Adaptive NEON: Mitigating Model Collapse via Block-Wise Extrapolation

Nir Soffer nirsof@post.bgu.ac.il
Amitay Granada granada@post.bgu.ac.il
Department of Industrial Engineering and Management
Ben-Gurion University of the Negev

February 17, 2026

Abstract

Generative AI models require vast amounts of high-quality data, yet the availability of real-world data is diminishing. Training models on synthetic data, however, leads to *Model Collapse* - a degradation in sample quality and diversity. Recent research, *NEON: Negative Extrapolation From Self-Training Improves Image Generation* [1], showed that this collapse follows a predictable direction in parameter space and can be mitigated via *Negative Extrapolation*.

In this work, we extend the NEON method by proposing a Block-Wise Adaptive Extrapolation approach. Instead of applying a single global scalar correction, we leverage the hierarchical structure of the U-Net architecture (Encoder, Middle, Decoder) and assign distinct extrapolation weights to each block.

We evaluate our method on both a controlled 2D Gaussian toy problem and using Elucidated Diffusion Models (EDM) on the CIFAR-10 dataset. Our results suggest that collapse dynamics vary across architectural components, and that block or layer-specific correction can outperform global extrapolation.

The code for this project is available at: https://github.com/NirSof/BGU-IEM-DL-Adaptive-NEON

1 Introduction

The rapid evolution of generative AI is fundamentally constrained by the scarcity of high-quality real-world data. To address this limitation, the field is increasingly adopting self-training, a paradigm where a model is refined using synthetic data generated by previous iterations of itself. While theoretically scalable, recent studies indicate that this "self-consuming" loop leads to a degenerative process known as Model Collapse or Model Autophagy Disorder (MAD) [2, 3].

This phenomenon is fundamentally characterized by a significant degradation in both the sample quality and diversity of the generated content. As the model recursively learns from its own synthetic outputs, it tends to lose the "tails" of the original distribution - the rare and complex examples - and converges toward a homogenized mean representation [2]. Consequently, the generative variance decreases, and the model becomes "mode-seeking," failing to capture the full richness of the real-world data.

A pivotal insight from recent literature, specifically the NEON framework, is that this degradation is not random. [1] demonstrated that the collapse follows a predictable linear trajectory in the parameter space. Based on this finding, they proposed Negative Extrapolation: a method to actively "push" the model parameters in the opposite direction of the drift using a single global scalar, thereby restoring the model's performance.

In this project, we build upon the NEON hypothesis but identify a critical limitation in its application. We argue that applying a uniform correction weight across the entire network is suboptimal, as different blocks or layers may drift at different rates. We propose a Block-Wise Adaptive Extrapolation approach, which decomposes the U-Net architecture into functional blocks (Encoder, Middle, Decoder) and assigns distinct extrapolation weights to each. We demonstrate that this granular correction strategy effectively mitigates model collapse and outperforms the global baseline, as evidenced by our experiments on a 2D Gaussian toy problem and Elucidated Diffusion Models (EDM) trained on CIFAR-10.

2 Background

Our work builds directly upon the NEON framework [1]. The core insight of NEON is that model collapse, driven by self-training on synthetic data, is not merely random noise accumulation. Rather, it is a structural degradation stemming from the *mode-seeking* nature of generative models, which tend to oversample high-probability regions while neglecting the distribution tails. This results in a predictable directional drift in parameter space.

Let θ_r denote the original model trained on real data, and θ_s denote the model after fine-tuning on its own synthetic outputs. The vector $(\theta_s - \theta_r)$ represents the *collapse direction*. NEON proposes to counteract this drift via *Negative Extrapolation*, applying a correction using the following formula:

$$\theta_{\text{NEON}} = \theta_r - w \cdot (\theta_s - \theta_r)$$

where $w > 0$ is a global scalar extrapolation weight applied uniformly across all network parameters. By subtracting the collapse vector, the method actively pushes the model parameters away from the degradation path.

The authors validated this approach across various architectures (e.g., Diffusion models, Flow Matching and Auto-Regressive transformers) and datasets (CIFAR-10, ImageNet), demonstrating that the FID score typically exhibits a U-shaped behavior with respect to w. Optimal negative extrapolation was shown not only to recover the original performance, but in some cases to surpass it by enhancing generation diversity (Recall) with minimal loss of precision.

