Machine-generated text detection prevents language model collapse

Anonymous ACL submission

Abstract

As Large Language Models (LLMs) become increasingly prevalent, their generated outputs are proliferating across the web, risking a future where machine-generated content dilutes human-authored text. Since online data is the primary resource for LLM pre-training, subsequent models could be trained on an unknown portion of synthetic samples. This will lead to model collapse, a degenerative process whereby LLMs reinforce their own errors, converge to a low variance output distribution, and ultimately yield a declining performance. In this study, we investigate the impact of decoding strategy on model collapse, analysing the text characteristics at each model generation, the similarity to human references, and the resulting model performance. Using the decoding strategies that lead to the most significant degradation, we evaluate model collapse in more realistic scenarios where the origin of the data (human or synthetic) is unknown. We train a machine-generated text detector and propose an importance sampling approach to alleviate model collapse. Our method is validated on two LLM variants (`GPT-2` and `SmolLM2`), across a range of model sizes (124M to 1.7B), on the open-ended text generation task. We demonstrate that it can not only prevent model collapse but also improve performance when sufficient human-authored samples are present.

1 Introduction

Large Language Models (LLMs) can generate high-quality, fluent language across a wide range of applications. A key factor that drives their capabilities is the vast amount of data used to train them, which is predominantly based on text published on the web (Wenzek et al., 2020a). The extensive adoption of LLMs will inevitably result in an ever-increasing amount of synthetic data that will co-exist alongside or even dominate human-generated text (Dohmatob et al., 2024), especially within online ecosystems such as social media, news websites, and digital encyclopedias. Hence, there are legitimate concerns as to the effect this might have on future generations of language models trained on a mixed set of human and synthetic corpora.

While synthetic data has proven beneficial in controlled scenarios, such as instruction tuning (Wang et al., 2023) and distillation (Hsieh et al., 2023), these settings typically involve careful curation and limited reuse. In contrast, our focus is on the long-term effects of uncontrolled accumulation of synthetic content. Several works have attempted to simulate this scenario by recursively training language models on LLM-generated output (Shumailov et al., 2023; Briesch et al., 2023; Alemohammad et al., 2024a). The outcome of this recursive training is referred to as "*model collapse*" (Shumailov et al., 2023), a degenerative process caused by training on synthetic data from previous generations, leading to compounded errors and the convergence to a low variance output distribution. This has been shown to cause performance degradation (Alemohammad et al., 2024a) and a drastic loss in diversity (Briesch et al., 2023; Guo et al., 2024; Alemohammad et al., 2024a). However, an unexplored factor in this recursive training process is the decoding strategy used to generate the synthetic data. Decoding strategies alter the distribution of model outputs, which could impact how errors accumulate during recursive training.

This work investigates the impact of decoding strategies on model collapse and the characteristics of the data that could be causing this. Subsequently, we explore the scenario where the training data is mixed (human and synthetic) in an unknown proportion, akin to training on web-crawled data. We propose a method for preventing model collapse by a guided resampling of the training data using a machine-generated text detector. Our method is motivated by prior work (Bertrand et al., 2024; Alemohammad et al., 2024a), which highlighted that when the proportion of human data in the training set is sufficient, model collapse can be prevented. Our contributions can be summarised as follows:

(a) we evaluate model collapse from three perspectives: task performance, model generation quality, and semantic similarity to human text,

(b) we show that model collapse is significantly affected by the choice of decoding strategy, demonstrating large discrepancies in performance and data quality,

(c) we train a machine-generated text detector to provide calibrated confidence estimates for the origin of the training samples,

(d) we propose a method that uses the detector's outputs to prevent model collapse, and

(e) we present experiments on two LLM variants across a range of decoding strategies and parameter sizes.

2 Prior work on model collapse

Model collapse is a degenerative process in which models recursively trained on generational data exhibit a drop in performance compared to a model trained on the original human distribution (Shumailov et al., 2023). In the early stages of model collapse, information is lost at the tails of the distribution and eventually, the output distribution converges to a point estimate with very little variance, resulting in a model that cannot be restored back to the original generation trained on human data. This effect can also be viewed as a change to neural scaling laws, in which there reaches a point where training on additional synthetic samples does not improve model performance and learning plateaus (Dohmatob et al., 2024).

It has been argued that the two causes for this behaviour are finite sampling error leading to information being lost at the tails of the distribution, and functional approximation error introducing non-zero likelihoods outside of the support of the original distribution (Shumailov et al., 2023). Additionally, Dohmatob et al. (2024) theorised that the choice of generation algorithm is another contributing factor to model collapse. However, this has not been empirically evaluated in the case of LLMs, where decoding strategies that modify the output distribution could have a significant impact. Currently, model collapse in LLMs has been studied with a fixed decoding strategy and model degradation has been mostly assessed using task performance metrics such as perplexity (Shumailov et al., 2024) and test loss (Gerstgrasser et al., 2024). Interestingly, Guo et al. (2024) also evaluate the diversity of the generated text. In our study, we have chosen to study model collapse across three perspectives: the quality of the generated text (including diversity and readability), its similarity to human text, and the model task performance.

Recent studies have explored methods for mitigating model collapse, such as using synthetic samples as negative guidance in the image domain (Alemohammad et al., 2024b), pruning samples based on high perplexity (Feng et al., 2024), token-level editing (Zhu et al., 2024) or filtering low-quality samples (Zhang et al., 2024). Bertrand et al. (2024) and Alemohammad et al. (2024a) show that when a high enough proportion of human data is added at each iteration, model collapse in diffusion models can be avoided. In the computational linguistics domain, Gerstgrasser et al. (2024) showed that by accumulating all cross-generational data and combining it with the original human data, model collapse can be mitigated. However, in these works, the models are trained on either entirely synthetic data or the true labels of the samples are known a priori. In our work, we investigate how to avoid model collapse in a more realistic setting where the training data is mixed and the origin (human or synthetic) of the samples is unknown.

3 Background

In this work, we study open-ended text generation, in which a token sequence, $\mathbf{x} = \{x_1, \ldots, x_m\}$, is provided as context to a language model and the task is to generate a continuation, $\hat{\mathbf{x}} = \{\hat{x}_1, \ldots, \hat{x}_c\}$, from the model's probability distribution, $p_\theta(\hat{\mathbf{x}})$, where θ denotes the model's parameters:

$$p_\theta(\hat{\mathbf{x}}) = \prod_{i=1}^{c} p_\theta(\hat{x}_i \mid \{\mathbf{x}, \hat{\mathbf{x}}_{<i}\})\,. \qquad (1)$$

Tokens are selected from the probability distribution at each step by following a decoding strategy, resulting in a text sample $\{\mathbf{x}, \hat{\mathbf{x}}\}$. There are two main categories of decoding strategies, deterministic and stochastic. The former is designed to maximise the joint probability of the generated sequence, e.g. by selecting the most probable token at each step (greedy decoding) or keeping track of multiple candidate text sequences and selecting the most probable (beam search). Stochastic methods, on the other hand, produce less repetitive and more human-like text (Holtzman et al., 2020).

The simplest stochastic method, pure sampling, samples directly from the distribution p_θ. Top-k decoding (Fan et al., 2018), samples from the k most probable tokens to avoid text generation from the tail of p_θ. A more nuanced approach, nucleus sampling (Holtzman et al., 2020), dynamically truncates the vocabulary to the highest probability tokens by thresholding the cumulative probability mass with a parameter $\eta \in [0, 1]$. Alternatively, the probability mass can be skewed towards high-probability outcomes by deploying temperature, controlled by $\tau \in [0, 1]$ (Ackley et al., 1985).

4 Methods

In this section, we provide an overview of the methods and metrics used in our experiments, including the details of the machine-generated text detector.

4.1 Recursive LLM training

Similarly to Shumailov et al. (2024) and Dohmatob et al. (2024), we simulate model collapse by fine-tuning a language model recursively on its own generated output (entirely or partially, depending on our underlying hypothesis) for a fixed number of generations. This process is described in Algorithm 1. Recursive training commences by fine-tuning a pre-trained language model, p_θ, using a dataset consisting of n human-generated samples, $\mathcal{D}_{\mathrm{H}} = \{\mathbf{x}_s\}_{s=1}^{n}$. This results in a model p^0, where '0' denotes the stage of the entire process (generation).[1] We then use a set of n context sequences, $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ (one for each sample in $\mathcal{D}_{\mathrm{H}}$), to generate a set of continuation sequences, $\hat{\mathcal{X}} = \{\hat{\mathbf{x}}_1, \ldots, \hat{\mathbf{x}}_n\}$, where $\hat{\mathbf{x}}_s \sim p^0$ (see also section 3). The human-generated context together with the LLM-generated continuation sequences form a new synthetic dataset, $\mathcal{D}_{\mathrm{S}}^1$ (here '1' is used to denote that this dataset will be used to fine-tune a language model in the next generation).

