SCALING BEHAVIOR OF DISCRETE DIFFUSION LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Modern LLM pre-training consumes vast amounts of both compute resources and training data, making the scaling behavior, or scaling laws, of different models a key distinguishing factor. Discrete diffusion language models (DLMs) have been proposed as an alternative to autoregressive language models (ALMs). However, their scaling behavior has not yet been fully explored, with prior work suggesting that they require more data and compute to match the performance of ALMs.

We study the scaling behavior of DLMs on different noise types by smoothly interpolating between masked and uniform diffusion while paying close attention to crucial hyperparameters such as batch size and learning rate. Our experiments show that the scaling behavior of DLMs strongly depends on the noise type and is considerably different from ALMs. Surprisingly, we find that uniform diffusion requires more parameters and less data for compute-efficient training compared to masked diffusion. Moreover, uniform diffusion models scale more favorably in both compute and data than their masked counterparts, making them a promising option in both compute- and data-bound training environments. In the process of deriving the scaling laws, we reformulate the discrete diffusion ELBO in terms of signal-to-noise ratio, closing the gap to continuous diffusion theory and simplifying both theory and implementation. We also find that DLMs have an optimal batch size with no signs of saturation, which is in contrast to ALMs, which typically show diminishing returns from scaling batches beyond 10^6 tokens. Training code and models are open-sourced: upon acceptance

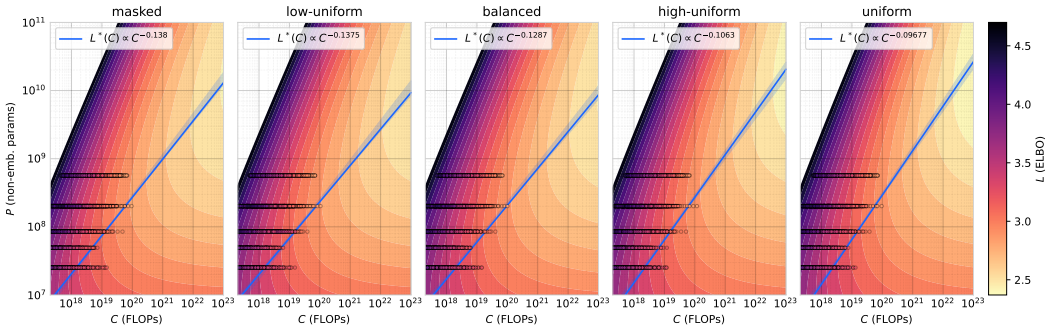


Figure 1: We propose five new scaling laws for discrete diffusion language models, finding that uniform diffusion scales most favorably in both compute- and data-bound settings.

1 INTRODUCTION

Diffusion language models (DLMs) have recently emerged as an alternative to autoregressive language models (ALMs), promising to address some fundamental limitations plaguing ALMs such as the inability to generate multiple tokens in parallel as well as the inability to revise previously generated tokens (Li et al., 2025). While DLMs' performance at small scales lags behind autoregressive models, they have the potential to solve both of these limitations by decomposing the generative process into a sequence of denoising steps where the entire generated sequence of N tokens is gradually refined, starting at pure noise and transforming it to pure signal over the course of T denoising

steps. The freedom to choose T independently of N enables the generation of multiple tokens in each step, while also retaining the ability to update every token at every step.

Within DLMs, masked diffusion models (MDMs) have emerged as the predominant DLM archetype next to alternative diffusion processes such as uniform diffusion (Austin et al., 2021) or hybrid-noise diffusion (von Rütte et al., 2025). MDMs work by gradually masking tokens and training a model to undo this degradation process by filling in the missing tokens. In contrast, uniform diffusion replaces tokens with random other tokens from the vocabulary until, eventually, every token in the sequence is completely random. Hybrid diffusion models lie on the spectrum between masking and uniform diffusion, utilizing some combination of both noise types. MDMs have gained popularity due to their superior performance at small scales, but face significant challenges despite their dominance. Prior work has suggested that MDMs are less efficient to train, requiring 16x more compute in a compute-optimal setting to match the training loss of ALMs (Nie et al., 2025). Additionally, like ALMs, MDMs suffer from the inability to revise previously generated tokens. This is due to the fact that every token experiences exactly one state transition (between its masked and unmasked state), hence prohibiting any transitions between two unmasked states. This has prompted the realization that alternative diffusion models were, perhaps, abandoned prematurely.

The performance gap (measured in perplexity, or training loss) between autoregressive, masked diffusion, and uniform diffusion models can be explained, at least in part, through the lens of task difficulty: MDMs are trained to generate the data in any random order, which includes, but is not limited to, generating the data in its natural, autoregressive order and is therefore a strictly more difficult problem (Kim et al., 2025). Similarly, uniform diffusion can be understood as a strictly more difficult version of masked diffusion where the model has to predict which tokens are noisy and which are noise-free in addition to subsequently imputing the noisy tokens. Put differently, going from autoregression to masking to uniform diffusion imposes progressively less structure on the generative process and therefore provides less inductive bias, suggesting that a more expressive model is required to learn the task effectively.Crucially, the scaling behavior of uniform and hybrid-noise DLMs remains an open question, with existing work being limited to small-scale ablations. Furthermore, prior work on scaling MDMs (Nie et al., 2025) makes some potentially undesirable design choices, such as assuming that the training loss can approach zero given infinite compute as well as fixing the learning rate and batch size to a constant value, which casts doubt on the optimality of the reported scaling laws.

In this work, we refine the strategy from Nie et al. (2025) by putting additional care on tuning crucial hyperparameters and modeling the loss surface as a power law of model size and training tokens. This way, we determine and compare the scaling behavior of masked, uniform, and hybrid-noise diffusion models. Our contributions are three-fold:

(1) Diffusion process. To aid with scaling across different noise types, we propose a new family of hybrid diffusion that allows us to easily and smoothly interpolate between masked and uniform diffusion by defining a transition point from masking to uniform diffusion depending on the signal-to-noise-ratio (SNR). We argue that defining the diffusion process through SNR rather than time is more natural and more principled, having become the standard for continuous-state diffusion (Kingma et al., 2021; Kingma & Gao, 2023; Karras et al., 2024). To derive the ELBO of the proposed diffusion process, we frame it as an instance of generalized interpolating discrete diffusion (GIDD) (von Rütte et al., 2025) and reparameterize the GIDD ELBO in terms of SNR. This reparameterization simplifies both theory and implementation, while also closing the gap to continuous-state diffusion theory and showing that discrete diffusion, like continuous diffusion, is invariant to the noise schedule (Kingma et al., 2021).

(2) Methodology. We then systematically analyze the scaling behavior across all noise types (masking, uniform, and hybrid), model sizes, training durations, and batch sizes. To aid with scaling, we utilize CompleteP (Dey et al., 2025) for stable learning rate transfer across model width and depth. Instead of fixing the batch size to a constant value, as is often done in prior work on scaling laws, we find it to be a crucial hyperparameter with an optimal value depending on the training token budget. Thus, it requires careful tuning at each scale, leading us to estimate the scaling laws without learning rate annealing in order to cope with this additional scaling dimension. This is motivated by the recent trend of treating pre-training and annealing as two distinct training stages conducted on potentially different datasets (Project Apertus, 2025; Allal et al., 2025), as well as our own ablations

showing that training with and without annealing yields similar optima and a similar loss, up to some constant factor.

