

# Block Diffusion: Interpolating Between Autoregressive and Diffusion Language Models

Ntountounakis Georgios, Markoulidakis Georgios, Vitalis Petros,  
Makras Ilias, Kritharidis Konstantinos, Kordas Nikolaos

Pattern Recognition, ECE  
National Technical University of Athens

February 2026

# Table of Contents

## 1 Paper Overview

## 2 Our Results

- Reproduction
- Extensions
  - Alternative Noise Schedules
  - Loss Reweighting in the NELBO Objective

## 3 Conclusion & Future Work

## 4 Team Organization

## 5 Retrospection

# Table of Contents

## 1 Paper Overview

## 2 Our Results

- Reproduction
- Extensions
  - Alternative Noise Schedules
  - Loss Reweighting in the NELBO Objective

## 3 Conclusion & Future Work

## 4 Team Organization

## 5 Retrospection

# Introduction to the Problem-Motivation

**Two main approaches for Language Models:**

**Autoregressive (AR):**

- Token-by-token generation
- High quality
- KV caching
- Variable length

**Diffusion:**

- Parallel generation
- Better controllability
- **Fixed length (limitation)**
- **Lower quality (Perplexity Gap)**

**Question**

Can we combine the advantages of both approaches?

# Core Idea: Block Diffusion



...

Diffusion within each block(parallel)  
Autoregressive over blocks

**Parameterization:** Trade-off through block size  $L'$ :

- $L' = 1 \rightarrow$  Pure AR
- $L' = L \rightarrow$  Pure Diffusion

## Technical Contribution:

- Optimized training and sampling algorithms
- Introduced clipped noise schedules for reduced gradient variance during training
- SoTA PPL among diffusion models + Variable length generation capabilities

# Table of Contents

1 Paper Overview

2 Our Results

- Reproduction
- Extensions
  - Alternative Noise Schedules
  - Loss Reweighting in the NELBO Objective

3 Conclusion & Future Work

4 Team Organization

5 Retrospection

# Table of Contents

## 1 Paper Overview

## 2 Our Results

- Reproduction
- Extensions
  - Alternative Noise Schedules
  - Loss Reweighting in the NELBO Objective

## 3 Conclusion & Future Work

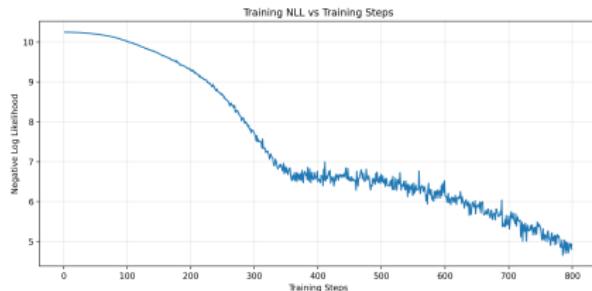
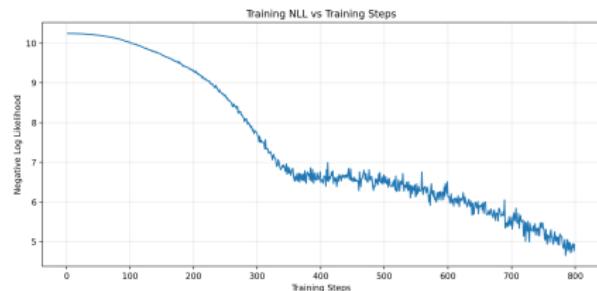
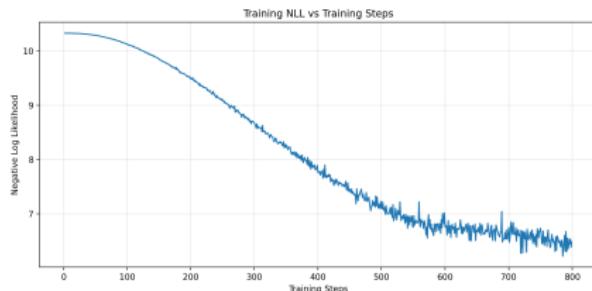
## 4 Team Organization

## 5 Retrospection

# AR vs BD3LM with $L'=1$

Test Perplexities for single token generation on LM1B dataset (800 Training Steps)

PPL ( $\downarrow$ )	
<b>Autoregressive</b>	<b>1893</b>
<b>BD3LM <math>L'=1</math></b>	2231
<b>BD3LM <math>L'=1</math> + Tuned Schedule</b>	2220



# The Effect of Clipped Noise Schedules

Test Perplexities for single token generation on LM1B dataset (400 Pretraining Steps + 100 Fine-tuning Steps).

