Diffusion Language Models are Super Data Learners

Jinjie Ni[1†], **Qian Liu**, **Longxu Dou**[2], **Chao Du**[2], **Zili Wang**[3], **Hang Yan**[4],
Tianyu Pang[2], **Michael Qizhe Shieh**[1]
[1]National University of Singapore, [2]Sea AI Lab, [3]StepFun, [4]Shanghai Qiji Zhifeng Co., Ltd.

As high-quality tokens become the bottleneck for scaling, we ask which paradigm extracts more signal per *unique* token: autoregressive (AR) or diffusion language models (DLMs). Under controlled pre-training with matched total-token budgets, we observe an *Intelligence Crossover*: when unique data is limited, masked (absorbing) DLMs consistently surpass size-matched AR baselines. Quantitatively, DLMs deliver $> 3\times$ effective data efficiency: a DLM trained on 0.5B unique tokens matches a converged AR model trained on ~ 1.5B unique tokens. The crossover shifts *later* with more or higher-quality data, *earlier* with larger models, and persists across dense and MoE architectures. We attribute the gains to three compounding factors: (i) any-order modeling, (ii) *super-dense* compute from iterative bidirectional denoising, and (iii) built-in Monte Carlo augmentation; input/parameter noise improves AR under scarcity but cannot close the gap. At scale, a 1.7B DLM trained with a ~ 1.5T-token compute budget on 10B unique Python tokens overtakes a matched AR coder on standard benchmarks. We also show that rising validation cross-entropy need not imply degraded downstream accuracy in this regime. The trade-off is compute: diffusion spends more FLOPs to extract more from repeated data. Our results suggest that in the data-bound era, diffusion language models are *super data learners*.

1. Introduction

Autoregressive (AR) language models have been the default recipe for modern LLMs: causal factorization, teacher forcing delivering high signal-to-FLOPs training and efficient model serving (Brown et al., 2020; Raffel et al., 2020). Yet the regime in which AR thrived—ever-growing curated corpora paired with efficient compute utilization—is shifting. High-quality data is becoming the primary bottleneck, while compute continues to scale (Common Crawl, 2025; Sevilla and Roldán, 2024). This raises a central question for the next phase of scaling: *which modeling paradigm extracts more signal per **unique** token when data, not FLOPs, is the constraint?*

We revisit a strong modeling paradigm–diffusion language models (DLMs) with masked/absorbing discrete diffusion (Lou et al., 2023; Nie et al., 2025; Ou et al., 2024; Shi et al., 2024). DLMs exhibit several modeling advantages over AR models. Such as any-order modeling, on-the-fly context modification, multi-token generation, high compute-to-parameter ratio, etc. In this work, we focus on their data potential, studying DLMs and AR models under a common pre-training regime and ask how far each can go when unique data is scarce but repetitions are allowed.

The main empirical finding is an *Intelligence Crossover*: when total training tokens are fixed but the number of unique tokens is limited, DLMs consistently surpass equally sized AR counterparts (Figure 1). This crossover is not an isolated artifact—it systematically shifts with core factors. With more unique data, it shifts *later*; with higher data quality, it shifts *later*; with larger models, the crossover arrives *earlier*; and it persists across dense and sparse (MoE) architectures (Figures 2, 3, 4). Quantitatively, we observe >3× higher effective data efficiency for DLMs: a DLM trained on 0.5B unique tokens matches a converged AR model trained on ~ 1.5B unique tokens, under identical total-token budgets. Under compute-bound settings with abundant *unique* data, AR recovers its edge

[†]Correspondence to: Jinjie Ni <jinjieni@nus.edu.sg>
This is an initial draft that will be further improved.

by fitting the distribution more efficiently; but in data-bound regimes—our focus and, increasingly, the practical reality—DLMs are *super data learners*.

Why does the crossover occur? Our analysis isolate three contributors. (i) **Any-order modeling** relaxes AR's causal inductive bias, enlarging the hypothesis space that a model can fit from the same corpus. (ii) **Super-density**: DLMs use substantially more FLOPs at train and test time (temporal refinement with bidirectional attention), enabling them to mine repeated data more thoroughly. (iii) **Richer Monte Carlo augmentation**: the diffusion objective explicitly averages over corruption patterns, turning each sequence into many informative variants. Injecting noise into AR inputs or parameters (dropout) does help under data scarcity, but cannot close the gap (Figures 5 and 6). We also clarify a common diagnostics pitfall: rising validation cross-entropy ("overfitting") does not necessarily imply degraded downstream accuracy (Figures 8, 9). Note that DLMs do overfit with enough epochs and small enough unique data, but later than AR, and they extract markedly more value before saturation; e.g., a 1B DLM trained for 480 epochs on 1B tokens reaches $\sim$ 56% HellaSwag and $\sim$ 33% MMLU without clear saturation (Figure 10).

To assess the data-efficiency advantages of DLMs in realistic large-scale settings, we trained two 1.7B-parameter models with a total budget of 1.5T tokens' compute. Models are trained on 10B unique Python tokens for $\approx$150 epochs, where 10B is an amount representative of practical high-quality data availability for certain programming languages. We observe clear performance crossovers on coding benchmarks, with the resulting diffusion coder achieving parity with state-of-the-art AR code models trained on trillions of unique tokens.

Key takeaways.

- **Intelligence Crossover.** Under limited unique data, DLMs surpass AR models of the same size; the crossover is robust across data budgets, model scales, and sparsity choices. We also ran 2 scaled-up runs up to 1.5 trillion tokens to verify the universal appearance of the crossover phenomenon.
- **Data potential.** DLMs achieve >3× effective data efficiency in our setting (e.g., 0.5B unique tokens for DLM $\approx$ 1.5B for AR), making them strong candidates when data, not FLOPs, is scarce.
- **Factors driving the gains.** Any-order modeling + super-dense compute + built-in noisy augmentation are jointly responsible; AR with input/parameter noise helps but cannot match DLMs.
- **Scaling trends.** More unique data moves the crossover later; larger models move it earlier; higher data quality moves it later. Parameter- or FLOPs-matched Dense > MoE in data-bound AR, while diffusion benefits consistently from scale regardless of sparsity.
- **Caveats.** Rising validation loss need not imply worse downstream performance—relative likelihood gaps can keep improving even as absolute NLL increases.
- **Compute trade-off.** DLMs require more training and (parallelizable) inference FLOPs to reach their potential; when unique data is abundant, compute-efficient AR can still win.

