Block Diffusion: Interpolating Between Autoregressive and Diffusion Language Models[1]

Reproduction Results and Two Practical Extensions

Nikolaos Kordas[1] **Konstantinos Kritharidis**[1] **Ilias Makras**[1]
Georgios Markoulidakis[1] **Georgios Ntountounakis**[1] **Petros Vitalis**[1]

Pattern Recognition Course, National Technical University of Athens

Scientific Partner: **Efthymios Georgiou**[3]
Course Instructor: **Alexandros Potamianos**[1,2]

[1] National Technical University of Athens
[2] Archimedes RU, Athena RC
[3] University of Bern

Abstract

In this report, we are evaluating the Block Discrete Denoising Diffusion Language Models (BD3-LMs), a class of models designed to combine the advantages of Autoregressive (AR) and Discrete Diffusion models. BD3-LMs maintain an autoregressive distribution over blocks while performing parallelized discrete diffusion within each block. We present a reproduction of the original experiments, as well as our own ideas for extensions that improve the performance of BD3-LMs. Specifically, we explore two new noise schedules and loss reweighting. Our reproduction confirms that BD3-LMs outperform standard diffusion models, achieving variable-length generation and high quality sampling efficiency, even under significant resource constraints. Our proposed extensions further enhance the performance of BD3-LMs, demonstrating the potential of this class of models for efficient language generation.

Resources: Original codebase: `https://github.com/kuleshov-group/bd3lms`. Our repository: `https://github.com/ntua-el21050/bd3lms`.

1 Introduction

Modern language models are typically built using one of two generation paradigms. *Autoregressive* (AR) models generate one token at a time, which empirically yields strong likelihoods and high sample quality, supports key-value (KV) caching, and naturally allows variable-length generation. *Diffusion* models instead aim to generate (or refine) many tokens in parallel through iterative denoising steps, which can improve controllability and parallelism, but often suffers from a perplexity gap and is limited to fixed-length outputs.

Block diffusion (BD3-LMs) is a hybrid: it is autoregressive over blocks and performs diffusion within each block. This design targets a controllable trade-off between quality and parallelism via the block size L'. At the extremes, $L' = 1$ reduces to token-level AR, while $L' = L$ (the full sequence length) becomes a fully parallel diffusion model.

2 Block Diffusion Models (BD3-LMs)

BD3-LMs factorize a sequence into contiguous blocks of length L'. Generation proceeds autoregressively at the block level: each block is generated conditioned on the previous blocks. Within the current block, denoising is performed using a masked diffusion process, enabling parallel prediction of the tokens in the block.

This hybridization yields:

- **Parallelism within blocks:** enables faster refinement than strictly token-by-token generation.
- **Autoregressive control across blocks:** enables variable-length generation and mitigates some diffusion limitations.
- **A single knob L':** a direct quality–parallelism trade-off.

3 Background: Masked Discrete Diffusion and Noise Schedules

We focus on masked discrete diffusion, where noising corresponds to masking tokens with probability $p(t)$ as a function of a continuous time index $t \sim \mathcal{U}[0, 1]$. t essentially corresponds to the noise level to be applied. In general, larger values of t are associated with a higher proportion of masking, whereas smaller values of t correspond to a lower proportion of masking. However, the exact masking probability $p(t)$, is determined by the specific noise schedule used. Accordingly, the complementary keep (no-mask) probability is defined as:

$$a(t) = 1 - p(t). \tag{1}$$

Another quantity derived as a function of t is the loss scaling:

$$\text{loss\_scaling}(t) = \frac{a'(t)}{1 - a(t)} = -\frac{p'(t)}{p(t)}, \tag{2}$$

which acts as a weighting of per-token log-likelihood terms across noise levels during training.

We consider standard schedules (loglinear, square, square root, logarithmic, cosine) and also investigate clipped schedules, where t is sampled from a restricted interval $\mathcal{U}[\tau_{\min}, \tau_{\max}]$.

4 Experimental Protocol and Metrics

To evaluate the efficacy of Block Diffusion models (BD3-LMs) against Autoregressive (AR) and standard Diffusion baselines (SEDD, MDLM), we adopted a rigorous experimental protocol tailored to resource-constrained environments.