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

## Original vs Reproduction

### Key takeaways:

- $L' = 128$  and  $L' = 4$ : Original clipping reduces Var. NELBO and PPL. Reproduction shows no significant changes.
- $L' = 16$ : Original clipped  $\mathcal{U}[0.3, 0.8]$  slightly improves PPL and clearly lowers variance vs  $\mathcal{U}[0, 1]$ ; Reproduction preserves the direction with smaller gaps.

Conclusion: the clipped-schedule advantage is strong in the original, but the gaps lower significantly in our reproduction (making the values identical in some cases), likely due to our small model/short run.

# BD3LMs vs ARs vs Diffusion Models on LM1B

Test perplexities (PPL; ↓) of models on LM1B (400 Pretraining Steps + 100 Fine-tuning Steps).

PPL (↓)	
<b>Autoregressive</b>	
Transformer	3042
<b>Diffusion</b>	
SEDD	1447
MDLM	1616
<b>Block diffusion (Ours)</b>	
BD3-LMs $L' = 16$	1278
$L' = 8$	1734
$L' = 4$	<b>1226</b>

## Original vs Reproduction

### Key takeaways:

- *AR performance gap:* Original Transformer beats diffusion/BD3LM (lowest PPL), while in reproduction Transformer is worst (highest PPL).
- *BD3LM trends partly preserved:* Both show smaller  $L'$  helps (best at  $L' = 4$ ), but original has all BD3LM variants beating diffusion, while reproduction has  $L' = 8$  worse than both diffusion baselines.

*Conclusion: the  $L'$  ranking within BD3LM is partly consistent, but cross-family comparisons (AR vs others, and diffusion vs BD3LM with  $L' = 8$ ) are not, likely from scaling/undertraining effects.*

# BD3LMs vs ARs vs Diffusion Models on OWT

Test perplexities (PPL; ↓) of models on OWT. (800 Pretraining Steps + 800 Fine-tuning Steps)

PPL (↓)	
<b>Autoregressive</b>	
Transformer	2036
<b>Diffusion</b>	
SEDD	2120
MDLM	2101
<b>Block diffusion (Ours)</b>	
BD3-LMs $L' = 16$	1939
$L' = 8$	1941
$L' = 4$	<b>1935</b>

## Original vs Reproduction

### Key takeaways:

- *AR vs diffusion preserved:* In both original and reproduction, AR outperforms diffusion baselines (lower PPL).
- *BD3LM vs AR flips:* Original ranks AR best overall (BD3LM worse than AR), while reproduction ranks BD3LM best (all BD3LM < AR).

*Conclusion: diffusion-vs-AR relationships match the paper, but the global winner changes (AR in original vs BD3LM in reproduction), possibly exaggerated at this small scale.*

# Performance on other Datasets

Zero-shot validation perplexities ( $\downarrow$ ) of models trained on OWT.(?training steps?)

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

## Original vs Reproduction

Key takeaways:

- *Wikitext*: Original and reproduction both rank **AR best**, but original has **BD3LM** beating diffusion, while reproduction has **BD3LM worse** than both diffusion baselines.
- *LM1B*: Original ranks **AR best** (BD3LM second), but reproduction ranks **BD3LM best**.
- *Lambada*: Original ranks **diffusion best** (followed by BD3LM), while reproduction ranks **BD3LM best**.