(3) Scaling behavior. The discovered scaling laws paint a picture that is exceedingly favorable for uniform diffusion: Not only does its compute-optimal loss scale most favorably with increased training compute, it also requires fewer training tokens per parameter compared to both masked diffusion and ALMs, making it more data efficient at compute-optimality. Furthermore, the scaling behavior across noise types changes smoothly, with MDMs having the most similar scaling coefficients to ALMs, albeit slightly more parameter-heavy. Beyond pure masking, increasing levels of uniform noise scale progressively better. This makes uniform diffusion a potential competitor to the predominant autoregressive paradigm, with the potential to even outperform ALMs at very large scales. Beyond the loss surface, we also find that the optimal values for batch size and learning rate are remarkably predictable, with the optimal batch size being a function of dataset size, optimal learning rate being a function of batch size and step count, and both being independent of model size and noise type.

The paper is structured accordingly. We will first introduce discrete diffusion, rederive the diffusion ELBO in terms of SNR, and introduce the diffusion process to be used in subsequent experiments. We then outline our methodology for estimating scaling laws, motivating and justifying our design choices. Finally, we present our experimental results, providing an overview of the model architecture and training procedure and analyzing the observed scaling behavior. An overview of related work is provided in App. B.

2 Discrete Diffusion

Diffusion models (Sohl-Dickstein et al., 2015) decompose the generative distribution into a Markov chain of progressively more noisy versions of the data, and train a neural network to reverse this degradation process. In its most general form, this Markov chain is given by an initial state $z_0 = x$ which represents the data, a state transition distribution $q_{t|s}(z_t|z_s)$ that defines how the state evolves between times s and t with $0 \leq s < t \leq 1$, and finally a prior distribution $p_{\text{prior}}(z_1)$ that represents the stationary distribution of $q_{t|s}$ as t approaches 1 and is easy to sample from. For discrete diffusion models (Austin et al., 2021; Campbell et al., 2022), we consider the special case where states exist in a discrete space $\mathcal{Z}$ and the state-transition distribution can therefore be simply described by a vector $\boldsymbol{q}_{t|s}(z_s) \in \Delta^{|\mathcal{Z}|-1}$ where Δ^k denotes the k-simplex and $|\mathcal{Z}|$ is the cardinality of $\mathcal{Z}$.

2.1 Generalized Interpolating Discrete Diffusion

Generalized interpolating discrete diffusion (GIDD) (von Rütte et al., 2025) provides a unified perspective on many existing discrete diffusion processes such as masked or uniform diffusion by obtaining a closed-form evidence lower-bound (ELBO) for arbitrary, time-varying mixing distributions $\boldsymbol{\pi}_t \in \Delta^{|\mathcal{Z}|-1}$. The marginal and conditional state-transition distributions of GIDD models are

$$\boldsymbol{q}_t(x) = \alpha_t \boldsymbol{x} + \beta_t \boldsymbol{\pi}_t \tag{1}$$

$$\boldsymbol{q}_{t|s}(z_s) = \alpha_{t|s} \boldsymbol{z}_s + \beta_{t|s} \boldsymbol{\pi}_{t|s}, \tag{2}$$

with $\beta_t = 1 - \alpha_t$, $\alpha_{t|s} = \alpha_t / \alpha_s$, and $\beta_{t|s}\boldsymbol{\pi}_{t|s} = \beta_t \boldsymbol{\pi}_t - \alpha_{t|s}\beta_s \boldsymbol{\pi}_s$. Under the condition that α_t and $\boldsymbol{\pi}_t$ are differentiable in time, the diffusion negative ELBO (NELBO) of GIDD is given by

$$-\log p_\theta(x) \leq \mathbb{E}_{t\sim\mathcal{U}(0,1), z\sim\boldsymbol{q}_t(x)} \left[\boldsymbol{w}_t(x)_z \{ D_{KL}(\boldsymbol{q}_t(x) \| \boldsymbol{q}_t(\boldsymbol{x}_\theta)) + D_{IS}(\boldsymbol{q}_t(x)_z \| \boldsymbol{q}_t(\boldsymbol{x}_\theta)_z) \} \right] + C, \tag{3}$$

with $D_{IS}(p\|q) = p/q - \log p/q - 1$ denoting the (point-wise) Itakura-Saito divergence and $\boldsymbol{w}_t(x)$ is the weighting vector

$$\boldsymbol{w}_t(x) = \boldsymbol{q}_t(x)^{-1} \left(\beta_t \boldsymbol{\pi}_t' - \frac{\alpha_t'}{\alpha_t} \boldsymbol{\pi}_t \right). \tag{4}$$

We adopt the framework of von Rütte et al. (2025) as it allows us to train discrete diffusion models with different noising properties within a shared framework, reducing precisely to specialized variants in the literature under an appropriate mixing schedule. However, we improve this framework by showing how it can be reformulated in terms of signal-to-noise ratio, obtaining a simpler, more flexible likelihood bound and closing the gap to continuous-state diffusion theory.

2.2 Reframing Discrete Diffusion in terms of SNR

It is well-known that continuous-state diffusion models are invariant to the noise schedule (Kingma et al., 2021), with many approaches relying on this fact to accelerate training via adaptive noise schedules (Kingma & Gao, 2023; Karras et al., 2024; Dieleman, 2024). This stems from the insight that the notion of time in diffusion models is spurious and serves only as a proxy for the signal-to-noise ratio (SNR), and that SNR is sufficient and, arguably, a more natural way to describe the forward and backward diffusion process. In this section, we show that this invariance continues to hold for discrete diffusion models through the same proof technique as used by Kingma et al. (2021).

To begin, we define the SNR and log-SNR in terms of the mixing rate (or noise schedule) α_t as this is the quantity that determines the proportion of the data distribution (or signal) that is preserved at any given time t. Let

$$\mathrm{SNR} = \frac{\alpha}{1-\alpha} \quad \text{and} \quad \lambda = \log \mathrm{SNR} = \log \frac{\alpha}{1-\alpha}. \tag{5}$$

Notably, this results in α being a sigmoid function of λ, with

$$\alpha = \sigma(\lambda) = \frac{1}{1+e^{-\lambda}}. \tag{6}$$

We will then perform a change-of-variable on the GIDD ELBO, changing the differential from dt to $d\lambda$. Noting the relation between α and λ, we can rewrite the time-derivative of α as

$$\alpha_t' = \frac{d\alpha}{dt} = \frac{d\alpha}{d\lambda}\frac{d\lambda}{dt} = \frac{d}{d\lambda}\sigma(\lambda) \cdot \frac{d\lambda}{dt} = \sigma(\lambda)\sigma(-\lambda)\frac{d\lambda}{dt} \tag{7}$$

For $\boldsymbol{w}_t(x)$, we then get

$$\begin{aligned}\boldsymbol{w}_t(x) = \boldsymbol{q}_t(x)^{-1}\left(\beta_t\boldsymbol{\pi}_t' - \frac{\alpha_t'}{\alpha_t}\boldsymbol{\pi}_t\right) &= \boldsymbol{q}_t(x)^{-1}\left((1-\sigma(\lambda))\frac{d\boldsymbol{\pi}_\lambda}{d\lambda}\frac{d\lambda}{dt} - \frac{\sigma(\lambda)\sigma(-\lambda)}{\sigma(\lambda)}\frac{d\lambda}{dt}\boldsymbol{\pi}_\lambda\right) \\ &= \boldsymbol{q}_t(x)^{-1}\sigma(-\lambda)\frac{d\lambda}{dt}\left(\boldsymbol{\pi}_\lambda' - \boldsymbol{\pi}_\lambda\right). \end{aligned} \tag{8}$$