Together, these results sharpen a forecast for the data-bound era: if high-quality tokens remain the scarcest resource, masked diffusion models are a compelling modeling paradigm for pushing frontier capability per unit of *unique* data, even at the cost of greater compute.

2. Preliminaries

2.1. Autoregressive Language Models

Autoregressive (AR) language modeling is the mainstream modeling scheme in state-of-the-art LLMs. It parameterizes the joint distribution over a tokenized sequence $x_{1:T} \in \{0, \ldots, K-1\}^T$ via the chain rule,

$$p_\theta(x_{1:T}) = \prod_{i=1}^{T} p_\theta(x_i \mid x_{<i}) \,. \tag{1}$$

Modern decoder-only LLMs realize this factorization with stacks of causal self-attention blocks (Radford et al., 2019): a triangular (causal) mask limits each position's receptive field to the prefix $x_{<i}$, allowing rich conditioning within the visible context while preserving left-to-right generation. Training uses teacher forcing—shifting the sequence so that all next-token conditionals are learned in parallel—whereas inference proceeds sequentially with KV-caching for efficiency. This recipe yields exact, normalized likelihoods; supports streaming generation and controllable decoding; and under sufficient scale enables strong in-context learning behaviors (Brown et al., 2020).

Learning objective AR training maximizes the log-likelihood (equivalently, minimizes cross-entropy) under the data distribution:

$$\mathcal{L}(\theta) = \mathbb{E}_{x_{1:T} \sim p_{\text{data}}} \left[\sum_{i=1}^{T} -\log p_\theta(x_i \mid x_{<i}) \right]. \tag{2}$$

This objective computes in a single forward pass by predicting the next token at every position. While AR models are inherently one-directional and can suffer exposure bias from teacher forcing, their simplicity, exact normalization, and scalability make decoder-only Transformers the dominant foundation for high-quality text generation.

2.2. Masked Diffusion Language Models

Why masked diffusion? DLMs adopt a noising–denoising framework over sequences. Among their variants, *masked diffusion*—also known as absorbing discrete diffusion, which relies on an absorbing transition kernel—has emerged as the most effective formulation (Amin et al., 2025). Their bidirectional attention and diffusion objective enable any-order modeling, allowing data to be modeled in arbitrary directions during both training and inference. This property is particularly beneficial for tasks requiring non-causal dependencies and back-and-forth reasoning, such as coding (Wu et al., 2025; Xie et al., 2025), mathematics (Google DeepMind, 2025), and report generation (Han et al., 2025). DLMs' bidirectional attention natively support on-the-fly context modification as new content is generated, a desirable feature in these tasks. Multi-token generation is also naively supported by DLMs, providing the foundation for their bleeding fast decoding. Moreover, DLMs spend more parallelable FLOPs at both the inference and training time, leading to its superior data learning capability and potentially stronger reasoning capabilities.

Forward (corruption) process Let K be the vocabulary size, L the sequence length, and m the mask token. Given a clean sequence $x_0 \in \{0, \ldots, K-1\}^L$, define a monotone diffusion schedule $\alpha_t \in [0, 1]$ with $\alpha_0 = 1$ and $\alpha_1 = 0$, where α_t is the probability that a token is *clean* (unmasked) at noise level

$t \in [0,1]$. The forward process independently masks tokens:

$$q_{t|0}(x_t \mid x_0) = \prod_{i=1}^{L} q_{t|0}\left(x_t^{(i)} \mid x_0^{(i)}\right), \qquad q_{t|0}\left(x_t^{(i)} \mid x_0^{(i)}\right) = \begin{cases} \alpha_t, & x_t^{(i)} = x_0^{(i)}, \\ 1-\alpha_t, & x_t^{(i)} = m\,, \end{cases}$$

so that the expected unmasked fraction at level t equals α_t.

Reverse (denoising) process Starting from the fully masked sequence x_1 and a decreasing schedule $1 = t_0 > t_1 > \cdots > t_N = 0$, the reverse dynamics from t to $s < t$ acts independently across positions:

$$q_{s|t}\left(x_s^{(i)} \mid x_t\right) = \begin{cases} 1, & x_t^{(i)} \neq m,\ x_s^{(i)} = x_t^{(i)}, \\ \dfrac{1-\alpha_s}{1-\alpha_t}, & x_t^{(i)} = m,\ x_s^{(i)} = m, \\ \dfrac{\alpha_s - \alpha_t}{1-\alpha_t}\, q_{0|t}\left(x_s^{(i)} \mid x_t\right), & x_t^{(i)} = m,\ x_s^{(i)} \in \mathcal{V} \setminus \{m\}, \\ 0, & \text{otherwise.} \end{cases}$$

i.e., already-revealed tokens stay fixed; masked tokens either remain masked with probability $\frac{1-\alpha_s}{1-\alpha_t}$ or are revealed by sampling from a *data-prediction* distribution $q_{0|t}(\cdot \mid x_t)$ with probability $\frac{\alpha_s-\alpha_t}{1-\alpha_t}$. A key *time-agnostic* property (Ou et al., 2024) of masked diffusion is that

$$q_{0|t}\left(x_0^{(i)} \mid x_t\right) = p_{\text{data}}\left(x_0^{(i)} \,\middle|\, x_t^{\text{UM}}\right),$$

the conditional distribution of the clean token depends only on the *unmasked* context x_t^{UM}; it does not depend on t beyond which tokens are visible. This allows the denoiser to be parameterized without an explicit time embedding.

