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

Table of Contents

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

Block 1 → Block 2 → Block 3 $\cdots$

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

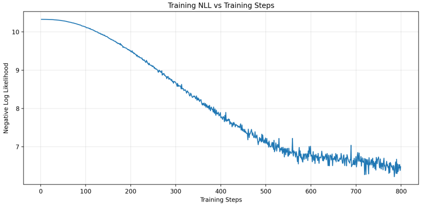
Table of Contents

Table of Contents

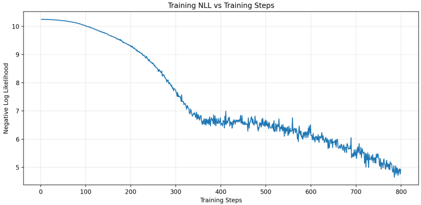
AR vs BD3LM with L'=1

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

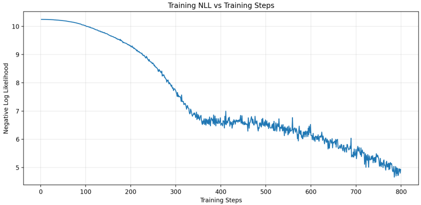
	PPL (↓)
Autoregressive	**1893**
BD3LM L'=1	2231
BD3LM L'=1 + Tuned Schedule	2220



AR



BD3LM



BD3LM + Tuned Schedule

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	$\mathcal{U}[0, 0.5]$	**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**

Original vs Reproduction

Key takeaways:

- **L'=4, 128:** Original clipping improves PPL & Variance; Reproduction shows no significant change.
- **L'=16:** Reproduction mirrors the original's improvements (lower PPL/Variance), but with smaller margins.

BD3LMs vs ARs vs Diffusion Models on LM1B

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

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

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, but original has *all* BD3LM variants beating diffusion.

BD3LMs vs ARs vs Diffusion Models on OWT

Test perplexities (PPL; ↓) of models on OWT. (3000 Pretraining Steps + 3000 Fine-tuning Steps)

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

Original vs Reproduction

Key takeaways:

- *AR > DIFF:* AR outperforms diffusion baselines (lower PPL).
- *BD3LM > AR:* Reverse arrangement of models

Conclusion: Bd3-lms outperform AR small scale models with constrained resources.

Performance on other Datasets

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

Original vs Reproduction

Key takeaways:

- *Wikitext:* Original → AR > **all** and **Bd3-lms** > Diff
  Reproduction → AR > all, **Diff** > Bd3-lms.
- *LM1B:* Reproduction **Bd3-lms** > all.
- *Lambada:* Original → **Diff** > all
  Reproduction → **Bd3-lms** > all

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 OWT. (400 + 100 training steps)

Model	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

Original vs Reproduction

Key takeaways:

- *Sampling Quality:* Confirmed Bd3-lms > Diff for same NFEs
- *AR performance gap:* AR ¿ all in original vs all ¿ AR in reproduction.

NFEs ≥ L. Sampling quality improves for Bd3-lms (Gen.PPL;↓). AR models converge slowly.

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	**15.23**
$\mathcal{U}[0.3, 0.8]$	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**
$\mathcal{U}[0.3, 0.8]$	798	6.41
Linear $\mathcal{U}[0, 1]$	662	44.73
Square	**627**	27.53
Cosine	634	20.80

Original vs Reproduction

Key takeaways:

- *Confirmed Variance Hypothesis:* Clipped schedules NELBO variance $\downarrow$
- *Perplexity trade-off:* Unlike the paper, standard noise schedules achieve better PPL.

Low variance is achieved in less training steps than PPL gains for small scale models.