4.1 Model Configuration and Datasets

Due to computational limitations, we utilized a **tiny** transformer configuration for all models to ensure a fair comparison under identical resource budgets.

- **Sequence Length:** A key feature of BD3-LMs is variable-length generation. We configured the BD3-LM with a maximum model length of **16K tokens** to test generation capabilities beyond the standard fixed-length caps (e.g., 1024 tokens) of baseline diffusion models.
- **Datasets:** We primarily utilized **LM1B** (One Billion Word Benchmark) and **OpenWebText (OWT)** for training and perplexity evaluations. For zero-shot transfer evaluation, models trained on OWT were assessed on **Lambada** and **Wikitext**.

4.2 Training Regimen

Our training protocol consists of distinct pretraining and fine-tuning phases. The specific step counts varied by experiment to balance convergence with compute availability:

- **Noise Schedule Ablations (LM1B):** 400 pretraining steps followed by 100 fine-tuning steps.
- **Large-Scale Comparisons (OWT):** Extended training of 3000 pretraining steps and 3000 fine-tuning steps.
- **Transfer & Sampling:** 800 pretraining steps followed by 500–800 fine-tuning steps.

4.3 Evaluation Metrics

We employed a comprehensive suite of metrics to evaluate likelihood, stability, and sample quality:

- **Test Perplexity (PPL):** We report test set perplexity, where lower values indicate better predictive performance.
- **Objective Stability (Var. NELBO):** To quantify training stability, particularly regarding noise schedules, we measure the Variance of the Negative Evidence Lower Bound (lower is better).
- **Sampling Quality (Gen.PPL):** We evaluate the quality of generated text using Generative Perplexity calculated on 300 samples per model.
- **Efficiency (NFEs):** We track the Number of Function Evaluations required for generation to measure sampling efficiency.

5 Reproduction Results

Below, we reproduce all tables from the original paper.

5.1 AR vs BD3-LM with $L' = 1$

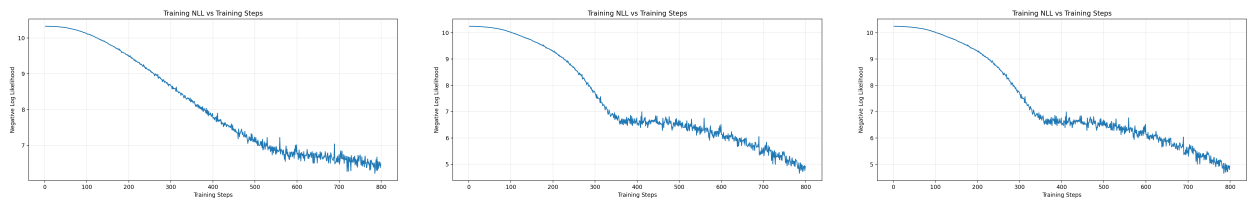
The first experiment we reproduced, was the comparison between AR and BD3-LM models with block size equal to 1. As the authors of the original paper explain, the expected value of the variance NELBO of the AR and the BD3-LM with $L' = 1$ are equal. What the authors found though, was that in practice the AR achieves better perplexity score than the BD3-LM. They explained that this was due to higher training variance in the BD3-LM's case. Using their default Log-linear noise schedule, meant that only half of the tokens would be masked. As a result the training cross-entropy would not be computed for half the tokens. They managed to overcome this, by setting the mask probability equal to 1, using a tuned schedule where $t \sim \mathcal{U}[1, 1]$.

Table 1 compares token-level AR and BD3-LM with $L' = 1$ on LM1B.

Table 1: Test perplexities for single-token generation on LM1B (800 training steps).

	PPL (↓)
Autoregressive (AR)	**1893**
BD3-LM ($L' = 1$)	2231
BD3-LM ($L' = 1$) + tuned schedule	2220

We also include the training curves used in the presentation (Figure 1) to visually compare training nll behavior.



(a) AR (b) BD3-LM (c) BD3-LM + tuned schedule

Figure 1: Training NLL curves for single-token generation on LM1B.

We indeed verify that the AR model scores better PPL than the BD3-LM model, but we fail to reproduce the same improvement when using the tuned schedule. The perplexity gap is not closed in this case, although it is slightly narrowed. With the help of the training nll curves, we observe that the BD3-LM models clearly suffer more from overfitting. Consequently, in our case the problem is not the higher variance, but the overfitting which is obviously caused by our limited resources. This of course becomes apparent if we compare our ppl scores with the ones of the original paper, where the AR scores 22.88 and the BD3-LM scores 25.56.