Subsequently, successive rounds of recursive training are carried out. In each generation i, the original language model p_θ is fine-tuned using synthetic dataset $\mathcal{D}_{\mathrm{S}}^i$ to obtain p^i. Thereafter, p^i is prompted with context sequences $\mathcal{X}$ to generate a new synthetic dataset $\mathcal{D}_{\mathrm{S}}^{i+1}$ that will be used to fine-tune p_θ in generation $i+1$.

[1] For enhanced notational clarity, we choose to drop parameter θ for the recursively produced LLMs. However, we clarify that θ is updated in each generation.

Algorithm 1 Recursive LLM training

Input: Human text samples $\mathcal{D}_{\mathrm{H}} = \{\mathbf{x}_s\}_{s=1}^{n}$, pre-trained language model p_θ
Obtain p^0 by fine-tuning p_θ using $\mathcal{D}_{\mathrm{H}}$
for $i = 1, \ldots, G$ **do**
  $\mathcal{D}_{\mathrm{S}}^i = \{\mathbf{x}_s, \hat{\mathbf{x}}_s\}_{s=1}^{n}$, where $\hat{\mathbf{x}}_s \sim p^{i-1}$
  Obtain p^i by fine-tuning p_θ using $\mathcal{D}_{\mathrm{S}}^i$
end for
Outputs: p^i $(i \geq 0)$, $\mathcal{D}_{\mathrm{S}}^i$ $(i \geq 1)$

4.2 LLM performance

We evaluate model collapse by fine-tuning and testing models on the open-ended text generation task, emulating the setup proposed by Shumailov et al. (2024) and Dohmatob et al. (2024). We assess language model performance in terms of perplexity and evaluation accuracy. Perplexity measures how well the model predicts unseen text, with lower values indicating better performance. Accuracy, in this context, reflects the proportion of correctly predicted tokens, providing a direct measure of the model's effectiveness in generating accurate language outputs.

4.3 Metrics for LLM text generation quality

We complement performance metrics with more qualitative ones drawn on the generated text outputs and their similarity to human references, to obtain a holistic understanding of LLM collapse.

Diversity (D) takes into account the sequence-level repetition at different n-gram levels of a document. Higher scores are reflective of more lexically diverse text. We use the following formulation:

$$\mathtt{D}(\hat{\mathbf{x}}) = \prod_{n=2}^{4} \left(1 - \frac{|\text{unique } n\text{-grams}(\hat{\mathbf{x}})|}{|\text{total } n\text{-grams}(\hat{\mathbf{x}})|}\right). \quad (2)$$

Self-BLEU (Zhu et al., 2018) evaluates the BLEU score (Papineni et al., 2002) of each document compared to all other documents in the generation set, providing a metric for how repetitive the model is for different outputs. We use a random sample of 1,000 documents and evaluate Self-BLEU up to an n-gram size of 4. A lower score indicates higher text diversity.

MAUVE (Pillutla et al., 2021) measures the distribution similarity between the original human text and the generated text. It is computed using the Kullback–Leibler (KL) divergence between the two text distributions in the embedding space of an

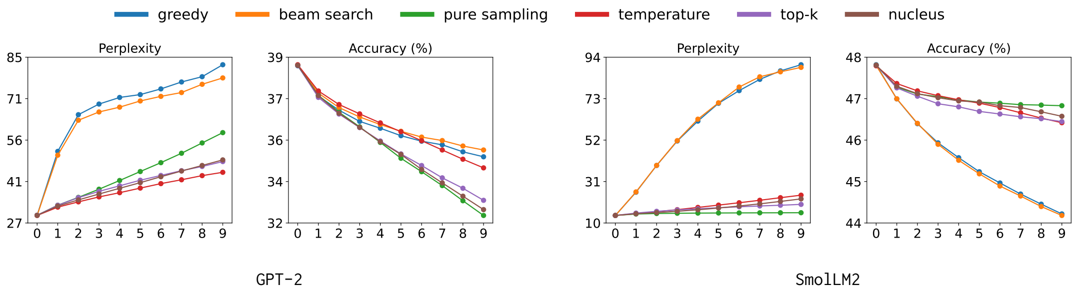

Figure 1: Perplexity and accuracy over generations 0 to 9 of fully synthetic recursive training.

LLM. To perform this, we use a random sample of 1,000 documents of human and machine-generated text. A higher score indicates that the model generates more human-like text.

Readability is evaluated using the Flesch-Kincaid Reading-Ease score (Flesch, 1948), which estimates how difficult it is to understand a passage based on the number of words, sentences, and syllables. We implement the metric using the `textstat` package.[2] Lower scores indicate more complex text, typically characterised by longer sentences and higher lexical density.

4.4 Machine-generated text detection

Machine-generated text detection methods can be divided into neural-based (Hu et al., 2023; Bhattacharjee et al., 2023) and metric-based approaches (Mitchell et al., 2023; Hans et al., 2024). The former use statistical features, often extracted from surrogate LLMs, to detect machine-generated text, whereas the latter are based on machine learning, such as fine-tuning a small pre-trained language model with a binary classification head. Here we deploy a neural classifier due to reported state-of-the-art (SOTA) performance on relevant machine-generated text detection benchmarks (Wang et al., 2024a; Li et al., 2024).

Our detector is based on an encoder-only transformer model with a sequence classification head that maps the `CLS` token representation to logits, $\mathbf{z}_i$, which are converted to pseudo-probabilities using a sigmoid function, σ. As LLM training is considerably resource-intensive, any data filtering or sampling methods must be able to efficiently process large quantities of data with minimal computational overhead (Wenzek et al., 2020b). With this in consideration, we evaluated the base variants of 3 pre-trained language models with under 200 million parameters: `RoBERTa` (Goyal et al., 2020) and `DeBERTav3` (He et al., 2023) due to their SOTA performance in machine-generated text detection (Li et al., 2024; Wang et al., 2024b) and `ModernBERT` (Warner et al., 2024) as a more recent variant that has achieved superior performance on a range of benchmarks. The added advantages of `ModernBERT` is the large context window (8,192 tokens), superior computational speed, and memory efficiency (Warner et al., 2024). See Appendix B for more details. Despite their strong performance, as with all modern neural networks, the confidence estimates are poorly calibrated (Guo et al., 2017), i.e. they are not representative of the true likelihood. To mitigate overconfidence in the predictions, we applied label smoothing. Additionally, we used temperature scaling to further calibrate the model's predictions. Given the logit vector $\mathbf{z}_i$, the new confidence prediction is $\sigma(\mathbf{z}_i/T)$ where T is a learnable temperature parameter.

5 The impact of decoding strategies on model collapse

We carry out recursive training as described in section 4.1 on the open-ended text generation task by fine-tuning LLMs on the WikiText-2 dataset (Merity et al., 2016) with `GPT-2` 124M (Radford et al., 2019) and `SmolLM2` 360M (Allal et al., 2025). The Wikipedia articles are segmented into non-overlapping chunks of 512 tokens, where the first 256 are used as the context ($\mathbf{x}$), and the remaining 256 as the continuation ($\hat{\mathbf{x}}$). We conduct full fine-tuning for 1 epoch and, to avoid information leakage between generation and training, define cross-entropy loss only on the generated sequence of each sample (Dohmatob et al., 2024). Additional details can be found in Appendix A.3. We evaluate a range of decoding strategies to assess the effect on model collapse: greedy decoding ($\tau = 0$), 5-way beam search, pure sampling ($\tau = 1$), temperature ($\tau = 0.9$), top-k (with $k = 50$), and nucleus

[2] `textstat` Python package, textstat.org.