*Conclusion: Generally, the winner per dataset shifts between original and reproduction, consistent with small-model transfer being noisy.*

# Variable-Length Sequence Generation

Generation length statistics from sampling 10 documents from models trained on OWT (800 Pretraining Steps + 500 Fine-tuning Steps). BD3-LM reproduction with model length = 16K.

	Median # tokens	Max # tokens
OWT train set	717	131K
AR	4008	131K
SEDD	1021	1024
BD3-LM $L' = 16$	798	2927

## Original vs Reproduction

### Key takeaways:

- *Fixed vs variable-length preserved:* SEDD is capped at 1024 max, while BD3LM exceeds 1024.
- *Median ordering preserved:* Both original and reproduction keep the same ranking (AR > SEDD > BD3LM > OWT train).

# Sample Quality

Generative Perplexity (Gen.PPL;↓) and number of Function Evaluations (NFEs;↓) of 300 samples. All models are trained on WikiText2. (200 + 100 training steps)

Model	PPL	NFEs
AR	5	5
SEDD	5	5
MDLM	5	5
BD3-LM $L' = 16$	5	5
BD3-LM $L' = 8$	5	5
BD3-LM $L' = 4$	5	5

## Original vs Reproduction

### Key takeaways:

- *Variance Hypothesis*  
*Confirmed:* Clipped schedules reduced NELBO variance by 17x (from 260.1 to 15.2), validating the method's stability.
- *Perplexity trade-off:* Unlike the paper, standard noise schedules achieve better PPL. Those noise schedules are probably easier to be learned faster but lack on convergence.

*The method effectively stabilizes training (low variance) immediately, but PPL gains from this stability likely require a longer training horizon, especially with our tiny run.*

# Effect of Different Noise Schedules

Effect of noise schedule on PPL and variance of 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	<b>15.23</b>
$\mathcal{U}[0.3, 0.8]$	1101	27.61
Linear $\mathcal{U}[0, 1]$	751	260.10
Logarithmic	750	125.62
Square root	<b>719</b>	83.83
$L' = 16$		
Clipped		
$\mathcal{U}[0.45, 0.95]$	1056	<b>4.35</b>
$\mathcal{U}[0.3, 0.8]$	798	6.41
Linear $\mathcal{U}[0, 1]$	662	44.73
Square	<b>627</b>	27.53
Cosine	634	20.80

## Original vs Reproduction

### Key takeaways:

- *Variance Hypothesis*  
*Confirmed:* Clipped schedules reduced NELBO variance by 17x (from 260.1 to 15.2), validating the method's stability.
- *Perplexity trade-off:* Unlike the paper, standard noise schedules achieve better PPL. Those noise schedules are probably easier to be learned faster but lack on convergence.

*The method effectively stabilizes training (low variance) immediately, but PPL gains from this stability likely require a longer training horizon, especially with our tiny run.*

# Table of Contents

## 1 Paper Overview

## 2 Our Results

- Reproduction
- Extensions
  - Alternative Noise Schedules
  - Loss Reweighting in the NELBO Objective

## 3 Conclusion & Future Work

## 4 Team Organization

## 5 Retrospection

# Table of Contents

## 1 Paper Overview

## 2 Our Results

- Reproduction
- Extensions
  - Alternative Noise Schedules
  - Loss Reweighting in the NELBO Objective

## 3 Conclusion & Future Work

## 4 Team Organization

## 5 Retrospection

# Noise Scheduling in Masked Diffusion (Summary)

**Continuous time index:**

$$t \sim \mathcal{U}[0, 1]$$

**Noise schedule  $\Rightarrow$  masked probability:**

$p(t)$  : masked probability induced by the noise schedule

**Keep (no-mask) probability:**

$$a(t) = 1 - p(t)$$

**Loss scaling induced by the schedule:**

$$\text{loss scaling}(t) = \frac{a'(t)}{1 - a(t)} = -\frac{p'(t)}{p(t)}$$

## Intuition

The noise schedule sets *where* the model learns via  $p(t)$  (masking rate). When sampling  $t \sim \mathcal{U}[0, 1]$ ,  $\text{loss\_scaling}(t) = \frac{a'(t)}{1 - a(t)}$  acts as a weight on the per-token log-likelihood term, stabilizing the training across different noise levels.