5.2 Effect of clipped noise schedules

Table 2 reports PPL and Var. NELBO for several L' values under clipped and unclipped uniform time sampling on LM1B.

Table 2: Effect of clipped noise schedules on LM1B (400 pretraining steps + 100 fine-tuning steps).

L'	Clipping	PPL (↓)	Var. NELBO (↓)
128	$\mathcal{U}[0, 0.5]$	**2106**	**1.27**
	$\mathcal{U}[0, 1]$	**2106**	**1.27**
16	$\mathcal{U}[0.3, 0.8]$	**1278**	**10.50**
	$\mathcal{U}[0, 1]$	1279	10.51
4	$\mathcal{U}[0.5, 1]$	**1226**	**44.41**
	$\mathcal{U}[0, 1]$	**1226**	**44.41**

Under our reproduced settings, clipping is most clearly beneficial for $L' = 16$ (small but consistent improvements in both PPL and variance), whereas for $L' = 4$ and $L' = 128$ the effect is negligible.

5.3 LM1B and OWT comparisons

Tables 3 and 4 compare AR, diffusion baselines, and BD3-LM variants.

Table 3: Test perplexities on LM1B (400 pretraining steps + 100 fine-tuning steps).

Model	PPL (↓)
Autoregressive	
Transformer	3042
Diffusion	
SEDD	1447
MDLM	1616
Block diffusion	
BD3-LM $L' = 16$	1278
BD3-LM $L' = 8$	1734
BD3-LM $L' = 4$	**1226**

Table 4: Test perplexities on OWT (3000 pretraining steps + 3000 fine-tuning steps).

Model	PPL (↓)
Autoregressive	
Transformer	2036
Diffusion	
SEDD	2120
MDLM	2101
Block diffusion	
BD3-LM $L' = 16$	1939
BD3-LM $L' = 8$	1941
BD3-LM $L' = 4$	**1935**

On LM1B, BD3-LMs outperform the diffusion baselines under the same training budget. On OWT, BD3-LMs are competitive and slightly improve over the AR baseline in our reproduced setting.

5.4 Transfer evaluation (trained on OWT)

Table 5 summarizes validation perplexities on other datasets for models trained on OWT.

Table 5: Zero-shot validation perplexities of models trained on OWT (800 pretraining steps + 800 fine-tuning steps).

	LM1B	Lambada	Wikitext
AR	2388	1550	**2875**
SEDD	2742	1562	3335
MDLM	2722	1556	3283
BD3-LM $L' = 4$	**2196**	**1438**	3143

5.5 Variable-length generation

Table 6 highlights a core benefit of block diffusion over pure diffusion: longer generations are possible because blocks are generated autoregressively.

Table 6: Generation length statistics for 10 sampled documents from OWT-trained models (800 pretraining steps + 500 fine-tuning steps). BD3-LM reproduction with model length = 16K.

	Median # tokens	Max # tokens
OWT train set	717	131K
AR	16384	16384
SEDD	1000	1000
BD3-LM $L' = 16$	798	2927

We observe that, indeed, BD3-LMs overcome the fixed-length generation limitation of diffusion models, creating outputs that can go well-beyond the context length of their training (1024). However, still the AR model achieves far longer generation, as happens in the original papers as well. The fact that the median number of tokens in the AR model is equal to the maximum number of tokens, is explained by the fact that with only 800 training steps, the AR model doesn't have the time to properly train, and as a result, doesn't learn when to stop generating outputs. This is in accordance to our empirical evaluation of the quality of the generation results, which was poor for the AR model but far better for the BD3-LM (with 800 pretraining and 500 fine-tuning steps).

5.6 Sample quality and schedule ablations

We report (i) sampling quality in terms of Gen.PPL and NFEs (Table 7) and (ii) an ablation over noise schedules (Table 8).

Specifically, in discrete diffusion the NFEs (number of generation steps) is upper bounded by the context length L. Thus, our reproduction resembles the paper's experiments. On the other hand, we observe that BD3-LMs achieve better PPL scores faster than their AR counterpart.