Model	Decoding	Perplexity ↓		Accuracy ↑		Diversity ↑		Self-BLEU ↓		MAUVE ↑		Readability ↑	
		Gen 0	Gen 9	Gen 0	Gen 9	Gen 0	Gen 9	Gen 0	Gen 9	Gen 0	Gen 9	Gen 0	Gen 9
GPT-2	greedy	29.29	82.74	38.72	34.93	0.96	0.70	61.02	67.13	0.99	1.00	60.47	8.25
	beam search	29.25	78.06	38.75	35.21	16.78	10.86	61.54	67.60	0.91	1.29	60.97	17.57
	pure sampling	29.29	58.64	38.74	32.49	94.88	99.82	24.12	6.76	90.16	7.18	40.62	−10.14
	temperature	29.23	44.55	38.77	34.47	87.76	25.10	33.45	54.56	94.15	22.69	46.80	**36.79**
	top-k	29.31	48.36	38.73	33.12	84.57	70.20	38.81	**42.14**	95.21	**70.01**	51.19	34.32
	nucleus	29.28	48.96	38.74	32.73	92.26	**86.73**	28.24	26.86	90.96	57.41	43.96	21.31
SmolLM2	greedy	13.96	85.69	47.58	43.76	6.68	2.22	57.54	50.12	3.23	0.99	62.98	47.04
	beam search	13.96	84.75	47.59	43.77	6.64	2.16	57.30	50.38	3.06	0.89	62.69	45.87
	pure sampling	13.96	15.39	47.58	46.59	90.74	**90.72**	47.23	**45.66**	86.00	**82.80**	47.55	**47.59**
	temperature	13.96	24.88	47.55	46.05	82.92	24.81	51.15	57.55	89.94	17.60	52.83	64.09
	top-k	13.96	19.86	47.59	46.02	82.72	56.77	52.62	57.91	85.97	59.84	55.52	63.18
	nucleus	13.96	22.81	47.58	46.23	87.33	44.27	49.62	58.39	92.39	53.00	51.20	64.17
Human		—		—		88.79		42.89		100		50.34	

Table 1: Impact of decoding strategies on the model performance and text generation quality(comparison between generations 0 and 9) in the fully synthetic recursive training setting. **Bold font** indicates the closest score to the human reference for generation 9 (↑ / ↓: higher / lower is better).

sampling ($\eta = 0.95$). The hyperparameter settings for these decoding strategies were based on recommendations from prior work (Holtzman et al., 2020; Shumailov et al., 2024; Arias et al., 2025).

Figure 1 depicts the perplexity and evaluation accuracy on the WikiText-2 test set for every model generation. Additionally, we obtain scores for the qualitative metrics using the outputs generated by the model (i.e. $\{\hat{\mathbf{x}}\}_{s=1}^{n}$ of $\mathcal{D}_{\text{S}}^{i}$ in Algorithm 1), and enumerate them in Table 1 for generations 0 and 9. We observe that deterministic decoding leads to the most severe model collapse. While stochastic sampling methods exhibit linear degradation across generations, collapse accelerates under greedy decoding and beam search before plateauing in later generations. At generation 0, deterministic strategies yield significantly less fluent and more repetitive text, with MAUVE scores below 5% and diversity scores less than 20%. The disparity in generation quality between deterministic and stochastic strategies in the open-ended text generation task has been explored in related literature (Holtzman et al., 2020; Pillutla et al., 2021). Here, we demonstrate that this disparity compounds across recursive training, resulting in significantly higher perplexity at generation 9. While deterministic methods are rarely used in open-ended generation, we included them for completeness and due to the choice of beam-search in the experiments of Shumailov et al. (2024), but exclude them in subsequent experiments due to their unrealistic collapse.

Pure sampling impacted the models differently. For `GPT-2`, sampling directly from the probability distribution produces diverse and fluent text at generation 0, but training recursively on these outputs results in the worst test perplexity among stochastic methods (58.64) and generated text that has low similarity to human text (MAUVE 7.18). In contrast, with `SmolLM2`, pure sampling yields the smallest decline in task performance and maintains the closest overall similarity to human references across all evaluated metrics.

Temperature sampling led to the most repetitive outputs after recursive training, with diversity decreasing by ∼70% in both models. For `SmolLM2`, it also resulted in the greatest semantic divergence from human-generated text, indicating pronounced model collapse. Performance with top-k sampling was consistent across models, with the smallest decline in diversity and Self-BLEU, the closest text to the human reference, and a smaller drop in test performance compared to nucleus sampling.

In our subsequent experiments on preventing model collapse, we seek to validate that our method can work in the most extreme scenario. For this reason, we evaluate the models using the worst-performing stochastic decoding method (pure sampling for `GPT-2` and temperature sampling for `SmolLM2`). In addition, to facilitate direct comparisons, we also evaluate with top-k decoding due to the consistent performance across models.

6 Preventing model collapse

So far, we have carried out recursive training in a setting where models are trained exclusively on the outputs of the previous generation without implicitly including any human-generated samples. We now turn our focus to the partially synthetic setting, a more realistic scenario where human data make up a portion of the training dataset and the synthetic data is a mix of the samples produced across generations. The training dataset for generation i, $\mathcal{D}^i$, consists of the aggregation of 3 samples:

$$\begin{aligned}\mathcal{D}^i \sim\ & \text{sample}_{i\geq 1}\ \mathcal{D}_{\text{H}}, \alpha \\ & \text{sample}_{i\geq 1}\ \mathcal{D}_{\text{S}}^i, \beta \\ & \text{sample}_{i\geq 2}\ \left\{\mathcal{D}_{\text{S}}^{i-1}, \ldots, \mathcal{D}_{\text{S}}^1\right\}, \frac{\gamma}{(i-1)}\ ,\end{aligned} \quad (3)$$

where α, β, and $\gamma \in [0, 1]$ are mixing coefficients that affect the distribution of human and machine-generated data as well as the proportion of cross-generational data in the training set.

We explore the following settings: (i) fully synthetic $(\alpha{=}0, \beta{=}1, \gamma{=}0)$, where training data consists entirely of synthetic samples from the previous generation, (ii) partially synthetic $(\alpha{>}0, \beta{=}1, \gamma{=}0)$, where the same proportion of human data is added to the training data at every generation, and (iii) partially synthetic with synthetic data accumulated across generations $(\alpha{=}0.5, \beta{=}0.5, \gamma{=}0.5)$ as proposed in (Gerstgrasser et al., 2024). We evaluate our method in the partially synthetic setting and vary the mixing coefficients α, β, and γ, however, our method does not assume access to the values of the mixing coefficients and hence the data distribution. To prevent model collapse when the origin of each training sample is unknown, we train a machine-generated detector that estimates the likelihood of text origin (section 6.1). We then use this information to conduct importance sampling (section 6.2) that ultimately mitigates model collapse.

6.1 Machine-generated text detection performance

We trained and evaluated `RoBERTa`, `DeBERTav3` and `ModernBERT` on the MAGE dataset (Li et al., 2024), a machine-generated text detection benchmark based on documents from 10 domains which have been used to generate text from 27 LLMs. We adopt the preset training / validation / test splits (80%/10%/10%). We also test on the more demanding out-of-distribution test set that contains human-curated text from 4 unseen domains and machine-generated samples by an unseen LLM (`GPT-4`). Each model was fine-tuned for 5 epochs using a binary cross-entropy loss. More details on the model training can be found in Appendix B. Performance is enumerated in Table 2. `ModernBERT` yielded the best classification performance on both aforementioned test sets with an AUC of .986 and .943, respectively. This is comparable to the top-performing model evaluated by Li et al. (2024), Longformer, which achieved an in-distribution and out-of-distribution AUC of .99 and .94, respectively.

Model	in-distribution			out-of-distribution		
	AUC	Acc.	F1	AUC	Acc.	F1
`RoBERTa`	.982	.940	.940	.846	.806	.804
`DeBERTav3`	.971	**.954**	**.954**	.817	.812	.810
`ModernBERT`	**.986**	.948	.948	**.943**	**.861**	**.860**

Table 2: Machine-generated text detection performance. Accuracy (Acc.) and F1-score are macro-averages.

6.2 Informed sampling of training data

Given a dataset at generation i, $\mathcal{D}_i \sim g(x)$, composed of an unknown mixture of human and synthetic samples, our goal is to sample a training dataset from a target human data distribution $\mathcal{D}_{\text{H}} \sim h(x)$ to prevent model collapse. We consider that a language model has collapsed if the inclusion of synthetic samples in the training data results in degraded performance compared to training exclusively on human samples.

We use Sampling Importance Resampling (SIR) (Rubin, 1988), a method for approximately sampling from a target distribution $h(x)$ based on sampling with importance weights from a proposal distribution $g(x)$ using the normalised likelihood ratio $h(x)/g(x)$. As this ratio is intractable in our case, we instead employ a machine-generated text detector to assign each sample $\mathbf{x}_i$, $\forall i$, a predicted probability $q(\mathbf{x}_i)$ of being machine-generated, which we treat as an approximation for the likelihood ratio.