Plugging this into Eq. 3, and abbreviating $E_z(\boldsymbol{p},\boldsymbol{q}) := D_{KL}(\boldsymbol{p}\|\boldsymbol{q}) + D_{IS}(\boldsymbol{p}_z\|\boldsymbol{q}_z)$ yields

$$-\log p(x) \le \mathbb{E}_{t,z}\left[\boldsymbol{w}_t(x)_z E_z(\boldsymbol{q}_t(x), \boldsymbol{q}_t(\boldsymbol{x}_\theta))\right] + C \tag{9}$$

$$= \int_0^1 dt \frac{d\lambda}{dt}\sum_z \sigma(-\lambda)(\boldsymbol{\pi}_\lambda' - \boldsymbol{\pi}_\lambda)_z E_z(\boldsymbol{q}_t(x), \boldsymbol{q}_t(\boldsymbol{x}_\theta)) + C \tag{10}$$

$$= \int_{\lambda_{\min}}^{\lambda_{\max}} d\lambda \sum_z \sigma(-\lambda)(\boldsymbol{\pi}_\lambda - \boldsymbol{\pi}_\lambda')_z E_z(\boldsymbol{q}_\lambda(x), \boldsymbol{q}_\lambda(\boldsymbol{x}_\theta)) + C. \tag{11}$$

This reveals that the ELBO is invariant not only to the SNR distribution induced by $p(\lambda) = -dt/d\lambda$ but also to the forward process marginals $\boldsymbol{q}_\lambda(x)$, and that their purpose is to approximate this integral through importance sampling. Accordingly, we can convert this back to an expectation like

$$-\log p(x) \le \mathbb{E}_{\lambda,z}\left[\frac{\boldsymbol{w}_\lambda(x)_z}{p(\lambda)}\{D_{KL}(\boldsymbol{q}_\lambda(x)\|\boldsymbol{q}_\lambda(\boldsymbol{x}_\theta)) + D_{IS}(\boldsymbol{q}_\lambda(x)_z\|q_\lambda(\boldsymbol{x}_\theta)_z)\}\right] + C, \tag{12}$$

with $\lambda \sim p(\lambda)$, $z \sim \boldsymbol{q}_\lambda(x)$, and $\boldsymbol{w}_\lambda(x)_z = \frac{\sigma(-\lambda)(\boldsymbol{\pi}_\lambda - \boldsymbol{\pi}_\lambda')_z}{\boldsymbol{q}_\lambda(x)_z}$ denoting the updated weighting term.

2.3 A Universal Hybrid Mixing Distribution

For our scaling experiments, we are looking for a mixing distribution $\boldsymbol{\pi}_\lambda$ that allows smoothly transitioning from masked to uniform diffusion, covering a range of hybrid mixtures in between. The basic idea is to interpolate between the pure masking and pure uniform noise based on the current log-SNR λ, thereby controlling how much masking and how much random perturbation happens proportionally at any point of the noising process. We define

$$\boldsymbol{\pi}_\lambda = \sigma(\lambda + b)\boldsymbol{u} + \sigma(-\lambda + b)\boldsymbol{m}, \tag{13}$$

with σ denoting the sigmoid function, $\boldsymbol{u} = \frac{1}{N-1}(1 - \boldsymbol{e}_m)$ and $\boldsymbol{m} = \boldsymbol{e}_m$ denoting the uniform and masking probability vector respectively, and b being a hyperparameter that controls the transition point between masking and uniform noise. Note that this mixing distribution approaches pure masking as $b \to -\infty$ and pure uniform noise as $b \to \infty$, with varying masking-to-uniform mixtures in between. The reparameterized ELBO enables trivial implementation of this mixing distribution, only requiring computation of the derivative of $\boldsymbol{\pi}'_\lambda$, which is given by $\boldsymbol{\pi}'_\lambda = \sigma'(\lambda + b)(\boldsymbol{u} - \boldsymbol{m})$.

3 Estimating Scaling Laws

Scaling laws have become an important ingredient of large-scale neural network training, particularly in the context of training LLMs. Due to the vast costs associated with large-scale training runs, key decisions are based on forecasts obtained through extrapolating the performance of smaller runs to the desired, bigger scale. Prior work on the scaling of MDMs (Nie et al., 2025) has made some assumptions that we would like to revisit. For example, the learning rate and batch size are fixed to constant values across all experiments, but this may not be optimal for different model sizes and token budgets (Bergsma et al., 2025). Additionally, the reported scaling law is the result of a power law fit without constant offset, thereby implicitly assuming that the ideal training loss is zero and can be reached given infinite compute, which is well-known not to be the case. These limitations prompt us to rederive the scaling from scratch, dropping any assumptions on the optimal batch size and learning rate, and applying a more standard power law fit. While our recipe largely follows the methodology by (Hoffmann et al., 2022), which is well-established and has been widely adopted for estimating scaling laws (Touvron et al., 2023; Bi et al., 2024; Shuai et al., 2024), there are some key differences.

Maximal Update Parameterization To aid with the scaling process, we adopt CompleteP (Dey et al., 2025), a variant of μP (Yang et al., 2022) that parameterizes the model in a way such that optimal learning rates transfer both across width and depth. Unlike the original work, we do not employ a base width to keep learning dynamics equivalent to some reference model and instead find the optimal values for weight initialization variance and base learning rate through a hyperparameter sweep on a 25M and 50M parameter model. This results in different optimal values for width-dependent parameters (bulk parameters) such as weight matrices compared to non-width-dependent parameters such as layer-normalization and bias parameters (auxiliary parameters), with bulk parameters requiring a larger initialization variance and learning rate. We find optimal values of $\sigma_{\text{base}} = 0.4$, $\sigma_{\text{aux}} = 0.02$ and $\eta_{\text{base}} = 0.3$, $\eta_{\text{aux}} = 0.02 \cdot \eta_{\text{base}}$ for initialization variances and learning rates respectively (at a batch size of 64). These values transfer remarkably well in our experiments, with only the base learning rate η_{base} requiring adjustments depending on the batch size.

Learning Rate Annealing. While adopting CompleteP enables learning rate transfer across model scales, the same learning rate is not optimal for different batch sizes and training horizons, implying that it is necessary to sweep the learning rate for each model and batch size in order to find the compute-optimal Pareto frontier. To cope with the computational demands of sweeping the batch size in addition to model and data size, we decide to omit learning rate annealing and analyze the scaling behavior without it, which allows capturing all possible training horizons in a single run per model and batch size. This decision is justified twofold: First, modern large-scale training often treats the annealing phase as a distinct phase from pre-training where the data mixture is often adapted to more closely resemble the test distribution by injecting more high-quality data geared towards the desired downstream tasks (Project Apertus, 2025; Allal et al., 2025). Second, we conduct small-scale ablations to study the effect of omitting annealing, finding that the optimal hyperparameters are preserved and that the performance difference is approximately a constant factor (see Sec. 4.2).