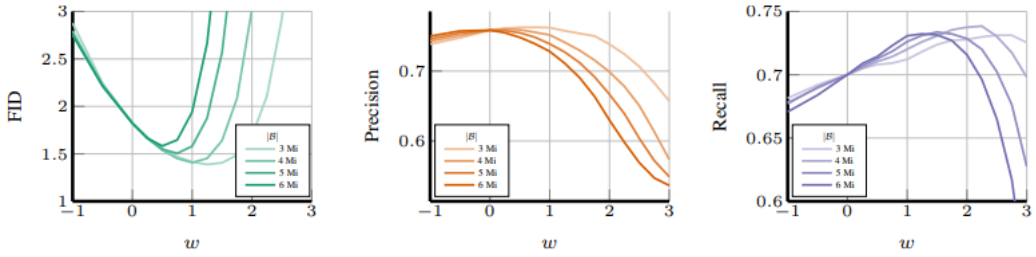


Figure 1: **Impact of Negative Extrapolation strength (w) on generation metrics.** Results for EDM-VP trained on CIFAR-10 showing FID, Precision, and Recall. $w = -1$ represents the collapsed model (θ_s), while $w = 0$ denotes the base model (θ_r). NEON ($w > 0$) improves FID primarily by restoring recall at the expense of precision. Source: [1].

3 Contributions and Innovations

While the NEON framework successfully demonstrated the efficacy of negative extrapolation, it inherently treats the model as a monolithic entity, applying a uniform scalar w across all parameters simultaneously. However, modern generative architectures, such as U-Nets, are hierarchical by design and composed of distinct functional components that process information at varying levels of abstraction.

We hypothesize that model collapse does not affect all network layers uniformly. Rather, the magnitude and direction of the collapse vector likely vary across different functional blocks. Consequently, a global extrapolation weight w is likely suboptimal - potentially being too aggressive for some layers while remaining too conservative for others.

To address this limitation, we propose Block-Wise Adaptive NEON, a novel refinement that decomposes the model into functional blocks and optimizes the extrapolation weight independently for each. We validate this approach through a two-stage experimental framework.

First, using a Toy Problem (6-layer MLP trained on a 2D Gaussian distribution), we analyze the sensitivity of input, hidden, and output layers to collapse, demonstrating that layer-specific weights yield superior restoration of the target distribution.

Second, we extend our hypothesis to a real-world application using the Elucidated Diffusion Model (EDM) on CIFAR-10. We partition the U-Net architecture into three functional blocks based on feature resolution: the Encoder Block, which processes high-resolution features during the downsampling path to compress the input into abstract representations; the Middle Block, which operates at the minimal bottleneck resolution to capture the most condensed semantic information; and the Decoder Block, which handles the upsampling path to reconstruct fine-grained details and ensure pixel-level fidelity.

4 Results

4.1 Toy Problem

In this preliminary experiment, we trained a simple diffusion model to reconstruct a 2D Gaussian distribution, aiming to validate our hypothesis on a controlled environment before scaling to high-dimensional image data.

First, we reproduced the global optimization results reported in the NEON paper. Our implementation yielded an optimal global extrapolation weight of $w \approx 0.28$, achieving a baseline FID score of 0.0355. This served as the benchmark for evaluating our layer-wise approach.

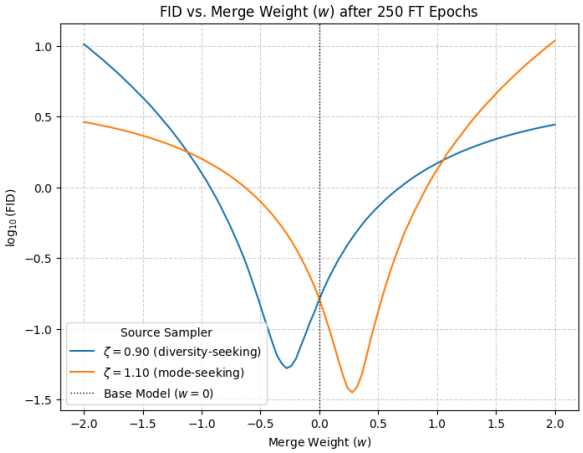
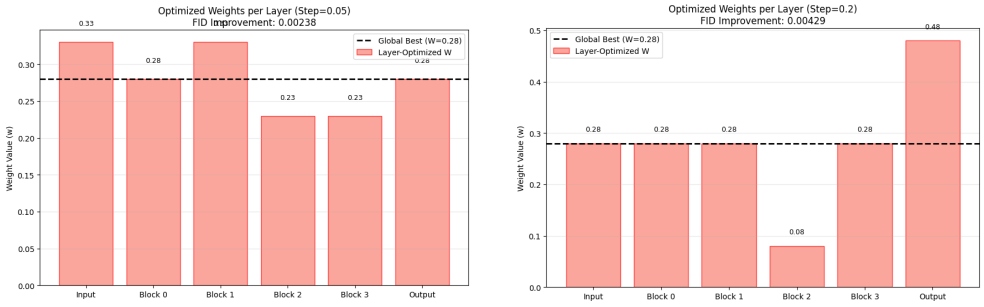