Learning objective Let $p_\theta\left(x_0^{(i)} \mid x_t\right)$ approximate $p_{\text{data}}\left(x_0^{(i)} \mid x_t^{\text{UM}}\right)$. Masked diffusion maximizes a variational bound on $\log p_\theta(x_0)$, which can be written as minimizing

$$\mathcal{L} \;=\; \int_0^1 w(t;\alpha)\, \mathbb{E}_{q_{t|0}(x_t|x_0)} \left[\sum_{i:\, x_t^{(i)} = m} -\log p_\theta\left(x_0^{(i)} \mid x_t\right) \right] \mathrm{d}t, \tag{3}$$

where the importance weight $w(t;\alpha)$ depends only on the schedule and, up to a constant factor, takes the natural form

$$w(t;\alpha) \;=\; \frac{\alpha_t'}{\alpha_t - 1}\,.$$

Intuitively, $w(t;\alpha)$ compensates for the varying expected number of masked positions across noise levels. For the widely used linear schedule $\alpha_t = 1 - t$, this reduces to the familiar integrand weight $w(t) = 1/t$.

3. The Intelligence Crossover

We present extensive results showing that, when trained on standard web tokens, masked DLMs consistently outperform AR counterparts across model sizes in data-constrained settings, achieving higher potential without performance saturation. To analyze this pattern more rigorously, we decompose its key factors and conduct controlled group experiments.

3.1. Experimental Settings

Unless specifically mentioned, most of the below sections use the same basic settings. It is worth noting that the hyperparameters adopted in our experiments are primarily optimized for AR models, reflecting extensive prior tuning efforts by the broader LLM research community. Although we aimed to maintain identical settings across AR and diffusion models, this is inherently unfair for diffusion models. Consequently, the observed performance advantages of diffusion models could be under-estimate.

All training runs were conducted using a significantly modified Megatron-LM codebase. Cross-over experiments were trained on a subset of the Nemotron-CC corpus (Su et al., 2024); the runs in Figure 11 utilized a subset of the c4-en corpus (Raffel et al., 2020); the coders are trained on a subset of the RefinedCode (Huang et al., 2024). Note that all token budgets used are randomly sampled from these corpus, without any special process. We used the same masked diffusion objective as in Nie et al. (2025), detailed in Equation 3. Specifically, we employed a batch size of 256, a sequence length of 2048, and a warmup-stable-decay (WSD) learning rate schedule peaking at 2e-4 with 1000 warmup steps, followed by a 10% exponential decay to 2e-5. Model parameters were randomly initialized from a normal distribution with s.t.d. 0.02. We adopted a performant architectural configuration, incorporating the GPT-2 tokenizer, RoPE, SwiGLU, pre-layer RMSNorm, bias-free, and qk normalization. All mixture-of-expert models in this work use token-choice routing with 1e-2 auxiliary loss and 1e-3 z loss (Zoph et al., 2022). Validation loss was evaluated on the c4-en validation set using distinct 100M-token subsets per evaluation. Benchmark evaluations strictly adhered to official protocols, detailed in Table 1.

3.2. Data Budget Decides the Crossover Timing

The unique data budget is the first key factor we ablate under data-constrained training. Here, we vary the number of unique tokens from 0.5B to 96B while fixing the total training tokens at 96B, constrained by computational resources, and keep the model size fixed at 1B. Notably, 0.5B is a realistic regime, as many domain-specific datasets—such as robotics logs, clinical records, and low-resource language corpora—often fall within this scale.

As shown in Figure 1, we consistently observe crossover behavior: diffusion language models surpass their autoregressive counterparts at low data budgets. Our results indicate that DLMs exhibit more than a threefold higher effective data efficiency compared to AR models. Empirically, a DLM trained on only 0.5B unique tokens (not fully converged) achieves comparable performance to an AR model trained on 1.5B unique tokens (converged).

Under compute-bound settings—where data is abundant—AR models fit the training distribution more effectively and achieve stronger end-of-training performance. In contrast, under data-bound conditions—reflecting today's reality where compute growth outpaces high-quality data availability—diffusion models eventually outperform AR models. Interestingly, the crossover timing is similar across both evaluation benchmarks. As the number of unique tokens increases, the crossover shifts later (0.5–1.5B) or beyond our observable range (1.5–96B). This delay is more clearly highlighted in §6.

We also aggregated all runs into a single plot for global comparison. Strikingly, DLMs exhibit minimal degradation across both benchmarks and the validation set even when the unique data budget is reduced from 96B to 0.5B tokens—demonstrating substantially higher data efficiency than typically assumed. The narrow gap between the 10B and 96B AR runs reflects the fact that crossover points are pushed further out, meaning that what we observe within the 0–96B range captures only the early phase, where differences remain modest. This also suggests that, despite being more