# Already Implemented Noise Schedules

---

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

# Gaussian & Bimodal Gaussian Noise Schedules

**Goal:** sample a masked probability  $p(t) \in (0, 1)$  from  $t \sim \mathcal{U}[0, 1]$ .

**Gaussian schedule (truncated to  $(0, 1)$ ):**

Let  $\alpha = \frac{0-\mu}{\sigma}$ ,  $\beta = \frac{1-\mu}{\sigma}$ ,  $\Phi_\alpha = \Phi(\alpha)$ ,  $\Phi_\beta = \Phi(\beta)$ . For  $t \in (0, 1)$ :

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

With  $Z = \Phi_\beta - \Phi_\alpha$  and  $\varphi(\cdot)$  the standard normal pdf:

$$\text{loss\_scaling}(t) = \frac{a'(t)}{1 - a(t)} = -\frac{p'(t)}{p(t)} = -\frac{\sigma Z}{\varphi(z(t)) p(t)}.$$

**Bimodal Gaussian schedule (mixture):**

Choose a split weight  $w \in (0, 1)$  (denote  $w_1 = w$ ,  $w_2 = 1 - w$ ). With probability  $w$  use  $(\mu_1, \sigma_1)$ , otherwise use  $(\mu_2, \sigma_2)$ :

$$t_1 = \frac{t}{w_1} \quad (t < w_1), \quad t_2 = \frac{t - w_1}{w_2} \quad (t \geq w_1), \quad p(t) = \begin{cases} \mu_1 + \sigma_1 z_1(t_1), & t < w_1 \\ \mu_2 + \sigma_2 z_2(t_2), & t \geq w_1 \end{cases}$$

where each  $z_i(\cdot)$  is defined as above (with its own  $\alpha_i, \beta_i, \Phi_{\alpha_i}, \Phi_{\beta_i}$  and  $Z_i$ ). The resulting scaling is piecewise:

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

# New Schedules' Results

- Perplexities (PPL) and Var. NELBO for implemented vs new schedules on LM1B (400 Pretraining Steps + 100 Fine-tuning Steps).
- B.G. stands for Bimodal Gaussian.
- We use constant values:  $\sigma^2 = 0.1$  (Gaussian) and  $\sigma_1^2 = 0.02, \sigma_2^2 = 0.08$  (B.G.) for all experiments. Training under varying  $\mu$  and  $\mu_1, w_1$  respectively.

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. ( $\mu_1 = 0.1, w_1 = 0.6$ )	<b>2089</b>	<b>1.19</b>
16	Loglinear	1279	10.50	Gaussian ( $\mu = 0.5$ )	<b>1234</b>	<b>10.28</b>
	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	<b>1226</b>	<b>44.41</b>	Gaussian ( $\mu = 0.5$ )	1250	46.76
	Loglinear $\mathcal{U}[0.5, 1]$	<b>1226</b>	<b>44.41</b>	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

# Using Bimodal Gaussian on Pretraining

We can also apply the Bimodal Gaussian noise schedule during pretraining.

The table reports, for each pretraining noise schedule, the best fine-tuning noise schedule we found, with the corresponding PPL and Var. NELBO (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. ( $\mu_1 = 0.3, w_1 = 0.6$ )	<b>2070</b>	<b>1.19</b>
16	Loglinear	Gaussian ( $\mu = 0.5$ )	1234	10.28
	Bimodal Gaussian	Cosine $\mathcal{U}[0.3, 0.8]$	<b>1227</b>	<b>10.22</b>
4	Loglinear	Loglinear	1226	44.41
	Bimodal Gaussian	Loglinear	<b>1213</b>	<b>44.39</b>

# Table of Contents

## 1 Paper Overview

## 2 Our Results

- Reproduction
- Extensions
  - Alternative Noise Schedules
  - Loss Reweighting in the NELBO Objective