As the detector has been trained on an unbalanced dataset (29% human samples), the predictions are biased towards attributing text as machine-generated, reflected in the optimal classification threshold of 0.8674 (0/1: human/machine-generated). To ameliorate this, we apply a bias term $b \geq 1$ (see Appendix C) to the probabilities, followed by normalising the weights using

$$w_i = \frac{(1 - q(\mathbf{x}_i))^b}{\sum_{j=1}^{n} (1 - q(\mathbf{x}_j))^b}\ , \quad (4)$$

where $w_i \in [0, 1]$ denotes the weight for sample $\mathbf{x}_i$, $\forall i$. From the n weighted training samples, we draw $k \times n$ samples with replacement, with $k = 1.5$ to allow for a 50% upsampling of the training data.

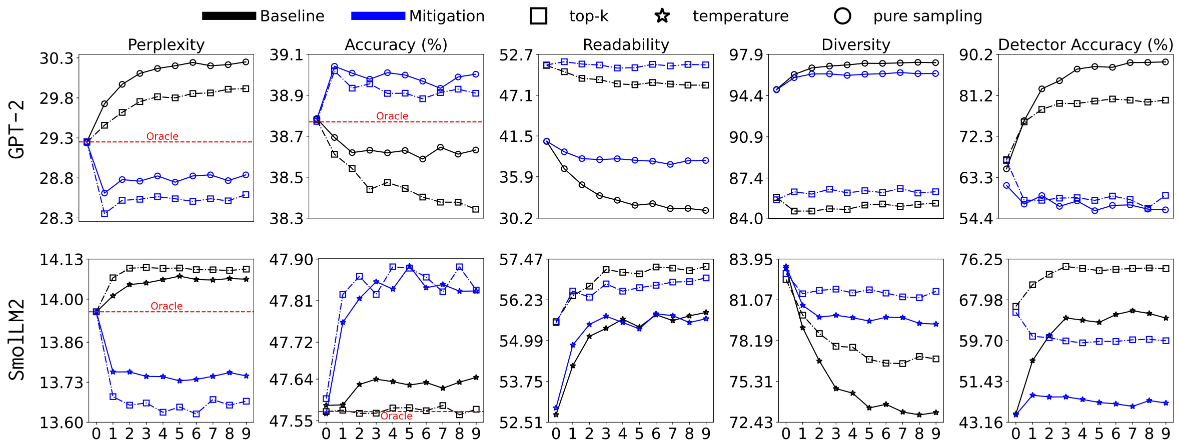

Figure 2: Model collapse mitigation with `GPT-2` or `SmolLM2` under partially synthetic recursive training ($\alpha=1, \beta=1, \gamma=0$) for generations 0 to 9. The baseline is equivalent to training on all the data in the pool and the Oracle performance represents a perfect machine-generated text detector that filters all synthetic samples.

In this way, we obtain a revised set of samples that we use in our recursive training regime.

6.3 Results on collapse prevention

As explained in section 5, we assess our approach by adopting the decoding strategy that caused the most significant model collapse, i.e. pure sampling for `GPT-2` and temperature sampling for `SmolLM2`, and for a more direct comparison, we also conduct experiments using top-k decoding. At each generation i, we compare against the baseline of training on all samples in the pool of data $\mathcal{D}^i$ (Eq. 3). We also provide an "Oracle" performance, which represents a perfect machine-generated text detector that filters all synthetic samples.

We evaluate recursive training in the partially synthetic setting under 3 mixing settings ($\alpha = 1$, $\beta = 1$, $\gamma = 0$), ($\alpha = 0.5$, $\beta = 1$, $\gamma = 0$), ($\alpha = 0.5$, $\beta = 0.5$, $\gamma = 0.5$). The task performance, data quality, and detector accuracy metrics over 10 generations of partially synthetic training are depicted in Figure 2 for the scenario where human and synthetic samples have equal proportion (Appendices S4 and S6 contain results for the other 2 scenarios). Weighted sampling (section 6.2) prevents model collapse and preserves the readability and diversity of the synthetic outputs, while the baseline degrades in task performance and data quality. The baseline generations become increasingly detectable, indicating divergence from human text. Notably, our method improves performance compared to training exclusively on human data (Oracle) across models and decoding strategies. These outcomes demonstrate both the value of using synthetic data in LLM training, but also the importance of selecting the right synthetic samples.

Table 3 enumerates the percentage difference for the final models under the baseline strategy vs. using our approach. We also enumerate the exact scores in Table S2. Our method improves the data quality and model performance across all metrics, except for pure sampling with `GPT-2`, where the baseline shows higher diversity and Self-BLEU but at the cost of lower MAUVE, readability, and task performance, indicating degraded quality. Notably, we observe that mixing cross-generational data has a minimal effect on the extent of model collapse compared to training solely on the previous generation, contrasting with the findings of Gerstgrasser et al. (2024). However, we note that Gerstgrasser et al. (2024) did not constrain the sample size. Our experiments adopt a more realistic and less extreme setting by sampling a fixed dataset under different mixing scenarios.

Additionally, we evaluate the effectiveness of our method at different model scales of `SmolLM2` (135M, 360M, 1.7B) for a fixed decoding strategy and data mixing setting. The results are enumerated at the bottom of Table 3 and the performance across generations is depicted in Figure S5. Our method prevents model collapse across all `SmolLM` variants. Notably, we find that smaller models exhibit greater relative improvements using our method compared to both the Oracle and the baseline.

In line with previous research (Shumailov et al., 2024), we also study the perplexity distribution of the synthetic data at each generation using model p^0 that was trained on human data. Figure 3 de-

Model	Decoding	α, β, γ	Perplexity↓	Accuracy↑	Diversity↑	Self-BLEU↓	MAUVE↑	Readability↑
GPT-2	top-k	.5, 1, 0	−7.28%	+2.37%	+4.54%	−1.77%	+0.46%	+9.03%
		.5, .5, .5	−5.94%	+1.72%	+2.43%	−4.28%	+3.99%	+10.16%
		1, 1, 0	−4.45%	+1.49%	+3.59%	−3.58%	+1.36%	+6.76%
	pure sampling	.5, 1, 0	−7.41%	+1.50%	−1.30%	+36.71%	+74.06%	+50.23%
		.5, .5, .5	−6.54%	+1.50%	−1.07%	+21.39%	+16.38%	+25.45%
		1, 1, 0	−25.65%	+0.96%	−0.71%	+20.70%	+16.42%	+20.99%
SmolLM2 360M	top-k	.5, 1, 0	-4.60%	+0.68%	+7.06%	-1.17%	+15.28%	-0.73%
		.5, .5, .5	-4.37%	+0.62%	+4.25%	-0.42%	+2.05%	-0.74%
		1, 1, 0	-3.05%	+0.54%	+6.20%	-1.74%	+10.62%	-0.61%
	temperature	.5, 1, 0	-3.91%	+0.55%	+14.62%	-1.46%	+20.92%	-1.86%
		.5, .5, .5	-3.24%	+0.48%	+7.44%	-1.98%	-0.33%	-1.35%
		1, 1, 0	-2.23%	+0.39%	+8.55%	-1.80%	+13.98%	-0.34%
SmolLM2 135M	top-k	1, 1, 0	-4.19%	+0.75%	+9.96%	-2.15%	+13.08%	-0.76%
SmolLM2 1.7B	top-k	1, 1, 0	-0.85%	+0.06%	+3.44%	-1.11%	+8.46%	-0.65%

Table 3: Percentage of change in data quality when using our proposed mitigation strategy versus the baseline. Results are shown for top-k decoding and pure sampling/temperature for different values of α, β, and γ (blue / red: positive / negative results, ↑ / ↓: higher / lower is better).

picts these distributions for generations 0, 1, and 9 for the GPT-2 model with top-k and pure sampling, compared to the baseline (Appendix S3 contains the results for SmolLM2). For top-k decoding, similarly to Shumailov et al. (2024), we observe that for the baseline, perplexity shifts towards regions of lower scores and the distribution becomes more peaked, showing signs of early model collapse. For pure sampling, on the other hand, we observe that the baseline distribution shifts to higher perplexity scores and displays increased variance. This is an interesting finding that demonstrates that by removing truncation from the decoding strategy, the narrowing effect of model collapse is diminished, and instead, model collapse is reflected by long-tail incoherent text that is completely distinct from the original human samples. By deploying our mitigation strategy, however, we observe very little change in the perplexity distribution for both sampling strategies.