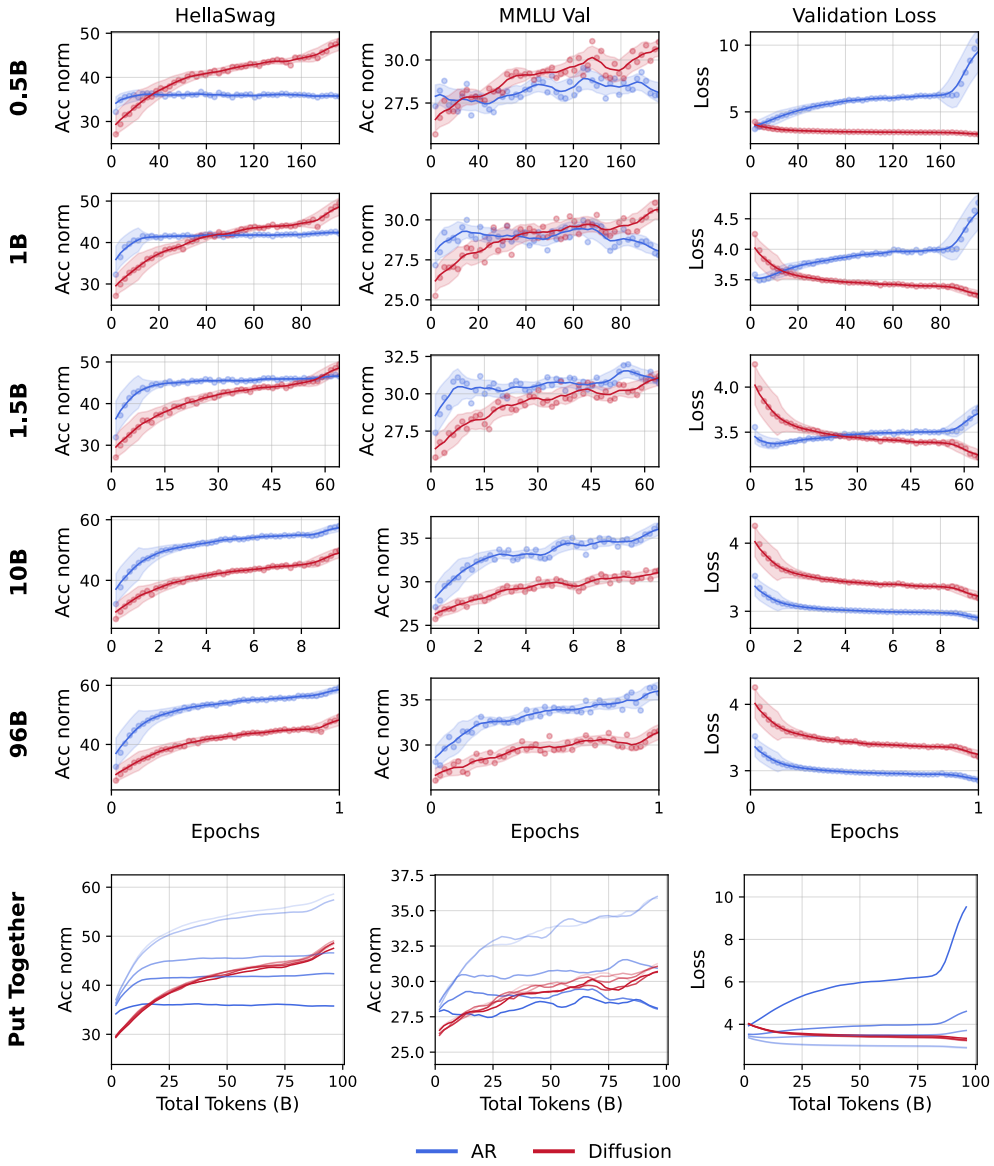

Figure 1 | **Diffusion vs. AR with various unique data budgets.** All models are trained on 96B total tokens (including repetition), varying the unique tokens from 0.5B to 96B. In the bottom "Put Together" row, a darker color means a smaller unique data size, i.e., trained for more epochs.

sensitive to data repetition, current AR training pipelines may underutilize available data: repeating the given 10B tokens for 10 epochs leads to only mild short-term degradation, more than the 4 epochs empirically observed in Muennighoff et al. (2023).

3.3. Varying Data Quality

Unique data size is not the only factor influencing scaling behavior; data quality is equally critical. Prior studies show that quality can even outweigh quantity, i.e., "less is more" in LLM training (Magnusson et al., 2025; Zhou et al., 2023). To isolate this effect, we conduct controlled experiments on datasets of varying quality. Specifically, we train 1B-parameter AR and diffusion models on three quality tiers—low, medium, and high—sampled from the same distribution (Su et al., 2024), using 1B unique tokens over 96 epochs.