Optimal Batch Size. Sweeping the learning rate across batch and model sizes reveals the clear existence of a compute- and data-optimal batch size that scales almost linearly in the number of training tokens (Fig. 3). Similar findings have been reported for ALMs when training below the *critical batch size* (Hu et al., 2024; Shuai et al., 2024; Bergsma et al., 2025). The critical batch size refers to the phenomenon where scaling the batch size past a certain *critical* point yields diminishing returns and becomes compute-inefficient. It is worth noting that we find no dependence of the optimal batch size on the target loss, a claim that has been raised for the critical batch size of ALMs

(Zhang et al., 2024) and also more generally (McCandlish et al., 2018). As our experiments show no signs of saturation even at batch sizes at 10^6 tokens, this suggests that the critical batch size of DLMs lies well above that of ALMs, which has been reported to saturate around 10^6 tokens (Shuai et al., 2024; Zhang et al., 2024). Another hyperparameter that is known to have optimal values depending on the batch size is Adam's β_2 parameter. However, also for this parameter we find little benefit in deviating from our default value of 0.99: Neither do we observe benefits from using larger values for small batch sizes[1] nor from using smaller values at larger batch sizes.[2] We do decrease β_2 to 0.98 starting at batch sizes of 256 as we found this to slightly improve stability without any noticeable performance degradation.

4 Experiments and Results

4.1 Model Architecture and Training

Architecture. Our model architecture follows a standard Transformer (Vaswani et al., 2017) with some key modifications. As described in Section 3, we implement CompleteP (Dey et al., 2025) for optimal learning rate transfer across width and depth. To ensure stable training, we add RMSNorm (Zhang & Sennrich, 2019) layers without bias before each attention and MLP block following LLaMA (Touvron et al., 2023). In the same spirit, we employ both QK-norm (Naseer et al., 2021; Dehghani et al., 2023) and attention logit soft-capping (Gemma Team, 2024). Finally, we add attention sinks in the form of attention biases (Sun et al., 2024) to further stabilize training and to prevent outlier features that can make quantization more challenging (Sun et al., 2024; He et al., 2024).

Data. We use Nemotron-CC (Su et al., 2024) without quality filtering as a representative dataset of internet-scale pre-training. Since it is known that a larger vocabulary facilitates better scaling (Takase et al., 2024; Huang et al., 2025) and to ensure efficient tokenization, we train a BPE tokenizer (Gage, 1994; Sennrich et al., 2015) with a vocabulary size of 2^{17} (131,072) tokens on a 256 GB subset of the data. The trained tokenizer is released with the model.

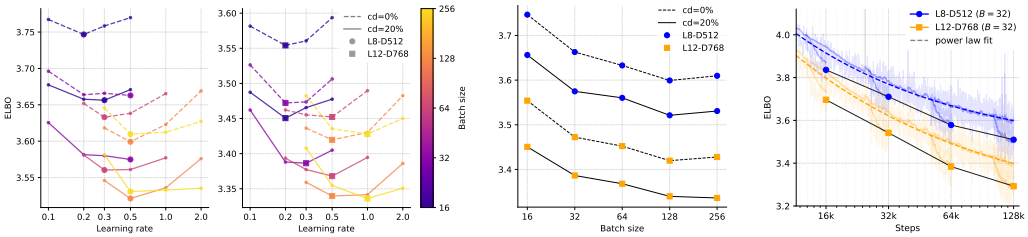
Diffusion process. We use the mixing distribution proposed in Section 2.3 with shift $b \in \{-1000, -2, 0, 2, 1000\}$, resulting in pure masking and pure uniform noise for $b = -1000$ and $b = 1000$ respectively, and hybrid noise with transition points at $t \in \{0.12, 0.5, 0.88\}$, which we refer to as low-uniform, balanced, and high-uniform noise respectively. To balance training stability and ELBO tightness, we restrict the log-SNR to $\lambda \in [-9, 9]$.

We design the diffusion process by aiming to maximize flexibility of the resulting model at inference time, supporting conditional prompt completion, advanced sampling algorithms, as well as flexible length generation. For conditional prompt completion, we select 20% of samples and leave the first $[N \cdot \arccos(r)]$, $r \sim \mathcal{U}(0, 1)$ tokens noise-free. Attention from prompt queries to completion keys is masked in order to enable KV-caching of the prompt during inference. To support both isotropic and anisotropic denoising, we implement diffusion forcing (Chen et al., 2024) by sampling independent per-token noise levels for 50% of samples. Finally, we augment the context with a random fraction $f \sim \mathcal{U}(0, 0.2)$ of empty tokens following Wu et al. (2025) to add some flexibility to the length of generated samples.

Optimization. Instead of directly minimizing the ELBO, we use the unweighted ELBO (Eq. 12 with $p(\lambda) := 1$) as a surrogate loss, as this has been found to give better convergence for both hybrid and masked diffusion models (von Rütte et al., 2025; Sahoo et al., 2025). Following Hafner et al. (2023), we use LaProp (Ziyin et al., 2020) over Adam for its improved stability on a wider range of β_2 and ϵ values. The learning rate is warmed up over the first 2000 steps of training and held constant, with most experiments not including a cooldown phase. For the experiments that do have a cooldown phase, we anneal the learning rate to 0 over the last 20% of training following the WSD schedule (Hu et al., 2024; Hägele et al., 2024).

[1] This finding is particular to half-precision training in `bfloat16`, as we did observe slight benefits from increasing β_2 for full-precision training at low batch sizes.

[2] This improved consistency may be a result of using the LaProp (Ziyin et al., 2020) variant of Adam, or due to the CompleteP (Dey et al., 2025) parameterization, although we do not investigate this further.



(a) (b) (c)

Figure 2: Comparing optimal hyperparameters (a, b) and different training horizons (c) both without and with 20% learning rate cooldown reveals that learning rate annealing does not affect optimal hyperparameter values and brings a roughly constant-factor improvement across the board.

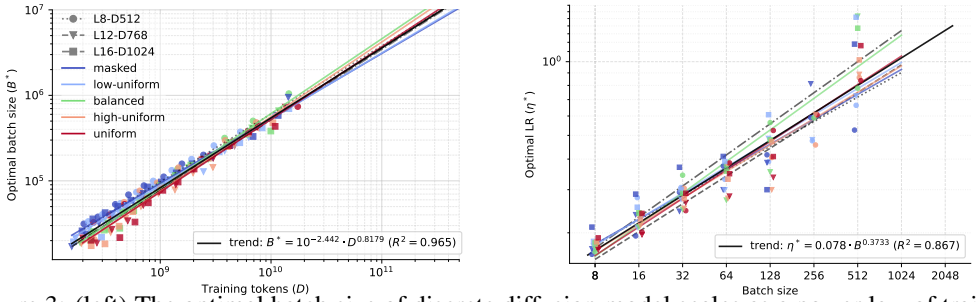


Figure 3: (left) The optimal batch size of discrete diffusion model scales as a power law of training tokens rather than training FLOPs or model size and has a scaling exponent of ~ 0.8. (right) The optimal learning rate with fixed step count (here: 50k steps) scales as a power law of batch size with an exponent of ~ 0.37, which is notably less than the often cited heuristic $\eta^* \propto B^{1/2}$.

4.2 Ablating the Effect of Learning Rate Annealing

For the sake of reducing the computational burden of scaling law estimation, we use a warmup-stable learning rate schedule without annealing, as outlined in Section 3. While we argue that this is a principled choice due to modern pre-training recipes spending most of the steps in the constant-LR regime and often treating the annealing as a separate phase (Project Apertus, 2025; Kimi Team, 2025; Allal et al., 2025), we empirically investigate the implications of this choice on a small-scale. Specifically, we investigate two settings: First, we fix the token budget while varying the batch size, sweeping the learning rate for each batch size both with and without annealing (Fig. 2; a, b). This reveals that both learning rate and batch size have stable optima that are largely unaffected by annealing, which only shifts the final loss by a constant factor. Second, we investigate how the annealed performance evolves over the course of a single training run, again finding that the shape of the annealed loss closely matches the unannealed trajectory (Fig. 2; c). We therefore conclude that the chosen simplification of omitting annealing from the scaling law estimation is valid, albeit that the projected loss will be higher by a constant factor. While these experiments are conducted on the balanced noise setting, we do not expect the conclusions to change depending on the noise type.