Figure 2: **Baseline Reproduction (Toy Problem).** Reconstruction of the 2D Gaussian distribution using the global NEON method, yielding an optimal scalar weight of $w \approx 0.28$ (FID: 0.0355).

To investigate layer-wise sensitivity, we performed a systematic grid search around it. We evaluated a comprehensive range of step sizes $\delta \in \{0.05, 0.1, 0.2, 0.4, 0.6, 1.0, 1.5, 1.72\}$. For each δ, the network layers were independently assigned weights from the set $\{w^* - \delta, \quad w^*, \quad w^* + \delta\}$, creating a 3^6 search space per step.

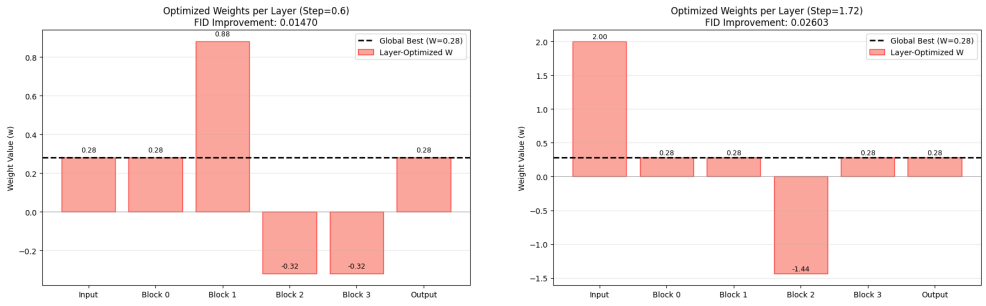
Local Refinement ($\delta \leq 0.2$): For smaller step sizes, the optimization acted as a fine-tuning mechanism. As visualized in the following figure (showing representative steps $\delta = 0.05$ and $\delta = 0.2$), allowing layers to deviate slightly from the global mean yielded consistent, albeit moderate, improvements. Specifically, $\delta = 0.2$ reduced the FID to 0.03126 (approx. 12% improvement).



Global Optimization with Positive Extrapolation ($\delta \geq 0.6$):

As we increased the step size δ, the search space expanded significantly to include *positive extrapolation* values (where $w_{layer} < 0$). This allowed the model to adopt a hybrid strategy: pushing some layers away from the collapsed model (negative extrapolation) while pulling others back towards it (positive extrapolation).

The results were striking. The configuration found at $\delta = 1.72$ achieved the global minimum with an FID of 0.00952 - a dramatic 73% improvement over the baseline.



This variety of results across different step sizes demonstrates that there is no single, uniform direction for optimal layer-wise correction. Instead, we observe that the network's functional components exhibit varying degrees of sensitivity to collapse. By allowing each layer to deviate independently from the global baseline, the model can adopt complex, heterogeneous weight configurations that significantly outperform any monolithic scalar weight. These findings suggest that layer-specific adjustment is essential for capturing the nuanced restoration needs of different architectural blocks.

4.2 EDM on CIFAR-10

For our primary experiment, we evaluated the proposed Block-Wise approach on the Elucidated Diffusion Model (EDM) trained on CIFAR-10. Leveraging the inherent structure of the U-Net architecture, we partitioned the model's parameters into three distinct functional groups: the Encoder Block (processing 32×32 and 16×16 resolutions), the Middle Block (8×8 semantic bottleneck), and the Decoder Block (reconstructing back through 16×16 and 32×32 resolutions).

While our attempts to exactly replicate the original paper's results yielded a slightly different baseline (optimal $w = 1.0$ with an FID of approximately 1.43, slightly higher than the 1.38 reported in the original paper due to hardware-specific stochasticity), this value served as our reference point. Consequently, we defined our grid search space around this local optimum, testing weight triplets $(w_{\text{enc}}, w_{\text{mid}}, w_{\text{dec}})$ from the set $\{0.75, 1.0, 1.25\}$. The search was performed across multiple training snapshots (from 550k to 2552k images) to analyze how the optimal layer-wise correction evolves as model drift progresses.