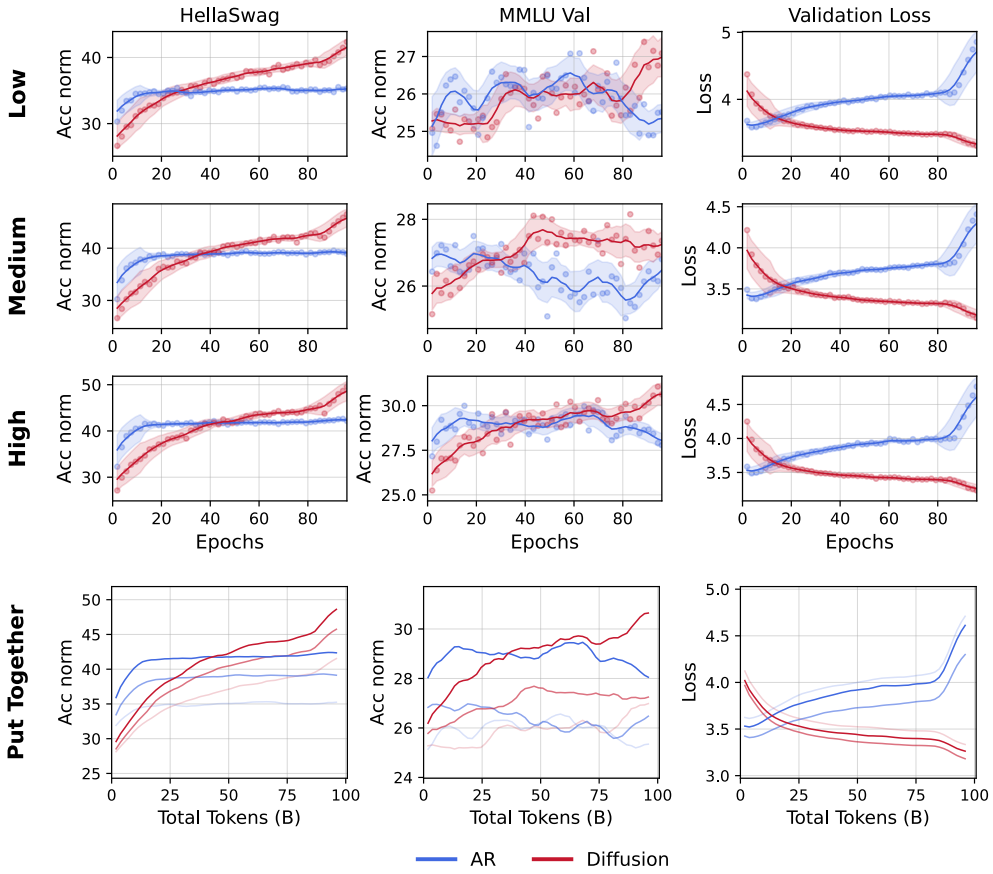

Figure 2 | **Diffusion vs. AR with various data qualities.** All models are trained on 1B unique tokens for 96 epochs. In the bottom "Put Together" row, a darker color means a higher data quality.

One might expect the crossover to amplify with higher data quality, since diffusion models typically outperform AR models in data-limited regimes. Our ablation shows that both diffusion and AR models benefit substantially from improved data quality across benchmarks (Figure 2). However, as quality increases, the crossover shifts slightly later, suggesting that AR models are more sensitive to quality variation. Notably, validation loss trends diverge from benchmark results: the medium-quality run achieves the lowest validation loss. This highlights that validation loss is not always a reliable comparator, as model overconfidence can heavily distort cross-entropy. We expand on this in §5.

3.4. How Much Does Model Size Impact the Data-Constrained Training?

According to compute- and data-constrained scaling laws (Hoffmann et al., 2022; Muennighoff et al., 2023), model size should grow with the data budget and training epochs. We investigate whether both model families benefit from increased scale and how the crossover evolves. To this end, we train diffusion and AR models from 1B to 8B parameters on 1B unique tokens for 96 epochs, keeping all other settings identical (§3.1).

As shown in Figure 3, the crossover shifts consistently earlier with larger models. This arises because, under data constraints, AR models quickly saturate the available data, and further scaling not only yields diminishing returns but also increases overfitting (see aggregated plots, bottom). In contrast, diffusion models do not fully exploit the 96B-token budget even at 1B parameters; scaling them accelerates learning and consistently improves performance. Remarkably, under these constraints, even the smallest diffusion model outperforms AR models at all sizes.

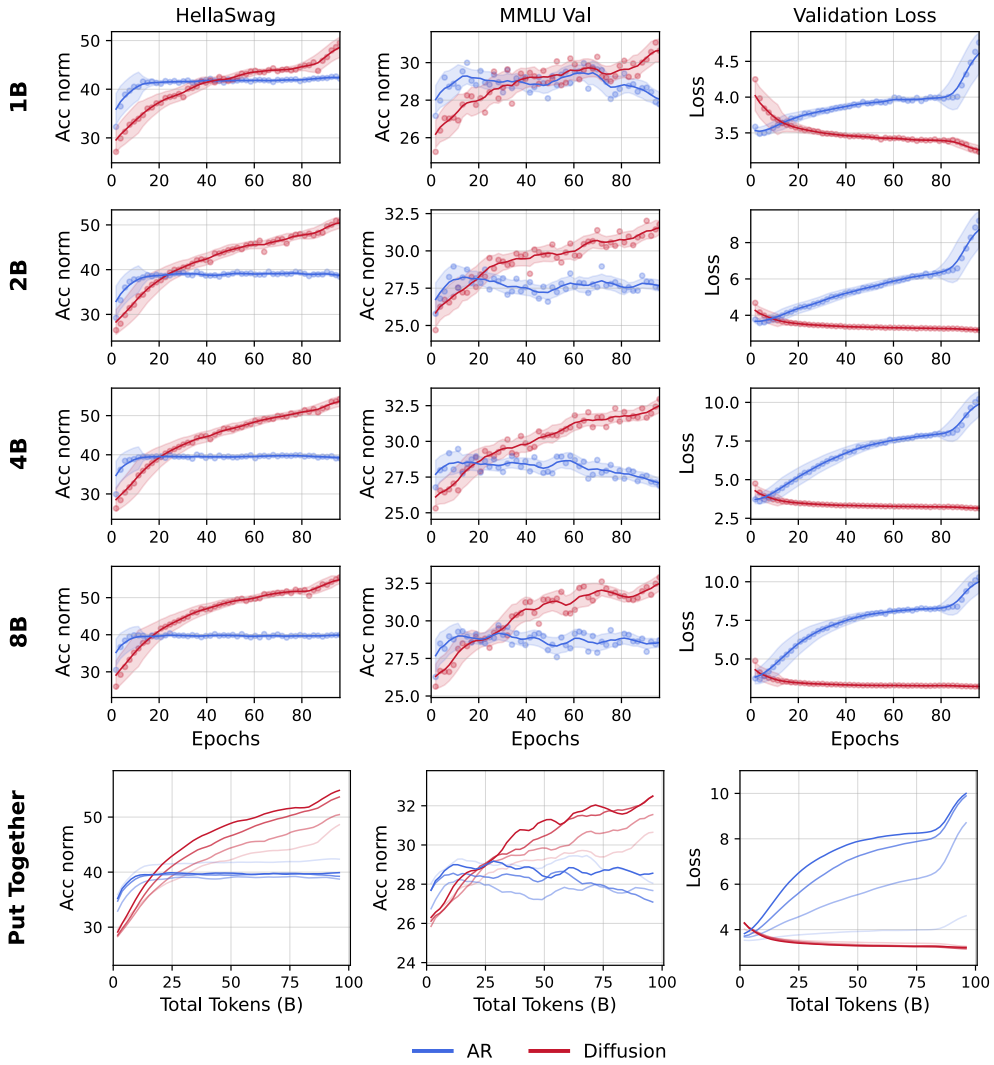

Figure 3 | **Diffusion vs. AR with various model sizes ranging from 1B to 8B.** All models are trained on 1B unique tokens for 96 epochs. In the bottom "Put Together" row, a darker color means a larger model size.