4.3 Relation Between Batch Size and Step Count

While the optimal batch size depends solely on the amount of training tokens, we additionally find that there is a tight relationship between batch size and training steps on ISO-loss curves. Specifically, we find that points with step count S, batch size B, optimal learning rate and the same observed loss L closely follow the relation

$$\left(\left[\frac{S}{S_{\min}}\right]^{\alpha} - 1\right)\left(\left[\frac{B}{B_{\min}}\right]^{\alpha} - 1\right) = 1, \tag{14}$$

which describes a hyperbola with asymptotes at $S_{\min}$ and $B_{\min}$ as well as a "stiffness" term α that control how fast the asymptotes are approached. This has some surprising implications: Not only does there appear to be a minimum step count and a minimum batch size required to reach a certain target loss for a fixed model size, but there also exists a token-optimal batch size and step count.

Noise type	$P^* \propto C^{\alpha_P}$	$D^* \propto C^{\alpha_D}$	$L^* \propto C^{-\alpha_L}$	$L(P, D)$
masked	0.554	0.446	0.139	$2.22 + \frac{43.8}{P^{0.252}} + \frac{634}{D^{0.313}}$
low-uniform	0.55	0.45	0.131	$2.17 + \frac{35.8}{P^{0.238}} + \frac{433}{D^{0.291}}$
balanced	0.539	0.461	0.129	$2.17 + \frac{36.8}{P^{0.239}} + \frac{365}{D^{0.28}}$
high-uniform	0.616	0.384	0.109	$2.01 + \frac{16.8}{P^{0.178}} + \frac{397}{D^{0.285}}$
uniform	**0.654**	**0.346**	**0.0963**	$1.83 + \frac{12}{P^{0.147}} + \frac{355}{D^{0.278}}$
MDM (Nie et al., 2025)	$0.634^\dagger$	$0.366^\dagger$	$0.0615^\dagger$	-
AR (Nie et al., 2025)	$0.644^\dagger$	$0.356^\dagger$	$0.0633^\dagger$	-
AR (Hoffmann et al., 2022)	0.452	0.548	0.154	$1.69 + \frac{406.4}{P^{0.34}} + \frac{410.7}{D^{0.28}}$
AR (Shuai et al., 2024)	0.464	0.536	0.153	$1.48 + \frac{314.4}{P^{0.331}} + \frac{460.5}{D^{0.286}}$

†Scaling coefficients are parsed from the provided plots as the original paper does not report them.

Table 1: We find that *uniform noise* has the best scaling behavior not only among diffusion models, but also in comparison to other scaling laws for autoregressive (AR) models on internet-scale datasets. The scaling behavior reported by Nie et al. (2025) differs considerably, which is likely due to the idealized assumption that the minimum achievable training loss is zero.

Optimizing the relation from Eq. 14 to minimize the token budget $D = BS$, we get optimal values $B^* = 2^{\frac{1}{\alpha}} B_{\min}$, $S^* = 2^{\frac{1}{\alpha}} S_{\min}$, and $D^* = 4^{\frac{1}{\alpha}} B_{\min} S_{\min}$. We find that α remains relatively constant across target losses, typically ranging between 0.1 and 0.2, while $S_{\min}$ and $B_{\min}$ appear to follow a power law in the target loss, growing to $10^{4.31}$ and $10^{9.9}$ respectively as the target loss approaches zero.[3] An illustrated example is given in App. A.1. This is in stark contrast to the often cited notion that small batches give better test accuracy (Keskar et al., 2016; Masters & Luschi, 2018; Smith et al., 2020). The tension can be explained by the fact that the cited results are in the context of better generalization in multi-epoch training due to gradient noise acting as a regularizer, whereas we operate under a sub-epoch training assumption where overfitting is not a concern. Indeed, more recent work corroborates the existence of optimal batch sizes as a function of training tokens in the context of ALM pre-training (Hu et al., 2024; Shuai et al., 2024; Bergsma et al., 2025). We leave the investigation of the dependence of this relation on the target loss and number of parameters to future work.

4.4 Scaling Laws and Compute-Optimal Frontier

To derive the scaling laws for the proposed class of diffusion models, we train models of five different sizes, ranging from 25M to 570M non-embedding parameters. For each model size, we sweep the learning rate across seven different batch sizes ranging from 2^{14} to 2^{20} tokens at a sequence length of 2048 tokens. We run smaller batch sizes for 10^5 optimizer steps, while reducing the number of steps to 5×10^4 starting at a batch size of 256 sequences.

We find that both the optimal batch size and the optimal learning rate follow a very predictable trend. The optimal batch size appears to depend primarily on the training horizon, with a remarkably strong, almost linear fit in the total number of training tokens. Similarly, the optimal learning rate follows a predictable trend in the batch size (Fig. 3). Recent work on scaling ALMs has reported similar predictable trends for both batch size and learning rate (Bi et al., 2024; Bergsma et al., 2025). Due to the predictability of the optimal learning rate, we sweep between only 2–3 different learning rates around the known optimal values for each batch size. Across all five noise types the resulting grid search spans 450 runs, of which 411 have completed as of the time of writing. See App. A.2 for details.

To determine compute-optimal settings for each model- and data-size pair, we select a set of target losses (in terms of ELBO and not the surrogate loss) and scan the observed loss curves for the minimum number of tokens required to achieve a given target loss across batch sizes and learning rates, grouped by noise type and model size. For smoothing, we apply a locally linear fit around the closest point to the target loss and determine the step count at the target loss based on the fit. This traces a compute-optimal Pareto frontier for each group, to which we fit a power law in the number

[3]Of course, a loss of zero is not achievable, so it is likely that this power-law relation will break down as the loss approaches its minimal value.

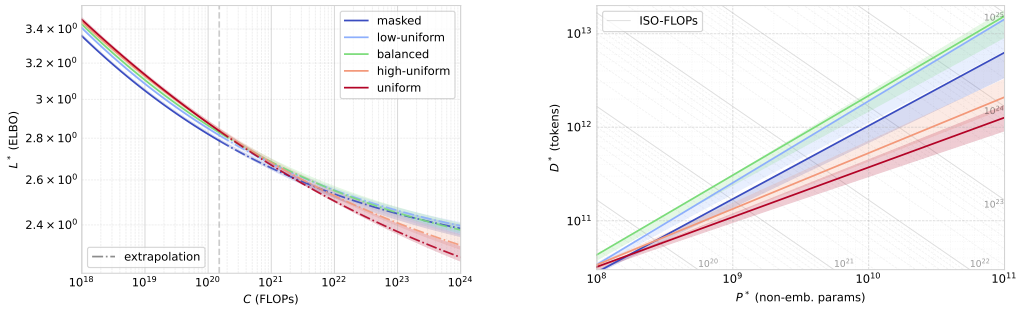


Figure 4: (left) According to the estimated scaling laws, we expect that uniform diffusion will outperform masked diffusion starting around 3×10^{21} FLOPs. (right) Compute-optimal scaling configurations vary considerably between noise types, with uniform diffusion having the lowest token-per-parameter ratio, making it the most data-efficient. Shaded regions represent 1σ-confidence intervals.

of non-embedding parameters P and the total number of training tokens D (Hoffmann et al., 2022):

$$L(P, D) = E + \frac{A}{P^\alpha} + \frac{B}{D^\beta} \tag{15}$$

We fit the coefficients to our observation using least-squares regression in log-space, employing a Huber loss for increased outlier robustness. From the fitted coefficients we then derive the compute-optimal scaling coefficients for the number of parameters P^* and training tokens D^* as well as the best achievable loss L^* as a function of training compute C in FLOPs (Tab. 1). As we observe the fit to be rather brittle and the extrapolation to large compute budgets being sensitive to slight changes in the scaling exponents, we estimate 1σ-confidence intervals using standard bootstrapping (Fig. 4).

