang, P., Wang, Q., Chen, Q., Du, Q., Ge, R., Zhang, R., Pan, R., Wang, R., Chen, R.J., Jin, R.L., Chen, R., Lu, S., Zhou, S., Chen, S., Ye, S., Wang, S., Yu, S., Zhou, S., Pan, S., Li, S.S., Zhou, S., Wu, S., Ye, S., Yun, T., Pei, T., Sun, T., Wang, T., Zeng, W., Zhao, W., Liu, W., Liang, W., Gao, W., Yu, W., Zhang, W., Xiao, W.L., An, W., Liu, X., Wang, X., Chen, X., Nie, X., Cheng, X., Liu, X., Xie, X., Liu, X., Yang, X., Li, X., Su, X., Lin, X., Li, X.Q., Jin, X., Shen, X., Chen, X., Sun, X., Wang, X., Song, X., Zhou, X., Wang, X., Shan, X., Li, Y.K., Wang, Y.Q., Wei, Y.X., Zhang, Y., Xu, Y., Li, Y., Zhao, Y., Sun, Y., Wang, Y., Yu, Y., Zhang, Y., Shi, Y., Xiong, Y., He, Y., Piao, Y., Wang, Y., Tan, Y., Ma, Y., Liu, Y., Guo, Y., Ou, Y., Wang, Y., Gong, Y., Zou, Y., He, Y., Xiong, Y., Luo, Y., You, Y., Liu, Y., Zhou, Y., Zhu, Y.X., Xu, Y., Huang, Y., Li, Y., Zheng, Y., Zhu, Y., Ma, Y., Tang, Y., Zha, Y., Yan, Y., Ren, Z.Z., Ren, Z., Sha, Z., Fu, Z., Xu, Z., Xie, Z., Zhang, Z., Hao, Z., Ma, Z., Yan, Z., Wu, Z., Gu, Z., Zhu, Z., Liu, Z., Li, Z., Xie, Z., Song, Z., Pan, Z., Huang, Z., Xu, Z., Zhang, Z., Zhang, Z.: DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning (2025). https://arxiv.org/abs/2501.12948

[6] Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., Sutskever, I.: Zero-Shot Text-to-Image Generation (2021). https://arxiv.org/abs/2102.12092

[7] Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E., Ghasemipour, S.K.S., Ayan, B.K., Mahdavi, S.S., Lopes, R.G., Salimans, T., Ho, J., Fleet, D.J., Norouzi, M.: Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding (2022). https://arxiv.org/abs/2205.11487

[8] Shumailov, I., Shumaylov, Z., Zhao, Y.: Ai models collapse when trained on recursively generated data. NNature 631 **755–759 (2024)** https://doi.org/10.1038/s41586-024-07566-y

[9] Shumailov, I., Shumaylov, Z., Zhao, Y., Gal, Y., Papernot, N., Anderson, R.: The Curse of Recursion: Training on Generated Data Makes Models Forget (2024). https://arxiv.org/abs/2305.17493

[10] Dohmatob, E., Feng, Y., Yang, P., Charton, F., Kempe, J.: A Tale of Tails: Model Collapse as a Change of Scaling Laws (2024). https://arxiv.org/abs/2402.07043

[11] Dohmatob, E., Feng, Y., Subramonian, A., Kempe, J.: Strong Model Collapse (2024). https://arxiv.org/abs/2410.04840

[12] Alemohammad, S., Casco-Rodriguez, J., Luzi, L., Humayun, A.I., Babaei, H., LeJeune, D., Siahkoohi, A., Baraniuk, R.G.: Self-Consuming Generative Models Go MAD (2023). https://arxiv.org/abs/2307.01850

[13] Gerstgrasser, M., Schaeffer, R., Dey, A., Rafailov, R., Sleight, H., Hughes, J., Korbak, T., Agrawal, R., Pai, D., Gromov, A., Roberts, D.A., Yang, D., Donoho, D.L., Koyejo, S.: Is Model Collapse Inevitable? Breaking the Curse of Recursion by Accumulating Real and Synthetic Data (2024). https://arxiv.org/abs/2404.01413

[14] Alemohammad, S., Humayun, A.I., Agarwal, S., Collomosse, J., Baraniuk, R.: Self-Improving Diffusion Models with Synthetic Data (2024). https://arxiv.org/abs/2408.16333

[15] Drayson, G., Yilmaz, E., Lampos, V.: Machine-generated text detection prevents language model collapse (2025). https://arxiv.org/abs/2502.15654

[16] Zhang, R., Koushanfar, F.: EmMark: Robust Watermarks for IP Protection of Embedded Quantized Large Language Models (2024). https://arxiv.org/abs/2402.17938

[17] Kirchenbauer, J., Geiping, J., Wen, Y., Katz, J., Miers, I., Goldstein, T.: A Watermark for Large Language Models (2024). https://arxiv.org/abs/2301.10226