Conceptually, diffusion models learn any-order mappings: an input sequence of length L can be corrupted into 2^L variations, whereas AR models only learn causal mappings with L sequence prefixes. This combinatorial expansion makes the diffusion learning space—and thus its upper performance bound—much larger than AR under limited data. While not every corruption yields distinct signal, the enlarged hypothesis space requires larger models to fully capture. We discuss this further in §7.

3.5. Sparse, Dense, Super-Density

A key distinction between AR and diffusion models lies in compute efficiency. Compared to dense AR models, diffusion models require substantially more FLOPs during training to realize the full data potential, and far higher parallel FLOPs during inference to scale along the time axis. We term these diffusion-based dense models "super-dense" architectures. Further analysis of training- and test-time compute trade-offs is provided in §7. Building on this, a natural question arises: how do dense and super-dense models compare against sparse architectures, and does sparsity alter performance under data-constrained regimes?

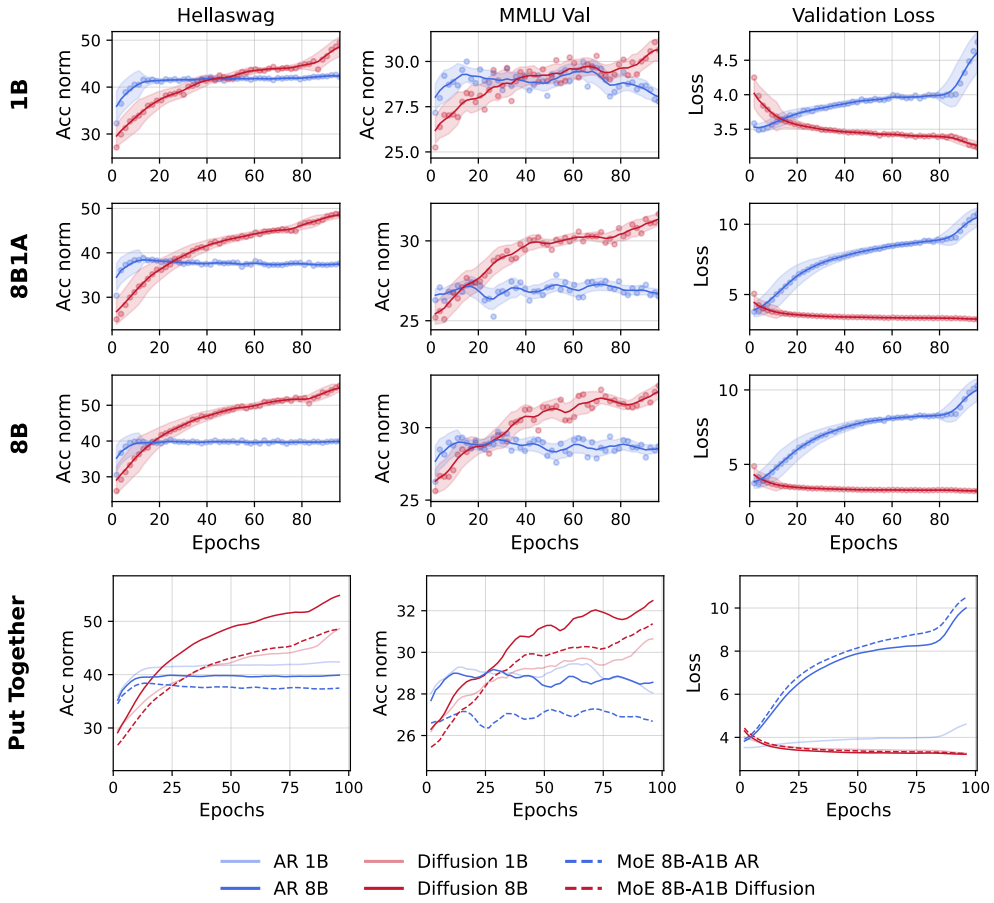

Figure 4 | **Diffusion vs. AR with various model sparsities.** All models are trained on 1B unique tokens for 96 epochs. Both 8B-A1B and 8B1A denote 8B total parameters and 1B activated parameters, parameter-matching the 8B dense and FLOPs-matching the 1B dense.

To study the effect of sparsity in data-constrained learning, we train Mixture-of-Experts (MoEs) (Shazeer et al., 2017) for both AR and diffusion models, using 8B total parameters with 1B active per forward pass (8B1A). For direct comparison, we also train 1B dense models to match the activated parameters (FLOPs-matching) and 8B dense models to match the total parameters (parameter-matching). The density ordering is: AR MoE < AR dense < DLM MoE < DLM dense. All models are trained on 1B unique tokens for 96 epochs.

Figure 4 reveals several key observations. First, across all sparsity levels, DLMs consistently surpass AR models, with crossover timing ordered as 8B dense < 8B1A MoE < 1B dense. Second, when comparing dense and sparse variants, the evaluation can be framed in two complementary ways: (1) under FLOPs-matching, how much benefit arises solely from increasing total parameters? (2) under parameter-matching, what is the performance cost of reducing per-task FLOPs?

Given the distinct behaviors of AR and diffusion models under data constraints, we analyze them separately. For AR models, which fit the data within a few epochs and then saturate (or overfit on the validation set), scaling from 1B to 8B parameters consistently degrades performance across evaluations, regardless of dense or sparse expansion. Sparse expansion (1B dense → 8B1A MoE) performs worst. Alternatively, framing the comparison as FLOP reductions from the 8B dense baseline reveals that reducing only FLOPs (8B dense → 8B1A MoE) hurts performance, while reducing both FLOPs and parameters (8B dense → 1B dense) improves it. Together, these perspectives lead to

a clear conclusion: **in data-constrained settings, given fixed parameter counts, higher FLOPs consistently improve performance**, as evidenced by 8B dense consistently outperforming 8B1A MoE.