Remarkably, we find a consistent trend in the scaling behavior of different noise types, with more uniform noise scaling more favorably with increased compute, requiring more parameters and less data to train compute-optimally. This is especially significant as the size of pre-training datasets is beginning to saturate while compute is continuing to become more abundant. Moreover, given that prior work has found comparable scaling behavior between autoregressive and masked diffusion models (Nie et al., 2025), this suggests that uniform diffusion models have the potential to outscale existing autoregressive training recipes. This finding is consistent with the notion that going from autoregressive modeling to masked diffusion to uniform diffusion imposes progressively less structure on the generation process and therefore less inductive bias, allowing it to scale more effortlessly with increased compute. Nevertheless, some limitations apply: When comparing scaling behavior across different datasets, the scaling coefficients can change depending on the data composition (Bi et al., 2024), thus making our numbers not directly comparable with those of Hoffmann et al. (2022) and Shuai et al. (2024) due to the use of Nemotron-CC (Su et al., 2024).

5 Conclusion

We have presented a comprehensive study of scaling laws of discrete diffusion language models, comparing different noise types ranging from masking to uniform noise and paying careful attention to crucial hyperparameters such as learning rate and batch size. The discovered scaling laws paint a favorable picture for both masked and uniform DLMs. We find comparable to slightly improved scaling of masked diffusion compared to autoregressive models, and significantly better scaling for uniform diffusion models. Remarkably, uniform diffusion models have better scaling coefficients than autoregressive models both in terms of data and compute, making them a strong contender for both data- and compute-bound scaling. This is consistent with the hypothesis that uniform diffusion imposes less of an inductive bias on the generative process.

Our findings support the case for discrete diffusion language models (DLMs) as a viable alternative to autoregressive language models (ALMs), the prevalent paradigm. DLMs can resolve core limitations of ALMs, enabling parallel generation for improved throughput, possessing the ability to revise and self-correct previously generated tokens, providing trivial ways of scaling test-time compute, and now also showing improved scaling behavior with increased training compute. All in all, we conclude that DLMs in general, and uniform diffusion in particular, are promising candidates for next-generation LLMs.

Reproducibility Statement

In order to facilitate transparency and reproducibility of our results, we release all of our training code as well as the code used for fitting the obtained scaling laws. Trained model weights are also released along with intermediate checkpoints.

Ethics Statement

This paper presents work whose goal is to advance the technical state-of-the-art in an area of Machine Learning. It shares potential societal consequences with much of the work in the general area of language modeling and foundation models.

References

Loubna Ben Allal, Anton Lozhkov, Elie Bakouch, Gabriel Martín Blázquez, Guilherme Penedo, Lewis Tunstall, Andrés Marafioti, Hynek Kydlíček, Agustín Piqueres Lajarín, Vaibhav Srivastav, et al. SmolLM2: When smol goes big–data-centric training of a small language model. *arXiv preprint arXiv:2502.02737*, 2025.

Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne Van Den Berg. Structured denoising diffusion models in discrete state-spaces. *Advances in neural information processing systems*, 34:17981–17993, 2021.

Gregor Bachmann, Sotiris Anagnostidis, and Thomas Hofmann. Scaling MLPs: A tale of inductive bias. *Advances in Neural Information Processing Systems*, 36:60821–60840, 2023.

Shane Bergsma, Nolan Dey, Gurpreet Gosal, Gavia Gray, Daria Soboleva, and Joel Hestness. Power lines: Scaling laws for weight decay and batch size in LLM pre-training. *arXiv preprint arXiv:2505.13738*, 2025.

Xiao Bi, Deli Chen, Guanting Chen, Shanhuang Chen, Damai Dai, Chengqi Deng, Honghui Ding, Kai Dong, Qiushi Du, Zhe Fu, et al. DeepSeek LLM: Scaling open-source language models with longtermism. *arXiv preprint arXiv:2401.02954*, 2024.

Andrew Campbell, Joe Benton, Valentin De Bortoli, Thomas Rainforth, George Deligiannidis, and Arnaud Doucet. A continuous time framework for discrete denoising models. *Advances in Neural Information Processing Systems*, 35:28266–28279, 2022.

Boyuan Chen, Diego Martí Monsó, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion forcing: Next-token prediction meets full-sequence diffusion. *Advances in Neural Information Processing Systems*, 37:24081–24125, 2024.

Lingjiao Chen, Hongyi Wang, Jinman Zhao, Dimitris Papailiopoulos, and Paraschos Koutris. The effect of network width on the performance of large-batch training. *Advances in neural information processing systems*, 31, 2018.

Enea Monzio Compagnoni, Tianlin Liu, Rustem Islamov, Frank Norbert Proske, Antonio Orvieto, and Aurelien Lucchi. Adaptive methods through the lens of SDEs: Theoretical insights on the role of noise. In *The Thirteenth International Conference on Learning Representations*, 2025.

Mostafa Dehghani, Josip Djolonga, Basil Mustafa, Piotr Padlewski, Jonathan Heek, Justin Gilmer, Andreas Peter Steiner, Mathilde Caron, Robert Geirhos, Ibrahim Alabdulmohsin, et al. Scaling vision transformers to 22 billion parameters. In *International conference on machine learning*, pp. 7480–7512. PMLR, 2023.

Nolan Dey, Bin Claire Zhang, Lorenzo Noci, Mufan Li, Blake Bordelon, Shane Bergsma, Cengiz Pehlevan, Boris Hanin, and Joel Hestness. Don't be lazy: CompleteP enables compute-efficient deep transformers. *arXiv preprint arXiv:2505.01618*, 2025.

Sander Dieleman. Noise schedules considered harmful, 2024. URL `https://sander.ai/2024/06/14/noise-schedules.html`.

Philip Gage. A new algorithm for data compression. *C Users J.*, 12(2):23–38, February 1994. ISSN 0898-9788.

Guillaume Garrigos and Robert M Gower. Handbook of convergence theorems for (stochastic) gradient methods. *arXiv preprint arXiv:2301.11235*, 2023.

Gemma Team. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*, 2024.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*, 2023.

Alex Hägele, Elie Bakouch, Atli Kosson, Leandro Von Werra, Martin Jaggi, et al. Scaling laws and compute-optimal training beyond fixed training durations. *Advances in Neural Information Processing Systems*, 37:76232–76264, 2024.

Bobby He, Lorenzo Noci, Daniele Paliotta, Imanol Schlag, and Thomas Hofmann. Understanding and minimising outlier features in neural network training. *arXiv preprint arXiv:2405.19279*, 2024.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022.