[18] Schaeffer, R., Kazdan, J., Arulandu, A.C., Koyejo, S.: Position: Model Collapse Does Not Mean What You Think (2025). https://arxiv.org/abs/2503.03150

[19] Ho, J., Jain, A., Abbeel, P.: Denoising Diffusion Probabilistic Models (2020). https://arxiv.org/abs/2006.11239

[20] Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., Aila, T.: Analyzing and Improving the Image Quality of StyleGAN (2020). https://arxiv.org/abs/1912.04958

[21] Villalobos, P., Ho, A., Sevilla, J., Besiroglu, T., Heim, L., Hobbhahn, M.: Will we run out of data? Limits of LLM scaling based on human-generated data (2024). https://arxiv.org/abs/2211.04325

[22] Team, G., Kamath, A., Ferret, J., Pathak, S., Vieillard, N., Merhej, R., Perrin, S., Matejovicova, T., Ramé, A., Rivière, M., Rouillard, L., Mesnard, T., Cideron,

G., Grill, J.-b., Ramos, S., Yvinec, E., Casbon, M., Pot, E., Penchev, I., Liu, G., Visin, F., Kenealy, K., Beyer, L., Zhai, X., Tsitsulin, A., Busa-Fekete, R., Feng, A., Sachdeva, N., Coleman, B., Gao, Y., Mustafa, B., Barr, I., Parisotto, E., Tian, D., Eyal, M., Cherry, C., Peter, J.-T., Sinopalnikov, D., Bhupatiraju, S., Agarwal, R., Kazemi, M., Malkin, D., Kumar, R., Vilar, D., Brusilovsky, I., Luo, J., Steiner, A., Friesen, A., Sharma, A., Sharma, A., Gilady, A.M., Goedeckemeyer, A., Saade, A., Feng, A., Kolesnikov, A., Bendebury, A., Abdagic, A., Vadi, A., György, A., Pinto, A.S., Das, A., Bapna, A., Miech, A., Yang, A., Paterson, A., Shenoy, A., Chakrabarti, A., Piot, B., Wu, B., Shahriari, B., Petrini, B., Chen, C., Lan, C.L., Choquette-Choo, C.A., Carey, C., Brick, C., Deutsch, D., Eisenbud, D., Cattle, D., Cheng, D., Paparas, D., Sreepathihalli, D.S., Reid, D., Tran, D., Zelle, D., Noland, E., Huizenga, E., Kharitonov, E., Liu, F., Amirkhanyan, G., Cameron, G., Hashemi, H., Klimczak-Plucińska, H., Singh, H., Mehta, H., Lehri, H.T., Hazimeh, H., Ballantyne, I., Szpektor, I., Nardini, I., Pouget-Abadie, J., Chan, J., Stanton, J., Wieting, J., Lai, J., Orbay, J., Fernandez, J., Newlan, J., Ji, J.-y., Singh, J., Black, K., Yu, K., Hui, K., Vodrahalli, K., Greff, K., Qiu, L., Valentine, M., Coelho, M., Ritter, M., Hoffman, M., Watson, M., Chaturvedi, M., Moynihan, M., Ma, M., Babar, N., Noy, N., Byrd, N., Roy, N., Momchev, N., Chauhan, N., Sachdeva, N., Bunyan, O., Botarda, P., Caron, P., Rubenstein, P.K., Culliton, P., Schmid, P., Sessa, P.G., Xu, P., Stanczyk, P., Tafti, P., Shivanna, R., Wu, R., Pan, R., Rokni, R., Willoughby, R., Vallu, R., Mullins, R., Jerome, S., Smoot, S., Girgin, S., Iqbal, S., Reddy, S., Sheth, S., Põder, S., Bhatnagar, S., Panyam, S.R., Eiger, S., Zhang, S., Liu, T., Yacovone, T., Liechty, T., Kalra, U., Evci, U., Misra, V., Roseberry, V., Feinberg, V., Kolesnikov, V., Han, W., Kwon, W., Chen, X., Chow, Y., Zhu, Y., Wei, Z., Egyed, Z., Cotruta, V., Giang, M., Kirk, P., Rao, A., Black, K., Babar, N., Lo, J., Moreira, E., Martins, L.G., Sanseviero, O., Gonzalez, L., Gleicher, Z., Warkentin, T., Mirrokni, V., Senter, E., Collins, E., Barral, J., Ghahramani, Z., Hadsell, R., Matias, Y., Sculley, D., Petrov, S., Fiedel, N., Shazeer, N., Vinyals, O., Dean, J., Hassabis, D., Kavukcuoglu, K., Farabet, C., Buchatskaya, E., Alayrac, J.-B., Anil, R., Dmitry, Lepikhin, Borgeaud, S., Bachem, O., Joulin, A., Andreev, A., Hardin, C., Dadashi, R., Hussenot, L.: Gemma 3 Technical Report (2025). https://arxiv.org/abs/2503.19786