Table 7: Generative perplexity (Gen.PPL; ↓) and number of function evaluations (NFEs; ↓) for 300 samples. Models trained on OWT (400 + 100 training steps).

Model	Gen.PPL	NFEs
AR	79165	1024
SEDD	29987	1023
MDLM	25632	1023
BD3-LM $L' = 16$	**7576**	1023
BD3-LM $L' = 8$	8176	1023
BD3-LM $L' = 4$	9785	1023

The ablation supports the presentation takeaway: clipped schedules reliably reduce variance, but standard schedules can yield lower perplexity given enough training.

Table 8: Effect of noise schedule on PPL and Var. NELBO for different L' on LM1B (5000 + 3000 training steps).

Noise schedule	PPL (↓)	Var. NELBO (↓)
$L' = 4$		
Clipped $\mathcal{U}[0.45, 0.95]$	1199	**15.23**
Clipped $\mathcal{U}[0.3, 0.8]$	1101	27.61
Linear $\mathcal{U}[0, 1]$	751	260.10
Logarithmic	750	125.62
Square root	**719**	83.83
$L' = 16$		
Clipped $\mathcal{U}[0.45, 0.95]$	1056	**4.35**
Clipped $\mathcal{U}[0.3, 0.8]$	798	6.41
Linear $\mathcal{U}[0, 1]$	662	44.73
Square	**627**	27.53
Cosine	634	20.80

6 Extension 1: Alternative Noise Schedules

6.1 Already Implemented Noise Schedules

Table 9 summarizes the noise schedules already provided in the codebase, expressed in terms of the masked probability $p(t)$ and the induced loss scaling loss_scaling$(t) = -\frac{p'(t)}{p(t)}$.

Table 9: Already implemented noise schedules: masked probability $p(t)$ and induced loss scaling.

Schedule	$p(t)$	**loss_scaling**(t)
LogLinear	t	$-\dfrac{1}{t}$
Square	t^2	$-\dfrac{2}{t}$
Square root	$t^{0.5}$	$-\dfrac{1}{2t}$
Logarithmic	$\dfrac{\log(1+t)}{\log 2}$	$-\dfrac{1}{(1+t)\,\log(1+t)}$
Cosine	$1-(1-\varepsilon)\cos\left(\dfrac{\pi t}{2}\right)$	$-\dfrac{\left(\frac{\pi}{2}\right)(1-\varepsilon)\sin\left(\frac{\pi t}{2}\right)}{1-(1-\varepsilon)\cos\left(\frac{\pi t}{2}\right)}$

6.2 Gaussian & Bimodal Gaussian Noise Schedules

We propose schedules that directly sample the masked probability $p(t) \in (0, 1)$. For a (truncated) Gaussian schedule, given mean μ and standard deviation σ, define $\alpha = \frac{0-\mu}{\sigma}$, $\beta = \frac{1-\mu}{\sigma}$, Φ the standard normal CDF, and φ the pdf. Let $\Phi_\alpha = \Phi(\alpha)$, $\Phi_\beta = \Phi(\beta)$, and $Z = \Phi_\beta - \Phi_\alpha$. Then for $t \in (0, 1)$,

$$z(t) = \Phi^{-1}\big(\Phi_\alpha + t(\Phi_\beta - \Phi_\alpha)\big), \qquad p(t) = \mu + \sigma z(t) \in (0, 1). \tag{3}$$

The induced scaling is

$$\text{loss\_scaling}(t) = -\frac{p'(t)}{p(t)} = -\frac{\sigma Z}{\varphi(z(t))\,p(t)}. \tag{4}$$

For a bimodal Gaussian schedule, we choose a mixture weight $w \in (0, 1)$ and two Gaussians (μ_1, σ_1) and (μ_2, σ_2). With probability w we use the first component; otherwise the second. Writing $w_1 = w$ and $w_2 = 1 - w$, we define a split time coordinate

$$t_1 = \frac{t}{w_1} \ (t < w_1), \qquad t_2 = \frac{t - w_1}{w_2} \ (t \geq w_1), \tag{5}$$

and sample

$$p(t) = \begin{cases} \mu_1 + \sigma_1 z_1(t_1), & t < w_1, \\ \mu_2 + \sigma_2 z_2(t_2), & t \geq w_1, \end{cases} \tag{6}$$