Shengding Hu, Yuge Tu, Xu Han, Chaoqun He, Ganqu Cui, Xiang Long, Zhi Zheng, Yewei Fang, Yuxiang Huang, Weilin Zhao, et al. MiniCPM: Unveiling the potential of small language models with scalable training strategies. *arXiv preprint arXiv:2404.06395*, 2024.

Hongzhi Huang, Defa Zhu, Banggu Wu, Yutao Zeng, Ya Wang, Qiyang Min, and Xun Zhou. Over-tokenized transformer: Vocabulary is generally worth scaling. *arXiv preprint arXiv:2501.16975*, 2025.

Stanislaw Jastrzebski, Zachary Kenton, Devansh Arpit, Nicolas Ballas, Asja Fischer, Yoshua Bengio, and Amos Storkey. Three factors influencing minima in SGD. *arXiv preprint arXiv:1711.04623*, 2017.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Tero Karras, Miika Aittala, Jaakko Lehtinen, Janne Hellsten, Timo Aila, and Samuli Laine. Analyzing and improving the training dynamics of diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24174–24184, 2024.

Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. *arXiv preprint arXiv:1609.04836*, 2016.

Jaeyeon Kim, Kulin Shah, Vasilis Kontonis, Sham M Kakade, and Sitan Chen. Train for the worst, plan for the best: Understanding token ordering in masked diffusions. In *Forty-second International Conference on Machine Learning*, 2025.

Kimi Team. Kimi K2: Open agentic intelligence. *arXiv preprint arXiv:2507.20534*, 2025.

Diederik Kingma and Ruiqi Gao. Understanding diffusion objectives as the elbo with simple data augmentation. *Advances in Neural Information Processing Systems*, 36:65484–65516, 2023.

Diederik Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. Variational diffusion models. *Advances in neural information processing systems*, 34:21696–21707, 2021.

Tianyi Li, Mingda Chen, Bowei Guo, and Zhiqiang Shen. A survey on diffusion language models. *arXiv preprint arXiv:2508.10875*, 2025.

Sadhika Malladi, Kaifeng Lyu, Abhishek Panigrahi, and Sanjeev Arora. On the SDEs and scaling rules for adaptive gradient algorithms. *Advances in Neural Information Processing Systems*, 35: 7697–7711, 2022.

Dominic Masters and Carlo Luschi. Revisiting small batch training for deep neural networks. *arXiv preprint arXiv:1804.07612*, 2018.

Sam McCandlish, Jared Kaplan, Dario Amodei, and OpenAI Dota Team. An empirical model of large-batch training. *arXiv preprint arXiv:1812.06162*, 2018.

William Merrill, Shane Arora, Dirk Groeneveld, and Hannaneh Hajishirzi. Critical batch size revisited: A simple empirical approach to large-batch language model training. *arXiv preprint arXiv:2505.23971*, 2025.

Muhammad Muzammal Naseer, Kanchana Ranasinghe, Salman H Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Intriguing properties of vision transformers. *Advances in Neural Information Processing Systems*, 34:23296–23308, 2021.

Shen Nie, Fengqi Zhu, Chao Du, Tianyu Pang, Qian Liu, Guangtao Zeng, Min Lin, and Chongxuan Li. Scaling up masked diffusion models on text. In *The Thirteenth International Conference on Learning Representations*, 2025.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4195–4205, 2023.

Project Apertus. Apertus: Democratizing open and compliant LLMs for global language environments, 2025. URL `https://github.com/swiss-ai/apertus-tech-report/blob/main/Apertus_Tech_Report.pdf`.

Subham Sekhar Sahoo, Zhihan Yang, Yash Akhauri, Johnna Liu, Deepansha Singh, Zhoujun Cheng, Zhengzhong Liu, Eric Xing, John Thickstun, and Arash Vahdat. Esoteric language models. *arXiv preprint arXiv:2506.01928*, 2025.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*, 2015.

Christopher J Shallue, Jaehoon Lee, Joseph Antognini, Jascha Sohl-Dickstein, Roy Frostig, and George E Dahl. Measuring the effects of data parallelism on neural network training. *Journal of Machine Learning Research*, 20(112):1–49, 2019.

Xian Shuai, Yiding Wang, Yimeng Wu, Xin Jiang, and Xiaozhe Ren. Scaling law for language models training considering batch size. *arXiv preprint arXiv:2412.01505*, 2024.

Samuel Smith, Erich Elsen, and Soham De. On the generalization benefit of noise in stochastic gradient descent. In *International Conference on Machine Learning*, pp. 9058–9067. PMLR, 2020.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pp. 2256–2265. pmlr, 2015.

Dan Su, Kezhi Kong, Ying Lin, Joseph Jennings, Brandon Norick, Markus Kliegl, Mostofa Patwary, Mohammad Shoeybi, and Bryan Catanzaro. Nemotron-CC: Transforming Common Crawl into a refined long-horizon pretraining dataset. *arXiv preprint arXiv:2412.02595*, 2024.

Mingjie Sun, Xinlei Chen, J Zico Kolter, and Zhuang Liu. Massive activations in large language models. *arXiv preprint arXiv:2402.17762*, 2024.

Sho Takase, Ryokan Ri, Shun Kiyono, and Takuya Kato. Large vocabulary size improves large language models. *arXiv preprint arXiv:2406.16508*, 2024.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. LLaMA: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Dimitri von Rütte, Janis Fluri, Yuhui Ding, Antonio Orvieto, Bernhard Schölkopf, and Thomas Hofmann. Generalized interpolating discrete diffusion. In *Forty-second International Conference on Machine Learning*, 2025.

Zirui Wu, Lin Zheng, Zhihui Xie, Jiacheng Ye, Jiahui Gao, Yansong Feng, Zhenguo Li, Victoria W., Guorui Zhou, and Lingpeng Kong. DreamOn: Diffusion language models for code infilling beyond fixed-size canvas, 2025. URL `https://hkunlp.github.io/blog/2025/dreamon`.

Greg Yang, Edward J Hu, Igor Babuschkin, Szymon Sidor, Xiaodong Liu, David Farhi, Nick Ryder, Jakub Pachocki, Weizhu Chen, and Jianfeng Gao. Tensor programs V: Tuning large neural networks via zero-shot hyperparameter transfer. *arXiv preprint arXiv:2203.03466*, 2022.

Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. Scaling vision transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12104–12113, 2022.

Biao Zhang and Rico Sennrich. Root mean square layer normalization. *Advances in neural information processing systems*, 32, 2019.

Guodong Zhang, Lala Li, Zachary Nado, James Martens, Sushant Sachdeva, George Dahl, Chris Shallue, and Roger B Grosse. Which algorithmic choices matter at which batch sizes? Insights from a noisy quadratic model. *Advances in neural information processing systems*, 32, 2019.

Hanlin Zhang, Depen Morwani, Nikhil Vyas, Jingfeng Wu, Difan Zou, Udaya Ghai, Dean Foster, and Sham Kakade. How does critical batch size scale in pre-training? *arXiv preprint arXiv:2410.21676*, 2024.

Liu Ziyin, Zhikang T Wang, and Masahito Ueda. LaProp: Separating momentum and adaptivity in Adam. *arXiv preprint arXiv:2002.04839*, 2020.

A Appendix

A.1 Relation Between Batch Size and Step Count

We given an example of the discovered hyperbolic relation between batch size and step count in Figure 5.