Snapshot (K)	w^* **Original NEON**	w^* **Block-Wise (E | D | M)**
550	0.75	E-1.25 | D-0.75 | M-1.0
1051	0.75	E-1.0 | D-0.75 | M-1.25
1551	1.0	E-1.25 | D-0.75 | M-1.25
2051	1.25	E-1.25 | D-1.0 | M-1.0
2552	1.0	E-1.0 | D-1.0 | M-1.0

Table 1: **Optimal Weight Configurations.** Comparison of the best extrapolation weights found for the global baseline (NEON) versus our Block-Wise approach across different self-training snapshots.

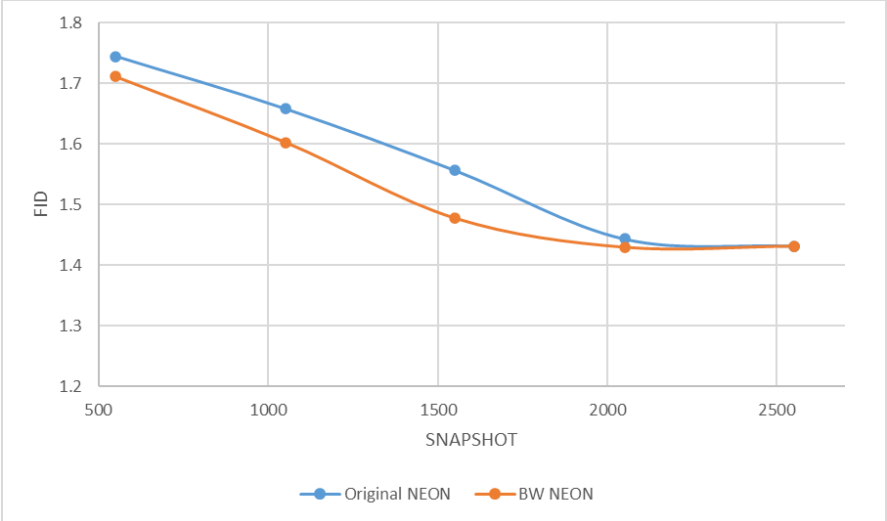


Figure 3: FID Performance Across Training Snapshots. The graph compares the Global NEON baseline with our Block-Wise NEON approach. Across all snapshots, representing increasing levels of model drift, our method consistently achieves a lower FID.

The experimental results validate the core hypothesis of this work: different architectural components drift at different rates during self-training. By decoupling the weights, the Block-Wise approach consistently achieved lower FID scores across the evaluated snapshots. Although the numerical improvement over the global baseline is relatively modest, the consistency of these results across multiple training points suggests that layer-specific adjustment provides a more precise correction mechanism.

A key observation from the grid search is the non-uniformity of the weights required to reach these minima. This confirms that the semantic core of the model and the pixel-level reconstruction layers drift in distinct ways. Even slight adjustments tailored to these functional roles can lead to measurable performance gains, suggesting that architectural awareness is essential when correcting for model collapse.

5 Conclusions and Future Directions

This work validates the hypothesis that model collapse is a structural and reversible phenomenon, while demonstrating that its dynamics are not uniform across the network architecture. By extending the NEON framework to a block-wise approach, we showed that treating neural networks as hierarchical structures allows for a more nuanced and effective mitigation of parameter drift.

Our experiments across both toy distributions and high-dimensional image data confirm that different functional components, possess distinct restoration needs.

Future research could employ automated optimization, such as gradient-based search, to dynamically determine layer-wise weights. While our grid search provided a clear proof of concept, exhaustive exploration of the weight space remains a scaling challenge requiring extensive computational resources. Extending this approach to larger architectures, such as Latent Diffusion Models, offers a promising path for analyzing drift in systems with more distinct semantic and visual components.

References

[1] Sina Alemohammad, Zhangyang Wang, and Richard G Baraniuk. Neon: Negative extrapolation from self-training improves image generation. *arXiv preprint arXiv:2510.03597*, 2025.

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

[3] Sina Alemohammad, Josue Casco-Rodriguez, Lorenzo Luzi, Ahmed Imtiaz Humayun, Hossein Babaei, Daniel LeJeune, Ali Siahkoohi, and Richard Baraniuk. Self-consuming generative models go mad. In *The Twelfth International Conference on Learning Representations*, 2023.