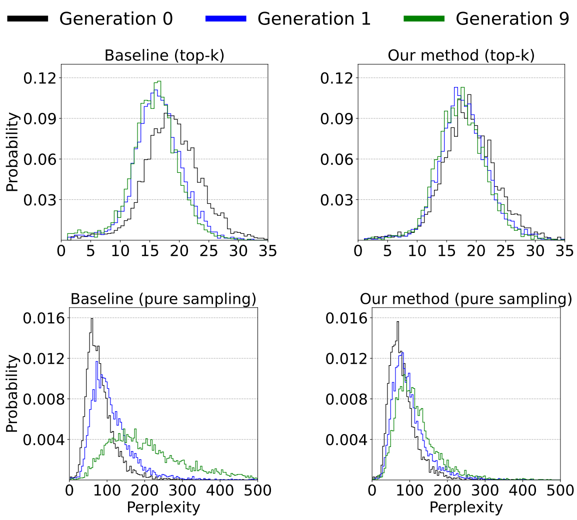

Figure 3: Perplexity distribution for machine-generated text of GPT-2 at generations 0, 1, and 9 in the partially synthetic scenario ($\alpha = 0.5, \beta = 1, \gamma = 0$). Perplexity is evaluated using the model trained on human text (p^0).

7 Conclusion

This work investigates model collapse across three dimensions: model performance, data quality, and resemblance to human samples. Through our analysis, we found that the extent of model collapse and the effect on the data distribution is influenced by the decoding strategy. Truncating can lead to peaked distributions and repetitive models while pure sampling can result in high perplexity and verbose outputs with low resemblance to human data. Using the decoding strategies that resulted in the most extreme collapse, we evaluated the partially synthetic scenario, where human data is mixed into the training data. We designed a novel method to mitigate model collapse based on resampling the training distribution using the predictions of a machine-generated text detector. We have validated our method on two popular model variants (GPT-2 and SmolLM2) across a range of decoding strategies and model sizes, showing that we can prevent model collapse in all cases. When there is an equal ratio of human to synthetic samples in the training pool, our method results in improved model performance compared to training only on the human data.

Limitations

As in previous studies (Shumailov et al., 2023; Dohmatob et al., 2024), we assess LLMs exclusively in a fine-tuning setting rather than pre-training from scratch. While pre-training experiments could provide deeper insights, the computational cost and complexity of training large-scale models from the ground up make such an approach impractical in our case. Nevertheless, given that model collapse has been primarily evaluated in LLMs from a fine-tuning setting, the conclusions made in this work still align with the current body of research.

In addition, our study focuses primarily on open-ended text generation tasks. While this is a crucial area for understanding model collapse, our findings may not fully generalise to other domains, such as structured prediction or code generation, where the impact of model collapse may manifest differently. Future work could explore whether our resampling method remains effective across these domains.

Finally, our method depends on the performance of the machine-generated text detector used to estimate the importance weights. Any inaccuracies or biases in the detector's predictions directly affect the quality of the resampling process, potentially leading to suboptimal mitigation of model collapse. Since detector performance varies with domain, LLM architecture, and decoding strategy, the generalisability of our approach is closely tied to the detector's robustness. The primary focus of this work, however, is not on optimising the detector itself, but on demonstrating that detector-based resampling can effectively mitigate model collapse. Future work could investigate improved methods of detection, such as using adaptive or ensemble-based detectors to improve reliability across data regimes and LLMs.

Ethics Statement

The authors declare no competing interests. The Wikitext-2 dataset (Merity et al., 2016) used for the experiments is licensed under the Creative Commons Attribution-ShareAlike License (CC BY-SA 4.0). `GPT-2` is licensed under the MIT license and `SmolLM2` is licensed under APACHE 2.0. AI assistants were not used for this research.

References

David H Ackley, Geoffrey E Hinton, and Terrence J Sejnowski. 1985. A learning algorithm for Boltzmann machines. *Cognitive Science*, 9(1):147–169.

Sina Alemohammad, Josue Casco-Rodriguez, Lorenzo Luzi, Ahmed Imtiaz Humayun, Hossein Babaei, Daniel LeJeune, Ali Siahkoohi, and Richard Baraniuk. 2024a. Self-Consuming Generative Models Go MAD. In *International Conference on Learning Representations*.

Sina Alemohammad, Ahmed Imtiaz Humayun, Shruti Agarwal, John P. Collomosse, and Richard G. Baraniuk. 2024b. Self-Improving Diffusion Models with Synthetic Data. *CoRR*, abs/2408.16333.

Loubna Ben Allal, Anton Lozhkov, Elie Bakouch, Gabriel Martín Blázquez, Guilherme Penedo, Lewis Tunstall, Andrés Marafioti, Hynek Kydlíček, Agustín Piqueres Lajarín, Vaibhav Srivastav, et al. 2025. SmolLM2: When Smol Goes Big–Data-Centric Training of a Small Language Model. *arXiv preprint arXiv:2502.02737*.

Esteban Arias, Meimingwei Li, Christian Heumann, and Matthias Assenmacher. 2025. Decoding Decoded: Understanding Hyperparameter Effects in Open-Ended Text Generation. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 9992–10020.

Quentin Bertrand, Joey Bose, Alexandre Duplessis, Marco Jiralerspong, and Gauthier Gidel. 2024. On the Stability of Iterative Retraining of Generative Models on their own Data. In *International Conference on Learning Representations*.

Amrita Bhattacharjee, Tharindu Kumarage, Raha Moraffah, and Huan Liu. 2023. ConDA: Contrastive Domain Adaptation for AI-generated Text Detection. In *Proceedings of the 13th International Joint Conference on Natural Language Processing*, pages 598–610.

Martin Briesch, Dominik Sobania, and Franz Rothlauf. 2023. Large language models suffer from their own output: An analysis of the self-consuming training loop. *arXiv preprint arXiv:2311.16822*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*, pages 4171–4186.

Elvis Dohmatob, Yunzhen Feng, Pu Yang, Francois Charton, and Julia Kempe. 2024. A Tale of Tails: Model Collapse as a Change of Scaling Laws. In *International Conference on Machine Learning*.

Liam Dugan, Alyssa Hwang, Filip Trhlík, Andrew Zhu, Josh Magnus Ludan, Hainiu Xu, Daphne Ippolito, and Chris Callison-Burch. 2024. RAID:

A Shared Benchmark for Robust Evaluation of Machine-Generated Text Detectors. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pages 889–898.

Yunzhen Feng, Elvis Dohmatob, Pu Yang, Francois Charton, and Julia Kempe. 2024. Beyond Model Collapse: Scaling Up with Synthesized Data Requires Reinforcement. In *International Conference on Machine Learning 2024 Workshop on Theoretical Foundations of Foundation Models*.

Rudolph Flesch. 1948. A new readability yardstick. *Journal of applied psychology*, 32(3):221.

Matthias Gerstgrasser, Rylan Schaeffer, Apratim Dey, Rafael Rafailov, Henry Sleight, John Hughes, Tomasz Korbak, Rajashree Agrawal, Dhruv Pai, Andrey Gromov, et al. 2024. Is Model Collapse Inevitable? Breaking the Curse of Recursion by Accumulating Real and Synthetic Data. *arXiv preprint arXiv:2404.01413*.

Saurabh Goyal, Anamitra Roy Choudhury, Saurabh Raje, Venkatesan Chakaravarthy, Yogish Sabharwal, and Ashish Verma. 2020. PoWER-BERT: Accelerating BERT inference via progressive word-vector elimination. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pages 3690–3699.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. 2017. On calibration of modern neural networks. In *International conference on machine learning*, pages 1321–1330.

Yanzhu Guo, Guokan Shang, Michalis Vazirgiannis, and Chloé Clavel. 2024. The Curious Decline of Linguistic Diversity: Training Language Models on Synthetic Text. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 3589–3604.

Abhimanyu Hans, Avi Schwarzschild, Valeriia Cherepanova, Hamid Kazemi, Aniruddha Saha, Micah Goldblum, Jonas Geiping, and Tom Goldstein. 2024. Spotting LLMs with binoculars: Zero-shot detection of machine-generated text. *arXiv preprint arXiv:2401.12070*.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2023. DeBERTaV3: Improving DeBERTa using ELECTRA-Style Pre-Training with Gradient-Disentangled Embedding Sharing. In *International Conference on Learning Representations*.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The Curious Case of Neural Text Degeneration. In *International Conference on Learning Representations*.

Cheng-Yu Hsieh, Chun-Liang Li, Chih-kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alex Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. 2023. Distilling Step-by-Step! Outperforming Larger Language Models with Less Training Data and Smaller Model Sizes. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8003–8017. Association for Computational Linguistics.