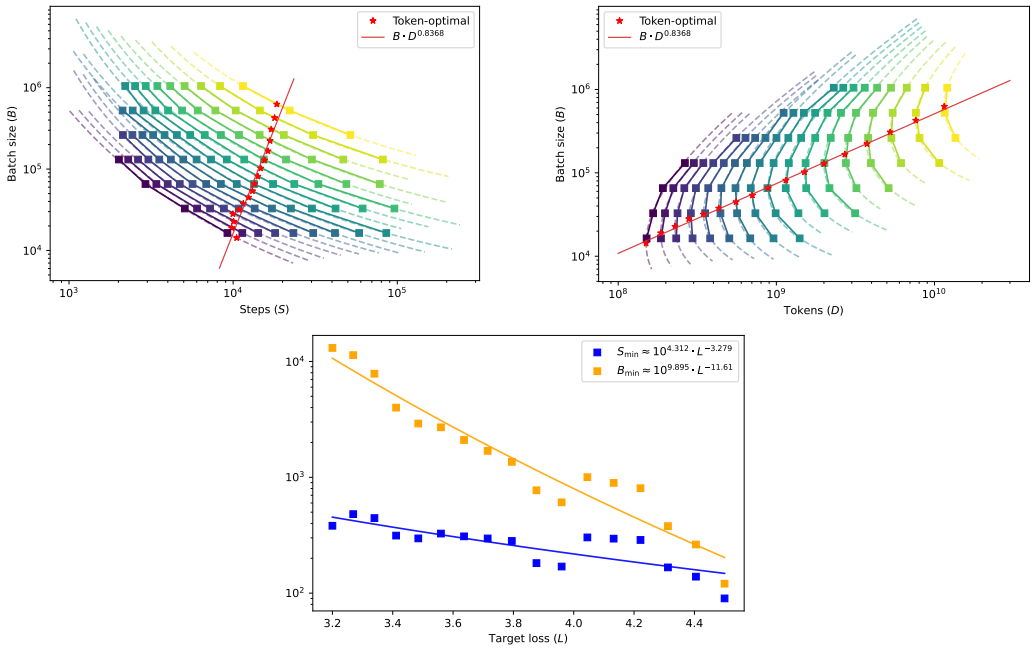


Figure 5: (top) There appears to be a tight relationship between batch sizes and step counts achieving the same loss, with ISO-loss curves following a hyperbolic relation. (bottom) The minimum batch size and step count, as per the asymptotes of the fitted hyperbolas, grow with what appears to be following a power law in the target loss, implying that as we get closer to some minimum achievable loss, the minimum required step count, but especially the minimum batch size, grow to large values. Here we display runs of a 85M (L12-D768) model trained on balanced hybrid noise.

A.2 Sweep Configuration

The exact configurations used for model sizes are given in Table 2 and the hyperparameter sweep settings are given in Table 3.

B Related Work

B.1 Scaling Laws

Scaling laws refer to a phenomenon where the performance of increasingly larger neural networks follows a predictable trend that usually takes the form of a power law in model parameters and dataset size (Kaplan et al., 2020; Hoffmann et al., 2022). While neural scaling laws have first been proposed in the context of language modeling (Kaplan et al., 2020; Hoffmann et al., 2022), they have since been observed across a variety of tasks, data modalities, and model architectures (Zhai et al., 2022; Bachmann et al., 2023; Peebles & Xie, 2023).

B.2 Critical Batch Size

Batch size is a hyperparameter that is often neglected when studying the scaling behavior of language models, with many studies fixing it to some constant value (Hoffmann et al., 2022; Hägele et al., 2024). This methodology is motivated by classical bounds in both convex and nonconvex smooth

Label	Params. P	Vocab. size V	Layers L	Hidden size d	Attn. heads H
L8-D512	25M	131,072	8	512	8
L10-D640	50M	131,072	10	640	10
L12-D768	85M	131,072	12	768	12
L16-D1024	200M	131,072	16	1024	16
L20-D1536	570M	131,072	20	1536	12

Table 2: Overview of the five different model sizes that were used in our experiments. Parameter counts refer to non-embedding parameters.

Parameter	Values
Noise type b	$\{-1000, -2, 0, 2, 1000\}$
Sequence length N	2048
LaProp β_1	0.9
LaProp β_2	0.99; 0.98 for $B \geq 256$
LaProp ϵ	$10^{-8} d^{-1} L^{-1}$ (CompleteP)
Init. std. $(\sigma_{\text{bulk}}, \sigma_{\text{aux}})$	$0.4 \cdot d^{-\frac{1}{2}}$, 0.02 (CompleteP)
Resid. multiplier	$4.0 \cdot L^{-1}$ (CompleteP)
Out. multiplier	512 (CompleteP)
Weight decay	0.0
LR warmup steps	2000
LR cooldown steps	0
Bulk LR η_{bulk}	$d^{-1} \cdot \eta_{\text{base}}$ (CompleteP)
Auxiliary LR η_{aux}	$0.02 \cdot \eta_{\text{base}}$
Param. precision	`bfloat16`
Activation precision	Manual mixed precision
Batch size B / base learning rate η_{base}	8 / $\{0.2, 0.3\}$
	16 / $\{0.2, 0.3, 0.5\}$
	32 / $\{0.2, 0.3, 0.5\}$
	64 / $\{0.3, 0.5, 1.0\}$
	128 / $\{0.5, 1.0\}$
	256 / $\{0.5, 1.0\}$
	512 / $\{0.5, 1.0, 2.0\}$

Table 3: List of key hyperparameters for our grid search. The parameters that are swept over are noise type, model size (Tab. 2), batch size, and learning rate.

stochastic optimization (Garrigos & Gower, 2023), and can be understood by analyzing the stationary distribution of stochastic gradient descent (SGD) on quadratic potentials (Jastrzebski et al., 2017) – at convergence, loss statistics in this setting are batch size invariant, once the learning rate is tuned correctly. This basic reasoning can be extended to adaptive methods (Zhang et al., 2019), and reveals useful scaling strategies for zero-shot adaptation of optimal learning rates as the batch size increases, see e.g. square root laws for Adam derived in (Malladi et al., 2022; Compagnoni et al., 2025). The well-known limitation of this analysis is that, as the batch size increases, the number of steps decreases, when training occurs at the same data budget. As such, it is unrealistic to assume, as the steps budget decreases, that optimization has reached stationarity, and hence utilizing larger batches can lead to diminishing returns. Though early works have assessed that indeed smaller batches are not *per-se* needed for strong generalization (Smith et al., 2020), it was also noted that assessing which batch size is *critical* (i.e., larger batches lead to diminishing returns) based purely on the number of network parameters is problematic (Shallue et al., 2019) as it is influenced, e.g., by the network width (Chen et al., 2018). Therefore, methods were developed (McCandlish et al., 2018) to empirically estimate the optimal batch size from online gradient statistics.

The method by McCandlish et al. (2018) has also been extended to autoregressive language models pre-training. However, large pre-training setups pose an additional challenge, as it is unrealistic to assume optimization is ever reaching a stationary distribution in standard scenarios. Hence, it is

natural to expect that the critical batch in this setting is a function of the token budget – an intuition which was recently validated empirically (Zhang et al., 2024; Merrill et al., 2025). Finally, while it is clear that standard theoretical considerations on the expected loss dynamics predict that, in early training ("curvature dominated", as detailed in (Zhang et al., 2019; Smith et al., 2020)), the optimal batch size is one (i.e. maximizing number of steps is always convenient), recent studies have suggested that there exists an even stronger notion of *optimal batch size*, where training both below and above a specific batch size is compute-inefficient (Hu et al., 2024; Shuai et al., 2024).