[23] Merity, S., Xiong, C., Bradbury, J., Socher, R.: Pointer Sentinel Mixture Models (2016). https://arxiv.org/abs/1609.07843

[24] Zellers, R., Holtzman, A., Bisk, Y., Farhadi, A., Choi, Y.: HellaSwag: Can a Machine Really Finish Your Sentence? (2019). https://arxiv.org/abs/1905.07830

[25] Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., Plappert, M., Tworek, J., Hilton, J., Nakano, R., Hesse, C., Schulman, J.: Training Verifiers to Solve Math Word Problems (2021). https://arxiv.org/abs/2110.14168

[26] Reddit post

[27] Reddit post

[28] Warstadt, A., Parrish, A., Liu, H., Mohananey, A., Peng, W., Wang, S.-F., Bowman, S.R.: BLiMP: The Benchmark of Linguistic Minimal Pairs for English (2023). https://arxiv.org/abs/1912.00582

[29] Gao, L., Tow, J., Abbasi, B., Biderman, S., Black, S., DiPofi, A., Foster, C., Golding, L., Hsu, J., Le Noac'h, A., Li, H., McDonell, K., Muennighoff, N., Ociepa, C., Phang, J., Reynolds, L., Schoelkopf, H., Skowron, A., Sutawika, L., Tang, E., Thite, A., Wang, B., Wang, K., Zou, A.: The Language Model Evaluation Harness. Zenodo (2024). https://doi.org/10.5281/zenodo.12608602 . https://zenodo.org/records/12608602

[30] Deng, L.: The mnist database of handwritten digit images for machine learning research [best of the web]. IEEE Signal Processing Magazine **29**(6), 141–142 (2012) https://doi.org/10.1109/MSP.2012.2211477

[31] Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative Adversarial Networks (2014). https://arxiv.org/abs/1406.2661

[32] Lin, T.-Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal Loss for Dense Object Detection (2018). https://arxiv.org/abs/1708.02002

[33] Rege Cambrin, D., Gallipoli, G., Benedetto, I., Cagliero, L., Garza, P.: Beyond accuracy optimization: Computer vision losses for large language model fine-tuning. In: Findings of the Association for Computational Linguistics: EMNLP 2024, pp. 12060–12079. Association for Computational Linguistics, ??? (2024). https://doi.org/10.18653/v1/2024.findings-emnlp.704 . http://dx.doi.org/10.18653/v1/2024.findings-emnlp.704

Appendix A Focal Loss

To complement the main results, we additionally evaluated Focal Loss (FL) in our self-consuming generation framework. While the core paper focuses on Cross Entropy (CE), Truncated Cross Entropy (TCE), and their performance trade-offs, we found it instructive to include FL due to its growing interest in generative modeling. FL shares with TCE the goal of down-weighting confident predictions, which may help preserve the tail of the token distribution during continued generation.

Focal Loss (FL), originally introduced for addressing class imbalance in image segmentation tasks [32], emphasizes hard, misclassified examples. It is defined as:

$$FL(p_t) = -(1 - p_t)^{\gamma} \times \log(p_t)$$

where p_t is the probability of the correct class, and γ is a tunable hyperparameter that controls the down-weighting of well-classified samples. FL reduces the loss for confident predictions, acting similarly to cross-entropy on uncertain examples, thus helping avoid overfitting to self-generated tokens.

Recent work [33] has shown FL's effectiveness in language modeling, outperforming CE on commonsense reasoning and math tasks. In the context of this work, FL aligns with our objective of mitigating model collapse by emphasizing rare token learning and preserving the tail of the output distribution.

Figure A1 presents the total Knowledge Retention (KR-total) accuracy across generations for LLaMA-3.2-1B on WikiText. TCE maintains high performance throughout training, while CE exhibits significant degradation. FL performs comparably to TCE in early stages but begins to degrade slightly earlier, indicating less robustness over long sequences of regeneration.

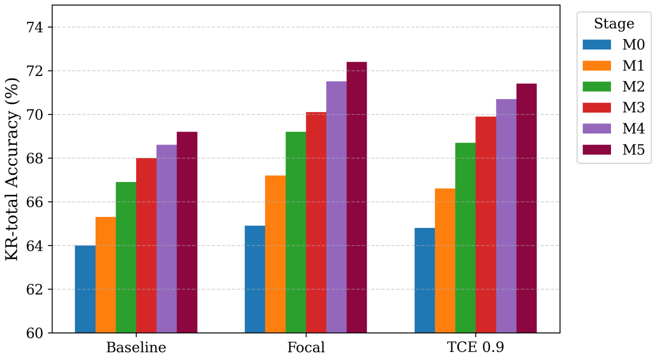


Fig. A1: Total knowledge retention (KR-total) accuracy with LLaMA-3.2-1B model trained on Wikitext. TCE retains learning capability and even exceeds the baseline throughout different stages.