Diffusion models behave differently: within the 96B-token window, they have not yet reached diminishing returns and remain far from overfitting (though §6 shows eventual overfitting). In this regime, the 8B1A diffusion MoE performs as expected, generally between the 8B dense and 1B dense counterparts, with its advantage most pronounced on knowledge-heavy benchmarks (e.g., MMLU). We believe the interplay of compute and parameterization under data constraints deserves a dedicated study.

3.6. Is Noise Deciding the Game?

In §7, we further analyze the factors underlying the advantage of DLMs over AR models. In summary, DLMs differ from AR models in three ways: (1) any-order modeling, (2) higher training- and inference-time FLOPs ("super-density"), and (3) richer Monte Carlo sampling via noisy data augmentation. We argue that noisy augmentation primarily benefits models in data-constrained regimes, whereas the first two constitute fundamental modeling advantages with universal impact. Ablating these factors is non-trivial: introducing any-order modeling into AR while holding other variables fixed is hard, and similarly, scaling AR FLOPs without altering other dynamics is difficult. Moreover, note that masked DLMs can be viewed as any-order models with more practical tractability.

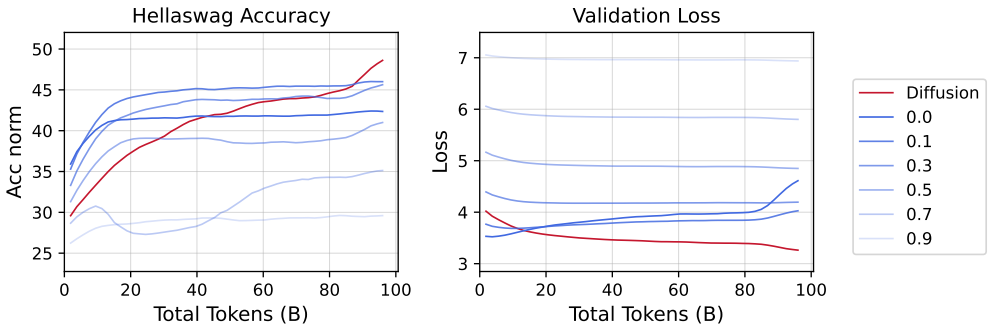

Figure 5 | **Injecting noise (random masking) to the AR model's inputs helps in data-constrained settings but does not beat diffusion models.** All models shown in this figure are 1B parameters trained on 1B unique tokens for 96 epochs.

We therefore focus on ablating the data augmentation feature to isolate the contributions of the other two factors. A direct approach is to mimic masked DLMs by injecting noise into AR inputs—masking a fraction of tokens—to evaluate the potential gains. While uncommon in large-scale LLM training, noisy data augmentation has precedent in computer vision. As shown in Figure 5, we apply input masking at ratios from 10% to 90%. Results show that AR performance improves with modest noise but collapses once inputs become overly corrupted. The substantial gains under low-noise regimes confirm that noisy data augmentation is indeed a key contributor to diffusion's advantage in data-constrained settings. However, even the best configuration (10% masking) falls well short of diffusion, and saturates by the end of the 96B-token training window, whereas diffusion remains far from saturation and would likely widen the gap with longer training.

Alternative to injecting noise to the input for noisy data augmentation, we can also inject noise to the parameters by zeroing out a random set of neuron outputs, a.k.a., dropout, to achieve similar effects. Similarly, we use a dropout ratio from 10% to 90%. As shown in Figure 6, adding noise to the parameter also raises the performance of AR models in data-constrained settings while not fully

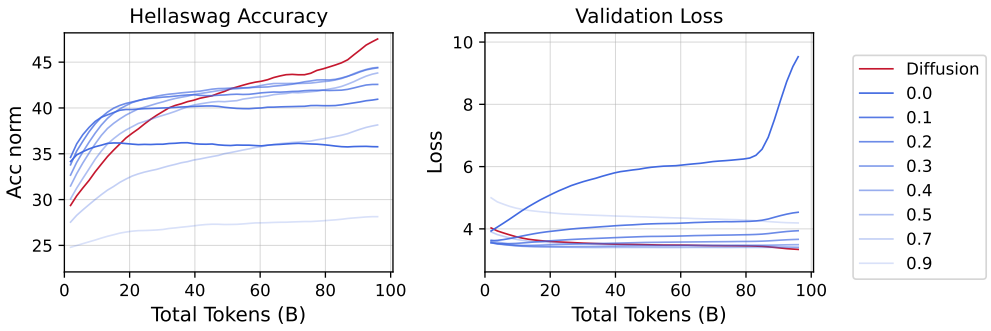

Figure 6 | **Injecting noise to AR model's parameters (dropout) helps in data-constrained settings but does not beat diffusion models.** All models shown in this figure are 1B parameters trained on 1B unique tokens for 96 epochs.

eliminating the advantage of DLMs. The AR diminishes likewise but diffusion doesn't.