Xiaomeng Hu, Pin-Yu Chen, and Tsung-Yi Ho. 2023. Radar: Robust ai-text detection via adversarial learning. *Advances in neural information processing systems*, 36:15077–15095.

Yafu Li, Qintong Li, Leyang Cui, Wei Bi, Zhilin Wang, Longyue Wang, Linyi Yang, Shuming Shi, and Yue Zhang. 2024. MAGE: Machine-generated Text Detection in the Wild. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, pages 36–53.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*.

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D Manning, and Chelsea Finn. 2023. Detectgpt: Zero-shot machine-generated text detection using probability curvature. In *International Conference on Machine Learning*, pages 24950–24962.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Krishna Pillutla, Swabha Swayamdipta, Rowan Zellers, John Thickstun, Sean Welleck, Yejin Choi, and Zaid Harchaoui. 2021. Mauve: Measuring the gap between neural text and human text using divergence frontiers. *Advances in Neural Information Processing Systems*, 34:4816–4828.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners.

Rubin. 1988. Using the SIR algorithm to simulate posterior distributions. In *Bayesian statistics 3*, pages 395–402.

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

Ilia Shumailov, Zakhar Shumaylov, Yiren Zhao, Nicolas Papernot, Ross Anderson, and Yarin Gal. 2024. AI models collapse when trained on recursively generated data. *Nature*, 631(8022):755–759.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. Self-Instruct: Aligning Language Models with Self-Generated Instructions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13484–13508. Association for Computational Linguistics.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Osama Mohammed Afzal, Tarek Mahmoud, Giovanni Puccetti, and Thomas Arnold. 2024a. SemEval-2024 task 8: Multidomain, multimodel and multilingual machine-generated text detection. In *Proceedings of the 18th International Workshop on Semantic Evaluation*, pages 2057–2079.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Osama Mohammed Afzal, Tarek Mahmoud, Giovanni Puccetti, Thomas Arnold, Alham Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2024b. M4GT-bench: Evaluation benchmark for black-box machine-generated text detection. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, pages 3964–3992.

Benjamin Warner, Antoine Chaffin, Benjamin Clavié, Orion Weller, Oskar Hallström, Said Taghadouini, Alexis Gallagher, Raja Biswas, Faisal Ladhak, Tom Aarsen, et al. 2024. Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference. *arXiv preprint arXiv:2412.13663*.

Guillaume Wenzek, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzmán, Armand Joulin, and Edouard Grave. 2020a. CCNet: Extracting high quality monolingual datasets from web crawl data. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4003–4012.

Guillaume Wenzek, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzmán, Armand Joulin, and Edouard Grave. 2020b. CCNet: Extracting high quality monolingual datasets from web crawl data. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4003–4012, Marseille, France. European Language Resources Association.

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

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

Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. 2018. Texygen: A benchmarking platform for text generation models. In *The 41st international ACM SIGIR conference on research & development in information retrieval*, pages 1097–1100.

Appendix

A Recursive training

A.1 Dataset

All our model collapse experiments use the raw variant of the WikiText-2 dataset (Merity et al., 2016). We train models on the training set, consisting of 36,718 documents, and evaluate on the test set of 4,358 documents. The WikiText-2 dataset was extracted from the 'Good' or 'Featured' article criteria specified by editors on Wikipedia and only covers the English language.

A.2 LLMs

`GPT-2` (Generative Pre-trained Transformer 2) (Radford et al., 2019) is a decoder-only transformer-based language model. `GPT-2` demonstrated that large-scale language models could perform various language tasks without task-specific training. We use the base variant, which contains 124M parameters. `SmolLM2` (Allal et al., 2025) is a family of compact and efficient language models developed by Hugging Face, available in three sizes: 135M, 360M, and 1.7B parameters. The majority of our experiments use the 360M parameter variant unless specified otherwise.

A.3 Hyperparameters

In our experiments, we conduct full fine-tuning using a learning rate of 5×10^{-5}, batch size of 8 and a dropout rate of 0.1. For the AdamW optimizer, we set $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$. Each model was trained for 1 epoch with the hyperparameters fixed for all experiments. We conducted 10 iterations of recursive training.

B Machine-generated text detection

B.1 Pre-trained models

Robustly Optimized `BERT` pre-training Approach (`RoBERTa`) by Liu et al. (2019) improves on the pre-training phase of `BERT` (Devlin et al., 2019), an encoder-only transformer model that leverages masked language models to enable pre-trained deep bidirectional representations. The `RoBERTa` model optimised the pre-training procedure for `BERT` by training the model for longer and on more data, changing the masking pattern, and removing the next sentence prediction objective. We use the base variant which has 125M parameters.

Decoding-enhanced `BERT` with Disentangled Attention (`DeBERTav3`) by He et al. (2023), is a `BERT`-based encoder only model enhanced with disentangled attention. `DeBERTav3` model improves on `DeBERTa` by using Enhanced Mask Decoding and an `ELECTRA`-style pre-training objective, Replaced Token Detection, instead of Masked Language Modelling. We use the base variant which contains 86M backbone parameters with an embedding layer of 98M parameters.

`ModernBERT` (Warner et al., 2024) is a recent addition to the encoder-only transformer models that has been designed to increase downstream performance and efficiency on GPUs, particularly in long-context scenarios due to its 8,192 native sequence length. The model was trained on 2 trillion tokens and improves on the original `BERT` architecture with rotary positional embeddings (RoPE), unpadding, GeGLU layers and alternating local-global attention demonstrating SOTA performance amongst encoder models across a range of classification and retrieval tasks. We conduct experiments with the base variant, which contains 150M parameters.

B.2 Hyperparameters

Each model was fine-tuned for 5 epochs. We select the best model based on the highest AUC on the validation set. Optimisation was performed using AdamW by setting $\beta_1 = 0.9$, $\beta_2 = 0.98$, $\epsilon = 10^{-6}$, and the weight decay to 10^{-2}. These parameters were chosen based on prior work (Warner et al., 2024). The label smoothing parameter α was set to 0.1, the seed was fixed at 42 and the training batch size to 16. The learning rate was set based on a hyperparameter sweep over $[1, 1.5, 2, 3, 4] \times 10^{-5}$. For `ModernBERT` the best-performing learning rate was 10^{-5}. We implemented temperature scaling by learning the temperature parameter using L-BFGS optimisation on the validation set. This was run for 50 iterations with a learning rate of 0.01.

B.3 Dataset

We trained and evaluated the machine-generated text detectors on the MAGE dataset (Li et al., 2024), which is based on documents from 10 domains: opinion statements (CMV & Yelp reviews dataset), news articles (XSum & TLDR dataset), question answering (ELI5), story generation (Reddit WritingPrompts & ROC), commonsense reasoning (HellaSwag), knowledge illustration (SQuAD) and Scientific writing (SciGen). The authors sampled 1,000 texts from each domain (apart from opinion statements and news articles with 804 and 777 samples respectively) and generated text using 27 LLMs from 7 model families, which include OpenAI, LLaMA, GLM, FLAN-T5, OPT, BigScience and EleutherAI. For each human-written sample in the dataset, they generate a machine-generated version by providing the first 30 tokens of human-written text as context to the LLM. In addition, for the OpenAI models, they implemented two other prompt strategies for relevant domains: 'topical' prompts such as an argument or news title and 'specified' prompts which contain information about the domain source. This results in 33,000 ($= 27{,}000 + 3 \times 2 \times 1{,}000$) machine-generated samples per source before processing and filtering. The authors split the dataset into train, validation and test splits in the ratio 80:10:10. To mitigate data imbalance in the validation and test sets they sample additional human data from each data source. The resulting test set contains 28,741 human and 28,078 machine-generated samples (49% machine-generated). The training set, however, is 71% machine-generated. The total dataset consists of 154,078 human-written and 294,381 machine-generated texts. In addition to the previously described test set, we also evaluate our detector on their more challenging test set containing text from four unseen domains (CNN/DailyMail, DialogSum, PubMedQA, IMDB) and generated by an unseen model (`GPT-4`). This out-of-distribution test set contains 762 human and 800 machine-generated samples.