where each $z_i(\cdot)$ is defined as in the Gaussian case but with its own truncation constants $\alpha_i = \frac{0-\mu_i}{\sigma_i}$, $\beta_i = \frac{1-\mu_i}{\sigma_i}$ and $Z_i = \Phi_{\beta_i} - \Phi_{\alpha_i}$. The induced scaling is also piecewise:

$$\text{loss\_scaling}(t) = -\frac{1}{p(t)} \begin{cases} \frac{1}{w_1} \frac{\sigma_1 Z_1}{\varphi(z_1(t_1))}, & t < w_1, \\ \frac{1}{w_2} \frac{\sigma_2 Z_2}{\varphi(z_2(t_2))}, & t \geq w_1. \end{cases} \tag{7}$$

6.3 Results

Table 10 compares the new schedules to existing ones under the same protocol.

Table 10: Already-implemented schedules vs newly implemented Gaussian and bimodal Gaussian (B.G.) schedules on LM1B (400 pretraining steps + 100 fine-tuning steps).

Block size	**Already implemented**			**Newly implemented**		
	Noise schedule	PPL	Var. NELBO	Noise schedule	PPL	Var. NELBO
128	Loglinear	2106	1.27	Gaussian ($\mu = 0.5$)	2115	1.29
	Loglinear $\mathcal{U}[0, 0.5]$	2106	1.27	Gaussian ($\mu = 0.6$)	2212	1.34
	Cosine	2154	1.31	B.G. ($\mu_1 = 0.3, w_1 = 0.6$)	2184	1.31
	Cosine $\mathcal{U}[0, 0.5]$	2150	1.30	B.G. ($\mu_1 = 0.1, w_1 = 0.6$)	**2089**	**1.19**
16	Loglinear	1279	10.50	Gaussian ($\mu = 0.5$)	**1234**	**10.28**
	Loglinear $\mathcal{U}[0.3, 0.8]$	1278	10.51	Gaussian ($\mu = 0.6$)	1235	10.31
	Cosine	1236	10.30	B.G. ($\mu_1 = 0.3, w_1 = 0.6$)	1254	10.42
	Cosine $\mathcal{U}[0.3, 0.8]$	1235	10.29	B.G. ($\mu_1 = 0.1, w_1 = 0.6$)	1295	10.59
4	Loglinear	**1226**	**44.41**	Gaussian ($\mu = 0.5$)	1250	46.76
	Loglinear $\mathcal{U}[0.5, 1]$	**1226**	**44.41**	Gaussian ($\mu = 0.7$)	1252	46.76
	Cosine	1228	45.28	B.G. ($\mu_1 = 0.3, w_1 = 0.6$)	1253	46.84
	Cosine $\mathcal{U}[0.5, 1]$	1229	45.26	B.G. ($\mu_1 = 0.1, w_1 = 0.6$)	1243	46.49

We also tested using the bimodal Gaussian schedule during pretraining; Table 11 reports the best fine-tuning schedule found per pretraining schedule.

Table 11: Applying bimodal Gaussian during pretraining on LM1B: for each pretraining schedule we report the best fine-tuning schedule found (400 pretraining steps + 100 fine-tuning steps).

Block size	Pretraining schedule	Fine-tuning schedule	PPL	Var. NELBO
128	Loglinear	B.G. ($\mu_1 = 0.1, w_1 = 0.6$)	2089	1.25
	Bimodal Gaussian	B.G. ($\mu_1 = 0.3, w_1 = 0.6$)	**2070**	**1.19**
16	Loglinear	Gaussian ($\mu = 0.5$)	1234	10.28
	Bimodal Gaussian	Cosine $\mathcal{U}[0.3, 0.8]$	**1227**	**10.22**
4	Loglinear	Loglinear	1226	44.41
	Bimodal Gaussian	Loglinear	**1213**	**44.39**

7 Extension 2: Loss Reweighting for the Masked Diffusion Objective

A second direction is to reweight the masked diffusion NELBO. Starting from a weighted objective

$$\mathcal{L}^{\tilde{w}}(\mathbf{x}) = -\int_0^1 \tilde{w}(t) \frac{\alpha_t'}{1-\alpha_t} \mathbb{E}_{q(\mathbf{z}_t|\mathbf{x})} \left[\delta_{\mathbf{z}_t,m} \cdot \mathbf{x}^\top \log \mu_\theta(\mathbf{z}_t)\right] \mathrm{d}t, \tag{8}$$

we can reparameterize via $\lambda(t) = \log \frac{\alpha_t}{1-\alpha_t}$ and express weighting in the λ domain. We evaluated multiple weightings migrated from continuous-time diffusion and a simple baseline weighting.