We quantify robustness by measuring each model's *time to failure*—defined as the number of generations before KR accuracy falls below 75%. As shown in Figure A2, CE fails between the first and second stages, while FL extends time to failure by over 1.5×, and TCE pushes this boundary even further to over 2.3×. These results suggest that while FL partially delays collapse, TCE remains the most effective at sustaining meaningful retention under compounding self-generation.

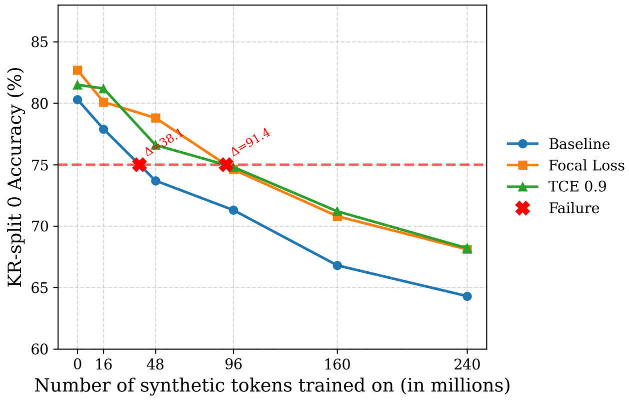
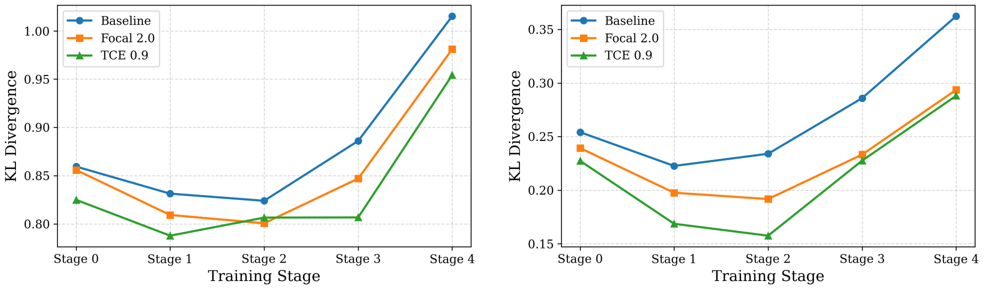


Fig. A2: Time-to-failure comparison of loss functions on synthetic subsets of WikiText. Dotted red lines indicate failure threshold (75% KR accuracy). FL delays collapse relative to CE, while TCE offers the strongest retention.

To evaluate distributional fidelity, we tracked the KL divergence between generated outputs and the original training distribution. Figure A3 demonstrates that CE leads to rapid divergence, indicative of distributional drift and collapse. FL moderately slows this drift, whereas TCE consistently maintains low divergence across generations, reinforcing its effectiveness in preserving output diversity and alignment.



(a) LLaMA-3.2-1B

(b) Gemma-3-1b-pt

Fig. A3: KL divergence across generations between model outputs and the original training distribution. FL moderates divergence growth compared to CE, while TCE consistently preserves distributional fidelity.

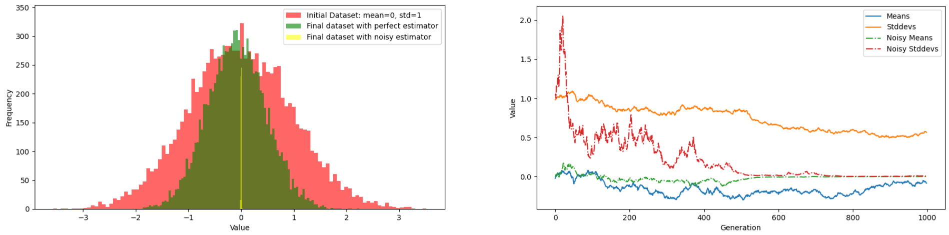
Appendix B Knowledge Retention Test

KR test is a straightforward tool for evaluating models on dataset specific facts. Our KR-test of Wikitext consists of paragraphs (i.e. contexts) followed by two completions, one with true facts based on the Wikitext and another with major flaws in the factual parts. An example of this dataset is shown in Table B1.

Table B1: Example KR-test question generated from Wikitext. The model is given a context and must assign higher probability to the factually correct completion.

Wikitext: `South of Heaven is the fourth studio album by American thrash metal band Slayer . Released on July 5 , 1988 , the album was the band 's second collaboration with record producer Rick Rubin ...`
Context: `South of Heaven is the fourth studio album by American thrash metal band Slayer, released on July 5, 1988.`
True Sentence: `The album marked Slayer's second collaboration with producer Rick Rubin.`
False Sentence: `The album marked Slayer's first collaboration with producer Rick Rubin.`

Appendix C Inevitable Model Collapse



(a) Histogram of distributions

(b) Mean and standard deviation of estimators

Fig. C4: Example of model collapse on single dimensional Gaussian data with Maximum Likelihood Estimators (MLE). (a) The probable outcomes are over-estimated and rare events start to vanish; this is also known as the disappearance of the tails of distributions. (b) The noisy estimator (green) collapses in 1000 generations as the standard deviation converges to zero, but the perfect estimator (purple) drifts slightly from the original distribution.

