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 ⋯

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

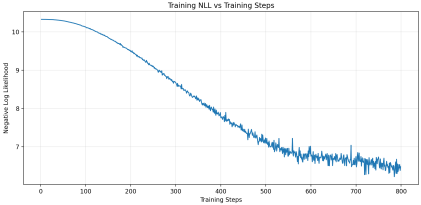
Table of Contents

Table of Contents

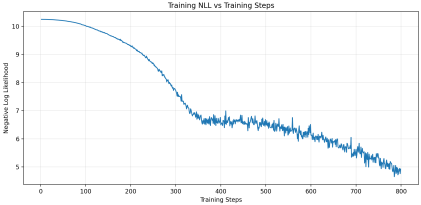
AR vs BD3LM with L'=1

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

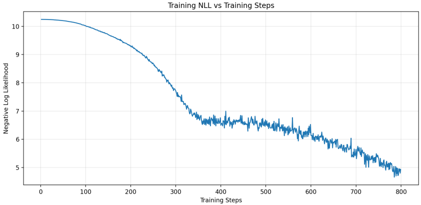
	PPL (↓)
Autoregressive	**1893**
BD3LM L'=1	2231
BD3LM L'=1 + Tuned Schedule	2220



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 (Orig. vs Repro.):

- **$L' = 128$ *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 $\mathcal{U}[0, 1]$; Repro. 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 (↓)
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 (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 (↓)
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 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 reprduction 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 (↓) of models trained on OWT.(?training 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 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).

	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:* In both, SEDD is capped at 1024 max, while BD3LM exceeds 1024; however original shows a much larger BD3LM max than reproduction.
- *Median ordering preserved:* Both original and reproduciton keep the same ranking (AR median > SEDD > BD3LM > OWT train).

Conclusion: reproduction preserves the ordering, but underestimates BD3LM's max-length gain, likely missing long-tail samples at this scale.

Sample Quality

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

	$L = 1024$		$L = 2048$	
Model	Gen. PPL	NFEs	Gen. PPL	NFEs
AR	14.1	1K	13.2	2K
Diffusion				
SEDD	52.0	1K	–	–
MDLM	46.8	1K	41.3	2K
Block Diffusion				
SSD-LM $L' = 25$	37.2	40K	35.3	80K
281.3	1K	281.9	2K	
BD3-LMs $L' = 16$	32.97	1K	31.42	2K
$L' = 8$	29.35	1K	27.42	2K
$L' = 4$	**24.74**	1K	**23.85**	2K

Effect of Different Noise Schedules

Effect of noise schedule on PPL and variance of NELBO for different L' on LM1B. (?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

Table 8 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

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:

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

Intuition

The noise schedule sets *where* the model learns via $p(t)$ (masking rate). When sampling $t \sim \mathcal{U}[0,1]$, loss_scaling$(t) = \frac{a'(t)}{1-a(t)}$ acts as a weight on the per-token log-likelihood term so the discrete estimator matches the continuous-time integral/ELBO and balances low- vs high-noise regions.

Already Implemented Noise Schedules

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	$1-(1-\varepsilon)\cos\left(\dfrac{\pi t}{2}\right)$	$-\dfrac{\left(\frac{\pi}{2}\right)(1-\varepsilon)\sin\left(\frac{\pi t}{2}\right)}{1-(1-\varepsilon)\cos\left(\frac{\pi t}{2}\right)}$

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: $\sigma^2 = x$ (Gaussian) and $\sigma_1^2 = x, \sigma_2^2 = x$ (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

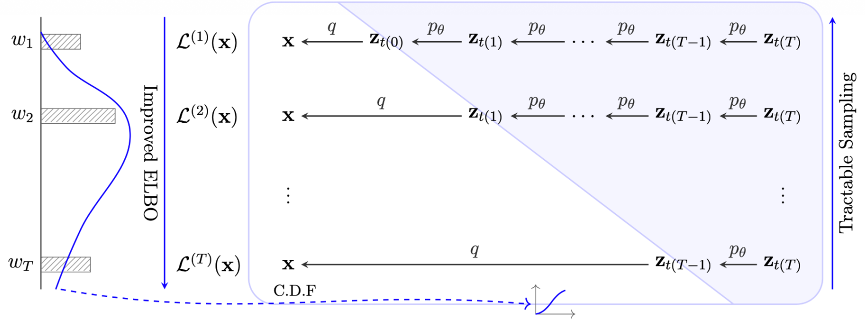
We can also apply the Bimodal Gaussian noise schedule during pretraining. The table reports, for each pretraining noise schedule, the *best* fine-tuning noise schedule we found, with the corresponding PPL and Var. NELBO (400 Pretraining Steps + 100 Fine-tuning Steps).

Block Size	Pretraining Schedule	Fine-tuning Schedule	PPL	Var. NELBO
128	Loglinear	?????	1000	10.00
	Bimodal Gaussian	?????	1000	10.00
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: $\mathcal{L}^{\tilde{w}}(\mathbf{x}) = \lim_{T\to\infty} \sum_{i=1}^{T} w_i \mathcal{L}^{(i)}(\mathbf{x}) = \int_0^1 \tilde{w}(t) \mathbb{E}_{q(\mathbf{z}_t|\mathbf{x})} \left[L_{\text{denoise}}(\mathbf{z}_t, \mathbf{x}, t) \right] dt + \mathrm{C}$

Reweighted Losses for Masked Diffusion

- Initial Reweighted NELBO:

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

- 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	252	49.88	36.06	76213	53070
$L' = 8$	1210	249	49.14	35.79	109169	36010741760
$L' = 4$	1176	246	49.01	**35.08**	67332	2396260

Table of Contents

Table of Contents

Table of Contents