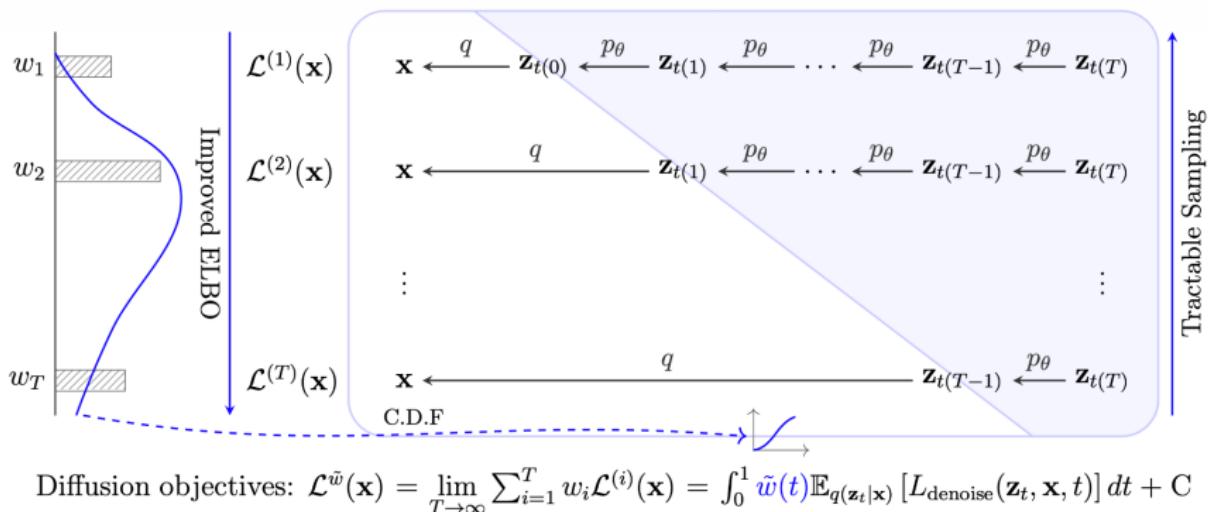
## 3 Conclusion & Future Work

## 4 Team Organization

## 5 Retrospection

# Reweighted Losses for Improved Diffusion Objective

## Reweighted Losses are Better Variational Bounds



# Reweighted Losses for Masked Diffusion

- Initial Reweighted NELBO:

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

- Reparameterization trick:  $\lambda(t) = \log \frac{\alpha_t}{1 - \alpha_t}$ :

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

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		$\text{sech}(\frac{\lambda}{2})$	$2\sqrt{\alpha_t(1 - \alpha_t)}$
Sigmoid	$\log \frac{\alpha_t}{1 - \alpha_t}$	$\text{sigmoid}(-\lambda + k)$	$\frac{1 - \alpha_t}{1 - (1 - e^{-k})\alpha_t}$
FM		$e^{-\frac{\lambda}{2}}$	$\sqrt{\frac{1 - \alpha_t}{\alpha_t}}$
Simple		-	$-\frac{1 - \alpha_t}{\alpha'_t}$

# Reweighted Losses Results

Extended Table 3: Test Perplexities

PPL ( $\downarrow$ )						
<b>Autoregressive</b>						
Transformer		1221				
<b>Diffusion</b>						
SEDD		1403				
MDLM		1370				
<b>Block diffusion</b>		Base	IDDPM	EDM	Sigmoid ( $k = 0$ )	FM
BD3-LMs $L' = 16$		1345	8048.88	1593.4	1150.15	76213
$L' = 8$		1210	7954.17	1569.75	<b>1078.57</b>	109169
$L' = 4$		1176	7857.33	1565.6	1093.7	67332
						2396260

# Table of Contents

## 1 Paper Overview

## 2 Our Results

- Reproduction
- Extensions
  - Alternative Noise Schedules
  - Loss Reweighting in the NELBO Objective

## 3 Conclusion & Future Work

## 4 Team Organization

## 5 Retrospection

# Conclusion

## Initial reproduction

In our initial reproduction, we observed noticeable gaps compared to the paper's reported results, largely due to insufficient training.