Table 12 lists the investigated weighting functions.

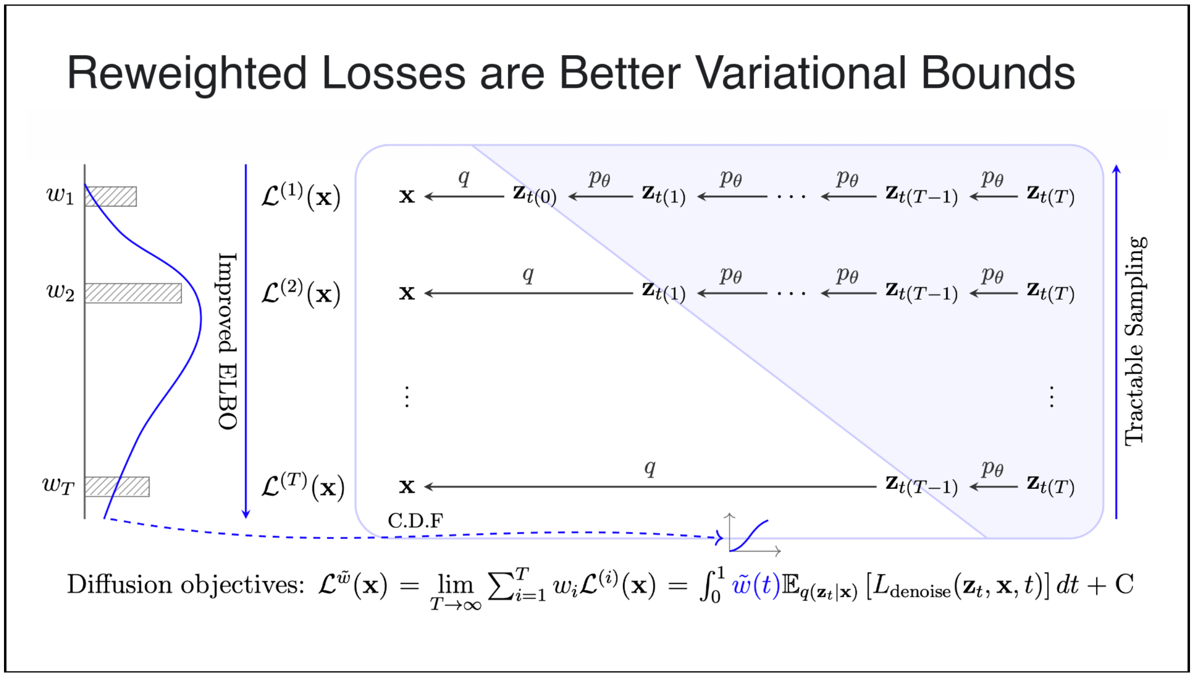

Reweighted Losses are Better Variational Bounds
w_1
w_2
w_T
Improved ELBO
$\mathcal{L}^{(1)}(\mathbf{x})$
$\mathcal{L}^{(2)}(\mathbf{x})$
$\mathcal{L}^{(T)}(\mathbf{x})$
$\mathbf{x} \xleftarrow{q} \mathbf{z}_{t(0)} \xleftarrow{p_\theta} \mathbf{z}_{t(1)} \xleftarrow{p_\theta} \cdots \xleftarrow{p_\theta} \mathbf{z}_{t(T-1)} \xleftarrow{p_\theta} \mathbf{z}_{t(T)}$
$\mathbf{x} \xleftarrow{q} \mathbf{z}_{t(1)} \xleftarrow{p_\theta} \cdots \xleftarrow{p_\theta} \mathbf{z}_{t(T-1)} \xleftarrow{p_\theta} \mathbf{z}_{t(T)}$
$\mathbf{x} \xleftarrow{q} \mathbf{z}_{t(T-1)} \xleftarrow{p_\theta} \mathbf{z}_{t(T)}$
C.D.F
Tractable Sampling
Diffusion objectives: $\mathcal{L}^{\tilde{w}}(\mathbf{x}) = \lim_{T\to\infty} \sum_{i=1}^{T} w_i \mathcal{L}^{(i)}(\mathbf{x}) = \int_0^1 \tilde{w}(t) \mathbb{E}_{q(\mathbf{z}_t|\mathbf{x})}\left[L_{\text{denoise}}(\mathbf{z}_t, \mathbf{x}, t)\right] dt + \mathrm{C}$

Figure 2: Illustration of reweighting schemes (from the final presentation).