When evaluating the `ModernBERT` model finetuned on MAGE on the `SmolLM2` models, we observed a drop in performance compared to `GPT-2`, with large variability across decoding strategies and model size. For `SmolLM2` 360M the detector achieved a classification accuracy of .601 for top-k decoding and .399 for temperature sampling. To ameliorate this, we finetuned a new `ModernBERT` model on a larger corpus, containing the MAGE dataset and a subset of the RAID dataset (Dugan et al., 2024) for the `SmolLM2` models. The RAID dataset is the largest machine-generated text detection dataset, includes text samples generated by 11 LLMs with 4 decoding strategies, and spans text across 8 domains. Additionally, RAID includes 11 types of adversarial attacks, such as homoglyph substitutions, number insertions, article deletions, and paraphrasing. We partitioned the dataset into training, validation, and test splits in the ratio 80:10:10, ensuring no cross-contamination of text segments generated from the same source of text across splits. We balanced each split so that it contained an equal number of human and machine samples, stratified across model, decoding strategy, source domain, and adversarial attack (the whitespace and paragraph attacks were included). This resulted in balanced train, validation, and test splits comprising 128,352, 16,044, and 16,056 samples, respectively.

Model	$r_{\max}$	b
`GPT-2`	10	10
`SmolLM 135M`	10	10
`SmolLM 360M`	5	10
`SmolLM 1.7B`	3	1

Table S1: Optimal hyperparameters for Sampling Importance Resampling across different model scales using top-k decoding.

C Informed sampling of training data

As we perform sampling with replacement, there is the risk of excessive duplication of high-weight samples. To account for this, we introduce a maximum resample count parameter, $r_{\max}$, which limits the number of times any individual sample can be selected. This constraint ensures diversity in the resampled dataset and prevents a small subset of high-weight samples from dominating the training distribution. To further correct for the classifier's bias toward labelling samples as machine-generated, we introduce a bias term $b \geq 1$ to adjust the weight distribution. This formulation increases the selection probability of samples likely to be human, counteracting the bias introduced by the classifier's skewed confidence distribution. We select values for $r_{\max}$ and b by evaluating each model on the Wikitext-2 validation set after 1 generation of recursive training. The optimal hyperparameters for each model configuration are reported in Table S1.

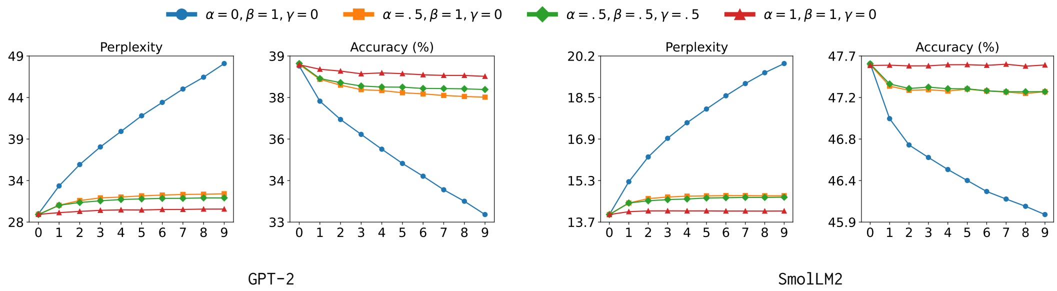

Figure S1: Perplexity and accuracy over generations 0 to 9 of fully synthetic recursive training for varying mixing coefficients (α, β, γ) using top-k decoding.

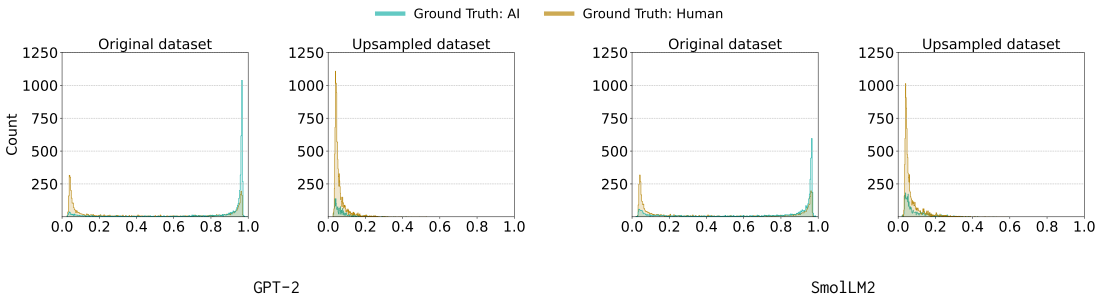

Figure S2: Classification score distribution of the machine-generated text detector on the dataset from generation 0.

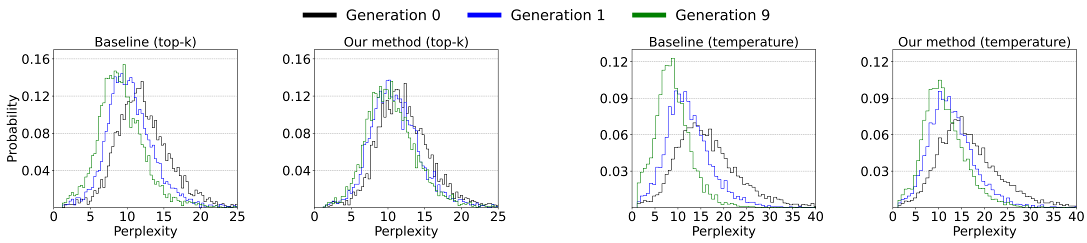

Figure S3: Perplexity distribution for machine-generated text of SmolLM2 at generations 0, 1, and 9 in the partially synthetic scenario ($\alpha=0.5, \beta=1, \gamma=0$). Perplexity is evaluated using the model trained on human text (p^0).

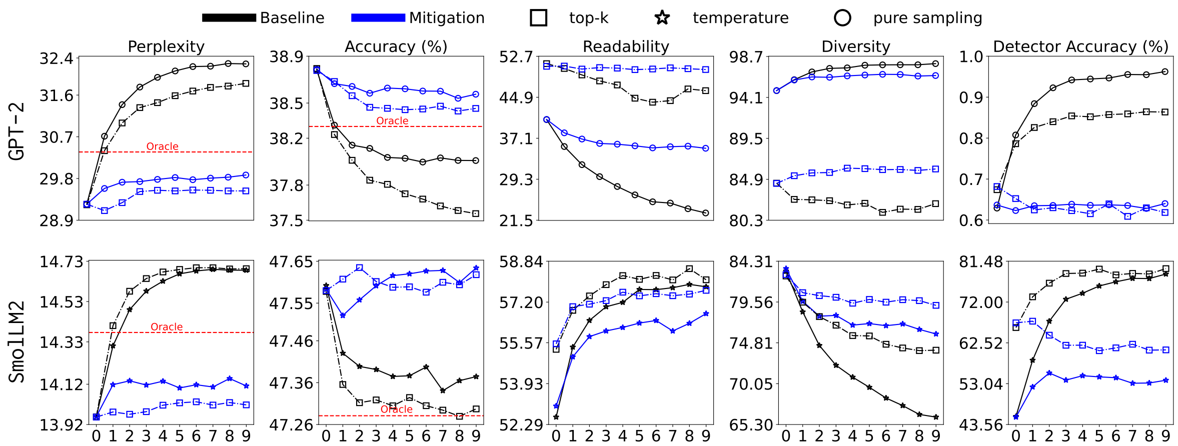

Figure S4: GPT-2 (top) and SmolLM2 (bottom) under partially synthetic recursive training ($\alpha=0.5, \beta=1, \gamma=0$) for 10 generations. The baseline is equivalent to training on all the data in the pool and the Oracle performance represents a perfect AI text detector that filters all synthetic samples.