## Impact of our extensions

By identifying and applying our two extensions, we were able to improve performance.

## Impact of using more resources

With sufficient training compute, our results are consistent with the trends reported in the paper.

This table shows test perplexities of models on OWT (3000 Pretraining Steps + 3000 Fine-tuning Steps).

Model	PPL
AR	402
SEDD	575
MDLM	588
BD3LM ( $L' = 16$ )	438
BD3LM ( $L' = 8$ )	433
BD3LM ( $L' = 4$ )	420

# Future Work

## Frequency-Informed Masking

Prioritize masking rare, information-rich tokens (which carry more semantics) rather than common function words, changing *which* tokens are masked, not only *how many*.

## Adaptive block size

Adaptively change the block size according to the need for fine-grained detail of each block.

## mHC (DeepSeek)

Explore mHC and related recent ideas from the DeepSeek group, and assess how they could be combined with masked/block diffusion to improve the quality–efficiency trade-off.

# Table of Contents

## 1 Paper Overview

## 2 Our Results

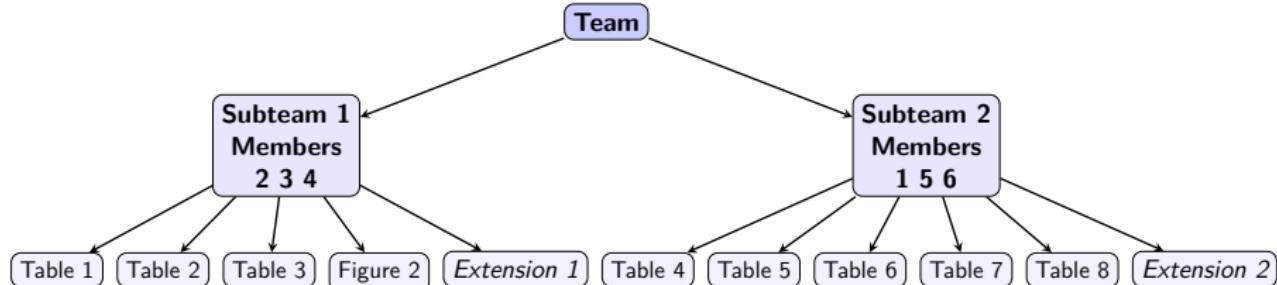
- Reproduction
- Extensions
  - Alternative Noise Schedules
  - Loss Reweighting in the NELBO Objective

## 3 Conclusion & Future Work

## 4 Team Organization

## 5 Retrospection

# Team Organization and Structure



## Coordination:

- Weekly team meetings
- Subteams internal coordination
- GitHub repo
- Discord server

# Member Roles

---

Phase	Description	Members
1	Paper & codebase study	Everyone
2	Environment & data setup	Georgios M., Petros
3	Initial reproduction	Georgios M., Petros, Ilias
4	Final reproduction	Georgios Nt., Konstantinos, Nikolaos
5	Extension experiments	Everyone
6	Analysis & Writing	Everyone

---

# Table of Contents

## 1 Paper Overview

## 2 Our Results

- Reproduction
- Extensions
  - Alternative Noise Schedules
  - Loss Reweighting in the NELBO Objective

## 3 Conclusion & Future Work

## 4 Team Organization

## 5 Retrospection

# Retrospection

- **What challenged us:**

- Limited resources, which constrained training length and the number of ablations we could run.
- The LM1B training pipeline depended on a Hugging Face repository that was temporarily unavailable, causing delays and forcing workarounds.

- **What we could have done better:**

- Align the two subteams on a shared training protocol (same number of training steps and checkpoints) to make results more directly comparable.

Thank You!

Questions?

**Paper:** Block Diffusion (ICLR 2025)

**Code:** <https://github.com/kuleshov-group/bd3lms>

**Our Repo:** <https://github.com/ntua-el21050/bd3lms>

**Authors:** Arriola, Gokaslan, Chiu, Yang, Qi, Han, Sahoo, Kuleshov