Table of Contents

Table of Contents

Noise Scheduling in Masked Diffusion (Summary)

Continuous time index:

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

Noise schedule $\Rightarrow$ masked probability:

$$p(t): \text{ masked probability induced by the noise schedule}$$

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	$p(t)$	$\textbf{loss\_scaling}(t)$
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 $t \in (0,1)$:

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

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 (μ_1, σ_1), otherwise use (μ_2, σ_2):

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

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 μ and μ_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 (μ =0.5)	2115	1.29
	Loglinear $\mathcal{U}[0,\ 0.5]$	2106	1.27	Gaussian (μ =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 (μ =0.5)	**1234**	**10.28**
	Loglinear $\mathcal{U}[0.3,\ 0.8]$	1278	10.51	Gaussian (μ =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 (μ =0.5)	1250	46.76
	Loglinear $\mathcal{U}[0.5,\ 1]$	**1226**	**44.41**	Gaussian (μ =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

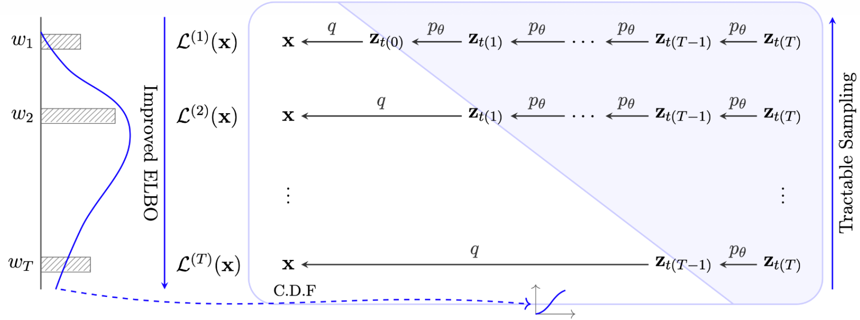
We can also apply the Bimodal Gaussian noise schedule during pretraining. The table reports, for each pretraining noise schedule, the *best* fine-tuning noise schedule we found, with the corresponding PPL and Var. NELBO (400 Pretraining Steps + 100 Fine-tuning Steps).

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. ($\mu_1 = 0.3, w_1 = 0.6$)	**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**

Table of Contents

Reweighted Losses for Improved Diffusion Objective

Reweighted Losses are Better Variational Bounds



Diffusion objectives: $\mathcal{L}^{\tilde{w}}(\mathbf{x}) = \lim_{T\to\infty} \sum_{i=1}^{T} w_i \mathcal{L}^{(i)}(\mathbf{x}) = \int_0^1 \tilde{w}(t) \mathbb{E}_{q(\mathbf{z}_t|\mathbf{x})} \left[L_{\text{denoise}}(\mathbf{z}_t, \mathbf{x}, t) \right] dt + \mathrm{C}$

- Initial Reweighted NELBO:

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

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

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

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		$\mathrm{sech}(\frac{\lambda}{2})$	$2\sqrt{\alpha_t(1-\alpha_t)}$
Sigmoid	$\log \frac{\alpha_t}{1-\alpha_t}$	$\mathrm{sigmoid}(-\lambda+k)$	$\frac{1-\alpha_t}{1-(1-e^{-k})\alpha_t}$
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 (↓)					
Autoregressive						
Transformer	1221					
Diffusion						
SEDD	1403					
MDLM	1370					
Block diffusion	**Base**	**IDDPM**	**EDM**	**Sigmoid (k = 0)**	**FM**	**Simple**
BD3-LMs $L' = 16$	1345	8048.88	1593.4	1150.15	76213	53070
$L' = 8$	1210	7954.17	1569.75	**1078.57**	109169	36010741760
$L' = 4$	1176	7857.33	1565.6	1093.7	67332	2396260

Table of Contents

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*.

Combine both extensions and test with more resources

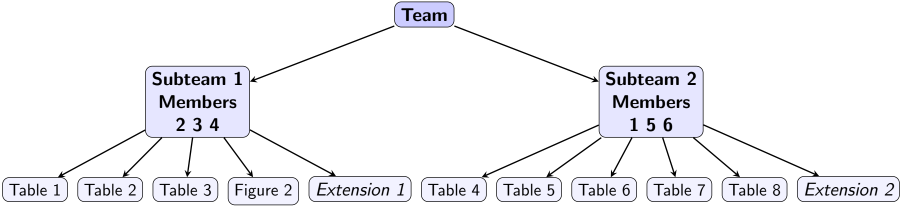
The results from our extensions although not decisive are promising, and both ideas are very well-suited for our setting, which when tested with more resources showed significant improvements.

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

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

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