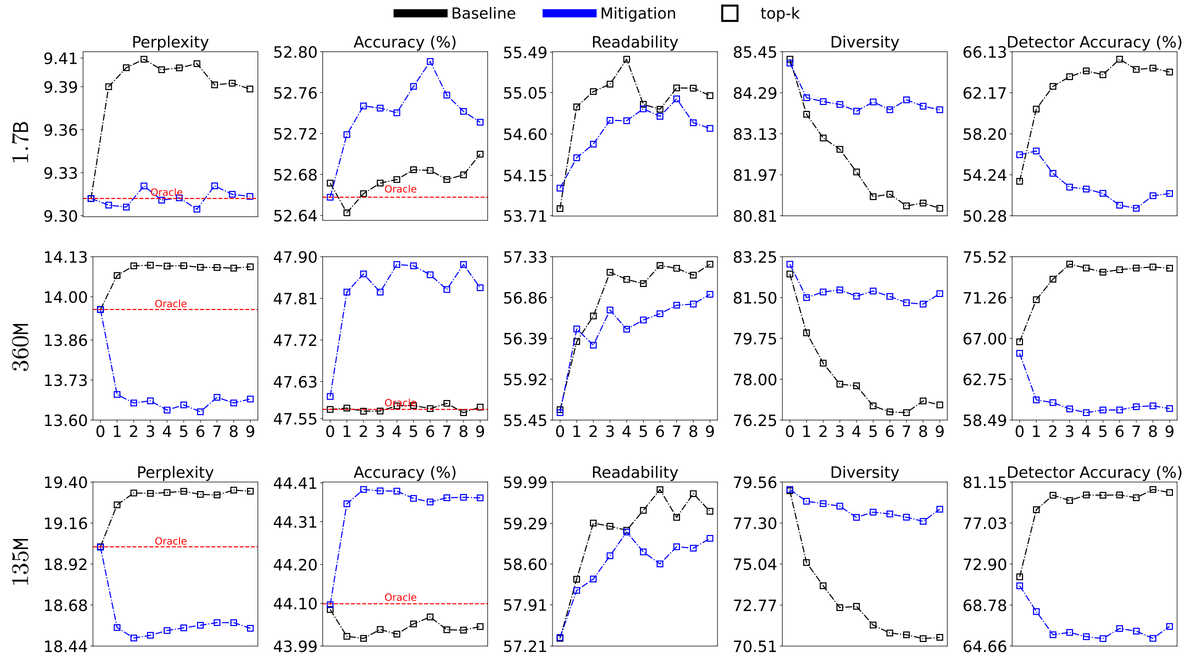

Figure S5: `SmolLM2` model size variants (1.7B, 360M, 135M) under partially synthetic recursive training ($\alpha=1, \beta=1, \gamma=0$) for 10 generations. The baseline is equivalent to training on all the data in the pool and the Oracle performance represents a perfect AI text detector that filters all synthetic samples.

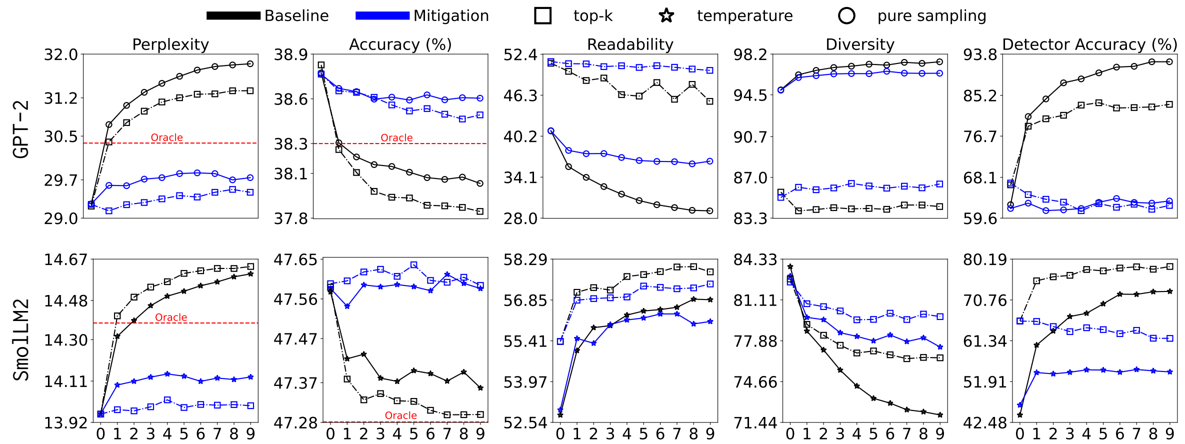

Figure S6: `GPT-2` (top) and `SmolLM2` (bottom) under partially synthetic recursive training with cross-generational data ($\alpha=0.5, \beta=0.5, \gamma=0.5$) for 10 generations. The baseline is equivalent to training on all the data in the pool and the Oracle performance represents a perfect AI text detector that filters all synthetic samples.

Model	Method	Decoding	α, β, γ	Perplexity↓		Accuracy↑		Diversity↑		Self-BLEU↓		MAUVE↑		Readability↑	
				Gen 0	Gen 9	Gen 0	Gen 9	Gen 0	Gen 9	Gen 0	Gen 9	Gen 0	Gen 9	Gen 0	Gen 9
GPT-2	baseline	top-k	.5, 1, 0	29.23	31.85	38.80	37.56	84.46	82.16	39.10	39.55	93.31	91.10	51.30	46.16
			.5, .5, .5	29.22	31.33	38.83	37.84	85.66	84.35	38.94	40.64	94.07	92.00	51.06	45.38
			1, 1, 0	29.25	29.92	38.78	38.34	84.66	82.68	39.19	40.20	93.24	92.69	50.99	48.09
		sampling	.5, 1, 0	29.25	32.27	38.78	38.01	94.86	97.77	24.09	14.52	91.08	46.87	40.64	23.69
			.5, .5, .5	29.26	31.82	38.77	38.03	94.95	97.50	24.15	17.02	90.07	76.01	41.02	29.08
			1, 1, 0	29.25	38.79	38.79	38.63	94.88	96.86	24.04	18.12	91.18	75.64	40.71	31.49
	ours	top-k	.5, 1, 0	29.25	29.53	38.78	38.45	84.43	85.89	39.43	38.85	94.59	91.52	50.79	50.33
			.5, .5, .5	29.25	29.47	38.77	38.49	85.18	86.40	39.28	38.90	94.75	95.67	51.31	49.99
			1, 1, 0	29.25	28.59	38.77	38.91	84.81	85.65	38.85	38.76	94.92	93.95	51.28	51.34
		pure sampling	.5, 1, 0	29.25	29.88	38.79	38.58	94.87	96.50	24.30	19.85	92.92	81.58	40.68	35.59
			.5, .5, .5	29.26	29.74	38.76	38.60	94.92	96.46	24.09	20.66	91.16	88.46	40.98	36.48
			1, 1, 0	29.24	28.84	38.78	39.00	94.61	96.17	24.01	21.87	91.76	88.06	40.78	38.10
SmolLM2 350M	baseline	top-k	.5, 1, 0	13.96	14.69	47.58	47.30	82.59	73.95	52.68	54.53	91.58	78.42	55.31	58.10
			.5, .5, .5	13.96	14.64	47.59	47.30	82.76	76.54	52.57	53.97	91.03	86.15	55.38	57.84
			1, 1, 0	13.96	14.10	47.57	47.57	82.51	76.90	52.47	54.17	91.32	81.11	55.57	57.24
		temperature	.5, 1, 0	13.96	14.69	47.59	47.37	83.16	66.16	51.21	54.46	83.67	70.15	52.59	57.81
			.5, .5, .5	13.96	14.60	47.58	47.36	83.74	72.03	51.22	54.15	86.94	81.27	52.80	56.86
			1, 1, 0	13.96	14.06	47.58	47.64	83.31	73.11	51.16	53.96	84.55	76.27	52.74	55.85
	ours	top-k	.5, 1, 0	13.96	14.02	47.58	47.62	82.75	79.17	52.58	53.89	89.33	90.39	55.53	57.67
			.5, .5, .5	13.96	14.00	47.59	47.59	82.52	79.79	52.64	53.74	88.69	87.92	55.39	57.41
			1, 1, 0	13.96	13.67	47.60	47.83	82.93	81.67	52.74	53.23	85.09	89.72	55.53	56.90
		temperature	.5, 1, 0	13.96	14.11	47.58	47.63	83.45	75.84	51.15	53.67	87.80	84.82	53.03	56.74
			.5, .5, .5	13.96	14.13	47.58	47.58	82.99	77.39	51.12	53.08	86.20	81.00	52.98	56.09
			1, 1, 0	13.96	13.75	47.57	47.83	83.43	79.36	51.21	52.99	88.16	86.94	52.94	55.66
SmolLM2 135M	baseline	top-k	1, 1, 0	19.02	19.35	44.08	44.04	79.08	70.99	53.08	54.87	85.18	72.73	57.34	59.49
	ours			19.02	18.54	44.10	44.37	79.15	78.06	52.90	53.69	82.92	82.24	57.35	59.04
SmolLM2 1.7B	baseline	top-k	1, 1, 0	9.31	9.39	52.67	52.70	85.24	81.02	53.22	54.28	91.11	85.95	53.79	55.02
	ours			9.31	9.31	52.66	52.73	85.13	83.81	53.16	53.68	88.22	93.23	54.01	54.66

Table S2: Test performance (perplexity and accuracy) and data quality at generation 0 and generation 9. Results are shown for top-k decoding and pure sampling/temperature for different values of α, β, and γ (↑ / ↓: higher / lower is better).