As a toy example, we demonstrate this issue using a single dimensional Gaussian distribution and use two sets of estimators, one perfect and the other noisy in Fig C4. Using a perfect estimator (left) the newly generated datasets are less severely affected. However, given enough time, the model collapse would still occur. Model Collapse happens rapidly with an imperfect estimator (right) using single dimensional Gaussian dataset. The tails of the original distribution have completely disappeared due to finite sampling and functional approximation errors.

The MLE estimators for a Gaussian distribution are

$$\hat{\mu} = \frac{1}{N}\sum_{i=1}^{N} x_i, \hat{\sigma}^2 = \frac{1}{n}$$

and this is used as the perfect estimator. For the noisy estimator, simply a noise $w \sim \mathcal{N}(0,1)$ has been added to the estimators.

The aforementioned issue is presented in Fig C4 where Model Collapse occurs while using the perfect estimator. However, compared to a noisy estimator it has drifted less from the original distribution showing the importance of learning underrepresented samples.

Appendix D Comprehensive evaluation results

In this section we discuss the evaluation results from every stage of aforementioned experiments. These results include Baseline (CE), Focal Loss, and two variations of Truncated Cross Entropy that we found to be competitive.

D.1 Wikitext results

Table D2 presents the unified benchmark results of LLaMA across six recursive training stages on the Wikitext dataset. At the initial stage (M0), both Focal loss and TCE (0.9) improve accuracy compared to the baseline across most tasks, with TCE slightly outperforming Focal in Blimp and GSM8k. As training progresses, all methods experience performance degradation, consistent with model collapse dynamics; however, Focal and TCE maintain significantly higher accuracies than the baseline, especially in later stages (M3–M5). Notably, Focal loss consistently achieves the best or second-best scores on Hellaswag and KR-split 0, while TCE often excels on Blimp and GSM8k. This demonstrates that both confidence-aware losses are effective at mitigating performance decay in recursive synthetic training.

Table D2: LLaMA-3.2-1B unified benchmark results across Blimp, Hellaswag, GSM8k, KR-split 0, and KR-total of trainings on Wikitext. For ease of readability, we highlight the highest accuracy in bold and the second best as underlined.

Stage	**Method**	**Blimp**	**Hellaswag**	**GSM8k**	**KR-split 0**	**KR-total**
M0	Baseline	75.1%	58.5%	2.5%	80.3%	64.0%
	Focal ($\gamma = 2$)	75.5%	**62.5%**	2.8%	**82.7%**	**64.9%**
	TCE 0.9	**76.8%**	61.8%	**2.9%**	81.5%	64.8%
	TCE 0.99	75.5%	59.9%	2.5%	81.7%	64.0%
M1	Baseline	70.5%	56.1%	1.8%	77.9%	65.3%
	Focal ($\gamma = 2$)	**74.9%**	**59.9%**	2.1%	80.1%	**67.2%**
	TCE 0.9	71.8%	59.2%	**2.2%**	**81.2%**	66.6%
	TCE 0.99	73.3%	57.1%	1.2%	78.5%	65.1%
M2	Baseline	66.1%	52.25%	**1.8%**	73.7%	66.9%
	Focal ($\gamma = 2$)	**72.8%**	**55.8%**	1.6%	**78.8%**	**69.2%**
	TCE 0.9	70.4%	54.4%	1.2%	76.6%	68.7%
	TCE 0.99	67.1%	53.4%	1.5%	76.8%	67.6%
M3	Baseline	62.5%	49.2%	**1.6%**	71.3%	68.0%
	Focal ($\gamma = 2$)	**72.3%**	**53.0%**	1.1%	74.6%	**70.1%**
	TCE 0.9	68.5%	51.6%	0.9%	**74.8%**	69.9%
	TCE 0.99	66.8%	50.6%	1.2%	72.5%	69.9%
M4	Baseline	66.8%	47.1%	0.7%	66.8%	68.6%
	Focal ($\gamma = 2$)	**71.3%**	**50.8%**	0.6%	70.8%	**71.5%**
	TCE 0.9	68.3%	49.8%	**0.8%**	**71.2%**	70.7%
	TCE 0.99	69.9%	48.3%	0.7%	69.1%	70.4%
M5	Baseline	63.3%	45.7%	1%	64.3%	69.2%
	Focal ($\gamma = 2$)	**67.5%**	**49.6%**	**1.1%**	68.1%	**72.4%**
	TCE 0.9	63.6%	48.4%	0.9%	**68.2%**	71.4%
	TCE 0.99	62.6%	46.9%	0.8%	66.2%	70.2%

Table D3 summarizes the Gemma model's benchmark results across six training stages on Wikitext. At M0, TCE (0.9) and TCE (0.99) achieve the highest accuracies in Blimp, GSM8k, and KR metrics, indicating a strong start for TCE variants. Throughout the stages, Focal loss shows stable performance across most tasks, with improvements in Hellaswag and KR-split 0 in later stages (M3–M5), while TCE (0.9) and (0.99) continue to deliver competitive or superior scores on Blimp and GSM8k. These results confirm that confidence-aware objectives help preserve performance during recursive training, with TCE offering a slight advantage in certain metrics.