Table 12: Weighting functions investigated for masked diffusion models (as reported in the final presentation).

Name	$\lambda(t)$	$\hat{w}(\lambda)$	$\tilde{w}(t)$
EDM		$p_{\mathcal{N}(2.4,2.4^2)}(\lambda)\frac{e^{-\lambda}+0.5^2}{0.5^2}$	$w(\lambda(t))$
IDDPM		$\operatorname{sech}(\frac{\lambda}{2})$	$2\sqrt{\alpha_t(1-\alpha_t)}$
Sigmoid	$\log\frac{\alpha_t}{1-\alpha_t}$	$\operatorname{sigmoid}(-\lambda+k)$	$\frac{1-\alpha_t}{1-(1-e^{-k})\alpha_t}$
FM		$e^{-\frac{\lambda}{2}}$	$\sqrt{\frac{1-\alpha_t}{\alpha_t}}$
Simple		–	$-\frac{1-\alpha_t}{\alpha_t'}$

Table 13: Test perplexities under different loss reweightings (from the final presentation).

	PPL (↓)					
	Base	IDDPM	EDM	Sigmoid ($k=0$)	FM	Simple
Autoregressive						
Transformer	1221					
Diffusion						
SEDD	1403					
MDLM	1370					
Block diffusion						
BD3-LM $L'=16$	1150.15	8048.88	1593.4	35.02	76213	53070
BD3-LM $L'=8$	1078.57	7954.17	1569.75	34.82	109169	3.6e10
BD3-LM $L'=4$	1093.7	7857.33	1565.6	**34.08**	67332	2.4e6

Finally, Table 13 reports the test perplexities for the different reweightings.

In these runs, sigmoid reweighting yields the best PPL for $L'=8$, but several weightings can become unstable (very large perplexities), highlighting the need for careful constraints (e.g., monotonicity conditions with specific schedules).

8 Discussion

Across reproductions and extensions we consistently observe:

- **Smaller blocks help perplexity:** decreasing L' often improves PPL, consistent with approaching the AR regime.
- **Variance–PPL trade-off:** clipped schedules reduce Var. NELBO, but the best perplexity can come from standard schedules after sufficient optimization.
- **Length behavior differs qualitatively:** BD3-LMs can exceed fixed-length diffusion caps because generation proceeds block-by-block.

9 Conclusion and Future Work

Our reproduction supports the main motivation for BD3-LMs: they interpolate between AR and diffusion and, under comparable budgets, can achieve strong perplexity and improved sampling quality relative to diffusion baselines, while retaining variable-length generation. We also found that (i) bimodal Gaussian schedules can improve PPL/variance in specific settings and (ii) loss reweighting (notably sigmoid weighting in our runs) can yield additional improvements but may be unstable.

Promising next steps include frequency-informed masking (masking rare, information-rich tokens more often), combining both extensions in a single training run, and scaling compute to reduce small-model noise.

References

[1] Marianne Arriola et al. "Block Diffusion: Interpolating Between Autoregressive and Diffusion Language Models". In: *The Thirteenth International Conference on Learning Representations*. 2025.

[2] Despoina Kosmopoulou et al. "Masked Diffusion Language Models with Frequency-Informed Training". In: *Proceedings of the First BabyLM Workshop*. Ed. by Lucas Charpentier et al. Suzhou, China: Association for Computational Linguistics, Nov. 2025, pp. 531–539. DOI: `10.18653/v1/2025.babylm-main.38`.

[3] Jiaxin Shi and Michalis K. Titsias. *Demystifying Diffusion Objectives: Reweighted Losses are Better Variational Bounds*. 2025. arXiv: `2511.19664 [cs.LG]`.