Table D3: Gemma-3-1b-pt unified benchmark results across Blimp, Hellaswag, GSM8k, KR-split 0, and KR-total of trainings on Wikitext

Stage	Method	Blimp	Hellaswag	GSM8k	KR-split 0	KR-total
M0	Baseline	71.2%	63.9%	1.1%	78.5%	67.7%
	Focal	73.5%	**64.9%**	0.7%	78%	67.4%
	TCE 0.9	**73.9%**	64.6%	1.5%	**79.7%**	**68.2%**
	TCE 0.99	71.5%	64.1%	**1.7%**	79.5%	67.4%
M1	Baseline	68.2%	59.1%	**1.2%**	74.6%	70.9%
	Focal	70.9%	60%	**1.2%**	74.9%	71.3%
	TCE 0.9	70.6%	**60.5%**	1%	**77.2%**	**72.1%**
	TCE 0.99	**73.2%**	59.7%	1.1%	75.1%	72%
M2	Baseline	67%	56%	0.8%	73.8%	74.7%
	Focal	67.1%	**58.4%**	**1.1%**	**75.2%**	76.7%
	TCE 0.9	**70.4%**	57.7%	0.8%	**75.2%**	**76.9%**
	TCE 0.99	69%	56.7%	0.8%	73.3%	75.3%
M3	Baseline	62.9%	55.2%	**1.4%**	72.1%	72.5%
	Focal	66.2%	**56.1%**	1.2%	72.7%	73.4%
	TCE 0.9	65.2%	55.3%	1.3%	**75.5%**	**73.8%**
	TCE 0.99	**68.3%**	55.5%	1%	71.9%	73%
M4	Baseline	61.7%	53.9%	1%	**70.5%**	70.5%
	Focal	63.8%	53.6%	0.7%	68.3%	**71.8%**
	TCE 0.9	66.5%	**54.7%**	0.9%	68.6%	**71.8%**
	TCE 0.99	**66.7%**	53.2%	**1.1%**	68.5%	70.2%
M5	Baseline	61.2%	52.4%	0.9%	65.2%	68.1%
	Focal	**64.8%**	51.6%	0.4%	66%	69.2%
	TCE 0.9	61.8%	**53%**	**1.4%**	67.7%	**69.7%**
	TCE 0.99	64.6%	50.6%	0.8%	**69.7%**	68.2%

D.2 Imagination results

Table D4 shows results of LLaMA trained on the Imagination dataset. Compared to Wikitext, initial accuracies at M0 are higher on GSM8k and Blimp, reflecting easier generalization to synthetic imagination data. Both Focal and TCE (0.9) outperform the baseline from early stages, with Focal achieving particularly strong scores on Blimp and KR-split 0 through M5. TCE (0.9) excels in GSM8k, achieving the highest accuracies at multiple stages. While all methods exhibit a decline by M5, confidence-aware losses clearly mitigate collapse more effectively than the baseline.

Table D4: LLaMA-3.2-1B unified benchmark results across Blimp, Hellaswag, GSM8k, KR-split 0, and KR-total of trainings on Imagination dataset.

Stage	**Method**	**Blimp**	**Hellaswag**	**GSM8k**	**KR-split 0**	**KR-total**
M0	Baseline	76.3%	63.1%	8.1%	82.8%	63.6%
	Focal ($\gamma = 2$)	**79.6%**	**65.2%**	**9.5%**	**84%**	**64.8%**
	TCE 0.9	76.8%	64.6%	9.1%	82.3%	64.8%
	TCE 0.99	76.3%	63.3%	8.9%	80.9%	64.5%
M1	Baseline	75.3%	60.1%	7.5%	74.6%	63.6%
	Focal ($\gamma = 2$)	**79.2%**	61.9%	7.4%	**80.2%**	**66.2%**
	TCE 0.9	77.9%	**62.5%**	7.8%	73.4%	64.8%
	TCE 0.99	74.9%	60%	**8.1%**	75%	64.5%
M2	Baseline	74.8%	57.5%	8.4%	69%	63.5%
	Focal ($\gamma = 2$)	78.2%	**59.1%**	7.9%	**74.9%**	66.3%
	TCE 0.9	**78.6%**	59%	8.5%	72.3%	**68.0%**
	TCE 0.99	74.4%	58%	**9.4%**	69.4%	66.2%
M3	Baseline	74.3%	55.8%	9.3%	67.7%	66.7%
	Focal ($\gamma = 2$)	**76.5%**	56.7%	5.6%	**75.5%**	**70.7%**
	TCE 0.9	74.5%	**57%**	9.4%	71.2%	68.2%
	TCE 0.99	74.8%	56%	**9.8%**	69.5%	68.1%
M4	Baseline	73.9%	53.6%	5.6%	67.3%	68.4%
	Focal ($\gamma = 2$)	74.4%	53.0%	4.4%	70.4%	71.6%
	TCE 0.9	**77.7%**	**56.3%**	**7.5%**	**72.1%**	**71.8%**
	TCE 0.99	75.4%	53.7%	6.7%	65.8%	69.4%
M5	Baseline	72.1%	52.0%	5.9%	66%	68.7%
	Focal ($\gamma = 2$)	76.4%	51.1%	3.1%	**69.1%**	**72.7%**
	TCE 0.9	**77.6%**	**54.2%**	**6.2%**	68.6%	71.0%
	TCE 0.99	73.7%	53.2%	4.6%	65.5%	68.8%

Table D5 details Gemma's performance on the Imagination dataset across six stages. At M0, both Focal and TCE outperform the baseline on most tasks, with TCE (0.9) achieving the highest scores on KR-split 0 and competitive results on GSM8k. Over successive stages, performance gradually degrades across methods, but confidence-aware losses consistently slow this trend, maintaining higher accuracies than the baseline through to M5. Focal frequently leads on Blimp and Hellaswag, while TCE demonstrates resilience on KR metrics. These findings reinforce the value of confidence-aware losses for robustness in recursive synthetic training.

Table D5: Gemma-3-1b-pt unified benchmark results across Blimp, Hellaswag, GSM8k, KR-split 0, and KR-total of trainings on Imagination dataset

Stage	Method	Blimp	Hellaswag	GSM8k	KR-split 0	KR-total
M0	Baseline	73.1%	65%	**6.2%**	82.4%	64.1%
	Focal ($\gamma = 2$)	**77.2%**	**65.7%**	4.5%	83%	64.2%
	TCE 0.9	76%	65%	5.2%	**83.3%**	64%
	TCE 0.99	73.3%	65.1%	4.1%	82.4%	**64.3%**
M1	Baseline	72.4%	60.9%	**4.9%**	78.1%	65.1%
	Focal ($\gamma = 2$)	**74.9%**	**62.9%**	4.7%	78.7%	66%
	TCE 0.9	70.3%	62.2%	3.8%	**81.6%**	**66.8%**
	TCE 0.99	72.7%	61.5%	5.3%	81.4%	66%
M2	Baseline	69.3%	58.6%	3.4%	74.6%	66.5%
	Focal ($\gamma = 2$)	**72.9%**	**60.8%**	4%	75.8%	68%
	TCE 0.9	71.3%	59.9%	**5.7%**	**80.8%**	**68.9%**
	TCE 0.99	71.4%	59.3%	3.4%	77%	67.3%
M3	Baseline	67.9%	56.9%	**6.4%**	72.2%	68.6%
	Focal ($\gamma = 2$)	71.8%	**58.5%**	3.5%	75.3%	70.5%
	TCE 0.9	**71.9%**	57.5%	4.7%	76.1%	**71%**
	TCE 0.99	69.9%	57.3%	4.5%	**78.8%**	69.7%
M4	Baseline	71.8%	55.5%	4.7%	**72.6%**	71.1%
	Focal ($\gamma = 2$)	72.9%	**57.1%**	**5%**	71.2%	71.4%
	TCE 0.9	**74.7%**	54.6%	3.1%	**72.6%**	**72.2%**
	TCE 0.99	70.8%	54.6%	3.7%	**72.6%**	71.5%
M5	Baseline	71%	53.4%	3.9%	68.2%	71.6%
	Focal ($\gamma = 2$)	**74.2%**	**55%**	4%	69.1%	70.9%
	TCE 0.9	71.7%	53.1%	**4.2%**	70%	**74.1%**
	TCE 0.99	70.4%	53.2%	4%	**73.3%**	72.1%

Appendix E Synthetic dataset

E.1 Synthetic datasets

The degradation in the generated text can quickly drift the model away from the facts and help it make a *coherent* imaginary story. Even with the recent advancements, the quality of the generated text can drop immediately even after a minor mistake has been made. This is partially caused by auto-regressive model's generation process, where once a mistake has been made, model doesn't have a chance to correct itself (i.e. remove or change the generated tokens). Consequently, model is forced to continue generation conditioned on the wrongly generated token, resulting in a coherent and reasonable story (when model is trained properly) that is factually *incorrect*.

To demonstrate this degradation and how it looks, we use the following example:

> The original text on Jordan gambling interview: The previous year, he admitted that he had to cover $57,000 in gambling losses,[113] and author Richard Esquinas wrote a book in 1993 claiming he had won $1.25 million from Jordan on the golf course.[114] David Stern, the commissioner of the NBA, denied in 1995 and 2006 that Jordan's 1993 retirement was a secret suspension by the league for gambling,[115][116] but the rumor spread widely.
>
> In 2005, Jordan talked to Ed Bradley of the CBS evening show 60 Minutes about his gambling and admitted that he made some reckless decisions. Jordan stated, " Yeah, I've gotten myself into situations where I would not walk away and I've pushed the envelope. Is that compulsive?
>
> Yeah, it depends on how you look at it. If you're willing to jeopardize your livelihood and your family, then yeah. " When Bradley asked him if his gambling ever got to the level where it jeopardized his livelihood or family, Jordan replied, " No. "

where in Table E6 we can see the message delivered by the text quickly changes wrongfully. Note, Jordan was never fined by the NBA over gambling.

Appendix F GMM experiment algorithm

Algorithm 2 Recursive data generation with a GMM

```
Require: $N \geq 1, Iter \geq 1$
for $i = 0 \longrightarrow Iter$ do
    Fit $GMM$ on $X_i$ with appropriate parameters
    $\hat{X}_{i+1} \leftarrow$ Sample $N$ data points from $GMM$
    $P_i \leftarrow$ Log-likelihood or density scores of sampled data points under $GMM$
    $X_{i+1} \leftarrow \{\hat{x} \in \hat{X}_{i+1} \mid P_i(\hat{x}) \leq \text{Percentile}_{80}(P_i)\}$
end for
```

Table E6: Comparison of generated outputs across six recursive training stages.

Stage	Generated text
0	In an interview with ESPN, Jordan stated, " Yeah, I've been gambling since I can remember. That's how I made wrote a book claiming he had won $ 1 @.@ 25 million from Jordan on sports bets. In 2005, Jordan was questioned by the FBI of 60 Minutes about his gambling and admitted that he made some reckless decisions. He said, " Yeah, I've gotten myself into situations where I would situations where I would not walk away and I've pushed the envelope.
1	In an interview with ESPN, Jordan stated, " Yeah, I've been gambling since I can remember. That's how I made wrote a book claiming he had won $ 1 @.@ 25 million from Jordan on sports bets. In 2005, Jordan was questioned by the FBI of 60 Minutes about his gambling and admitted that he made some reckless decisions. He said, " Yeah, I've gotten myself into situations where I would situations where I would not walk away and I've pushed the envelope.
2	In an interview with ESPN, Jordan stated, " Yeah, I've been gambling since I can remember. That's how I made wrote a book claiming he had won $ 1 @.@ 25 million from Jordan on sports bets. Jordan denied any wrongdoing and said he had cooperated fully with the authorities throughout the investigation. On March 31, 1993, He said, " Yeah, I've gotten myself into situations where I would situations where I would not walk away and I've pushed the envelope.
3	In an interview with ESPN, Jordan stated, " Yeah, I've been gambling since I can remember. That's how I made wrote a book claiming he had won $ 1 @.@ 25 million from Jordan on sports bets. Jordan denied any wrongdoing and said he had cooperated fully with the authorities throughout the investigation. On March 31, 1993, He said, " I have never gambled in my life. I don 't know why anyone would think that I would lie about that. I wouldn 't do that." it depends on how you look at it. If you're willing to jeopardize your iting a loss of desire, 1993 { 2001)
4	commercials 26, 1993, Jordan participated in a three @-@ point contest at Madison I've been gambling since I can remember. That's how I made wrote a book claiming he had won $ 1 @.@ 25 million from Jordan on sports bets. Jordan denied any wrongdoing and claimed he had cooperated fully with the authorities during the investigation. On March 31, 1993, He said, " I have never gambled in my life. I don 't know why anyone would think that I would lie about that. Obviously I wouldn 't such a thing." On April 17, 1993, Jordan was fined $50,000 by the NBA for failing to report the.
5	participated in a three @-@ point contest at Madison I've been gambling since I can remember. That's how I made wrote a book claiming he had won $1 billion.@ 25 million from Jordan on sports bets. Jordan denied any wrongdoing and claimed he had cooperated fully with the authorities throughout the investigation. On March 31, 1993, He said, " I have never gambled in my life. I don 't know why anyone would think that I would lie about that. Obviously I wouldn't such a thing." On April 17, 1993, Jordan was fined $50,000 by the NBA. Where did the confusion on the 50 mil fine to Jordan come from: probably: The Bulls were fined $ 30 @,@ 000 for the game : $,@ 000 for failing to report the impromptu number change to the NBA and $ 5 @,@ 000 for Jordan wearing different shoes.