4. Scaling Crossovers to Trillion-Level Total Tokens

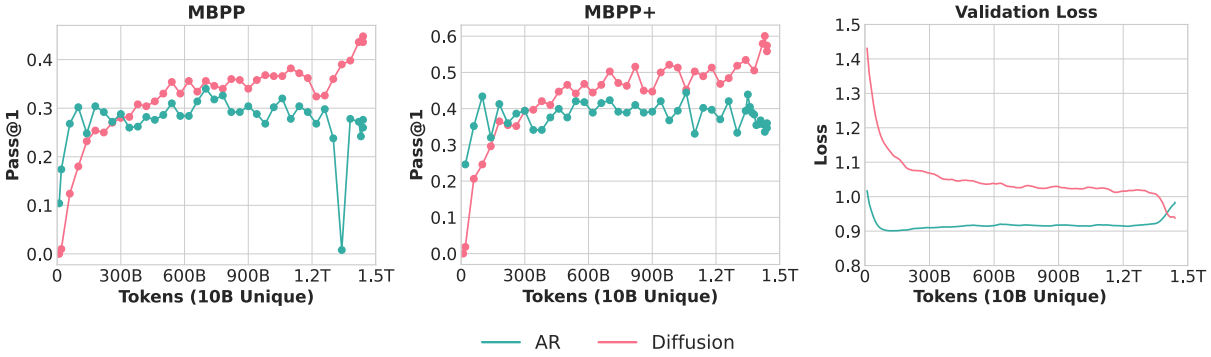

Figure 7 | **1.7B AR vs. DLM trained on 10B unique code tokens for ≈150 epochs.** On downstream evaluations, we observe early crossovers where DLMs surpass AR models. This provides another evidence that crossovers emerge at larger scales under limited unique data. Different crossover timings are observed on two additional coding benchmarks (Figure 13).

We further validate the crossover phenomenon under larger unique token budgets and on generative tasks. Among such tasks, code generation provides a particularly clean and popular benchmark, while also being heavily constrained by available training tokens. Prior work suggests that a 10B-token budget is already a practical scale for certain programming languages (Huang et al., 2024). To this end, we construct a 10B-token dataset by randomly sampling from RefineCode (Huang et al., 2024), consisting of 9B unique Python tokens and 1B annealing tokens, ensuring a strictly non-repetitive training corpus. On this dataset, we train a pair of 1.7B-parameter AR and DLM models under identical settings for ≈150 epochs. The training setup employs a warmup–stable–decay learning rate schedule (2000 warmup steps, followed by decay over 100B tokens), resulting in a total compute budget of ≈1.5T tokens. Additional hyperparameters include a sequence length of 4096, a global batch size of 1024, weight decay of 0.1, and a performant architecture: GPT-NeoX (Black et al., 2022) tokenizer, rotary position embeddings (RoPE), SwiGLU activations, pre-layer RMSNorm, bias-free layers, and qk normalization.

Interestingly, we observe a very clear crossover in the early stages of training across downstream

benchmarks, further reinforcing the apparent universality of the diffusion–AR crossover phenomenon. At the same time, we note that the DLM had not yet converged by the end of the 1.5T-token training cycle, suggesting substantial untapped potential if training were to continue. Beyond standard benchmarks, we evaluate both models on HumanEval and HumanEval+, and find that the crossover occurs at a different point compared to MBPP and MBPP+. Specifically, in HumanEval(-plus), the performance curves intersect only near the end of the annealing stage (see Figure 13), without seeing diminish returns. We hypothesize that this timing discrepancy is rooted in the evaluation protocols: MBPP and MBPP+ adopt a 3-shot setting, while HumanEval and HumanEval+ use a 0-shot setting. Such differences in evaluation configuration likely amplify or suppress model capabilities at different stages of training, thereby shifting crossover points.

These results highlight two key insights. First, the crossover phenomenon extends robustly to generative tasks, under more unique token budgets. Second, the precise timing of crossovers in generative tasks may be sensitive to evaluation protocols, which raises an important methodological consideration: the need for systematic studies that disentangle training dynamics from evaluation artifacts. We believe a more rigorous examination of evaluation settings will be critical for fully characterizing crossover behavior in future work.

5. High Validation Loss ≠ Degraded Intelligence

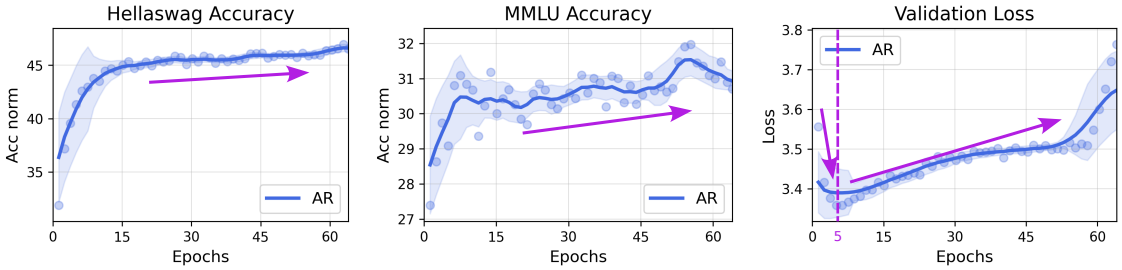

Figure 8 | When models get “overfit” on pre-training validation sets, their performance on down-stream evaluations doesn’t necessarily drop, and may keep improving till the end of training.

We observe that the AR models exhibiting signs of "overfitting"—indicated by an increase in validation loss—continue to improve on downstream tasks, as illustrated in Figures 8 and the previous ones. This phenomenon arises because validation loss is measured as an absolute Cross-Entropy loss (Negative Log-Likelihood, NLL), whereas accuracy on multiple-choice benchmarks depends on comparing the relative Cross-Entropy losses across options. Consequently, changes in absolute NLL values do not necessarily translate into changes in their relative ordering.

In Figure 9, we visualize the average negative log-likelihood (NLL) for the ground-truth and alternative options across multiple-choice evaluations, along with their respective differences (ΔNLL), during the pre-training of a 1B-parameter autoregressive model over 1.5B unique tokens for 64 epochs. Notably, even at the first validation checkpoint (after 3,600 training steps), the model already exhibits substantially lower NLL (higher likelihood) on the ground-truth options, indicating an early capacity to preferentially assign higher logits to correct choices. As training continues, the model begins to overfit, causing an increase in NLL values for both ground-truth and incorrect options. Interestingly, even after this "overfitting," the gap between ground-truth and alternative NLLs continues to widen consistently, indicating that the model’s underlying discriminative ability continues to improve despite the rise in validation loss. This phenomenon persists for both in-domain and out-of-domain training data.

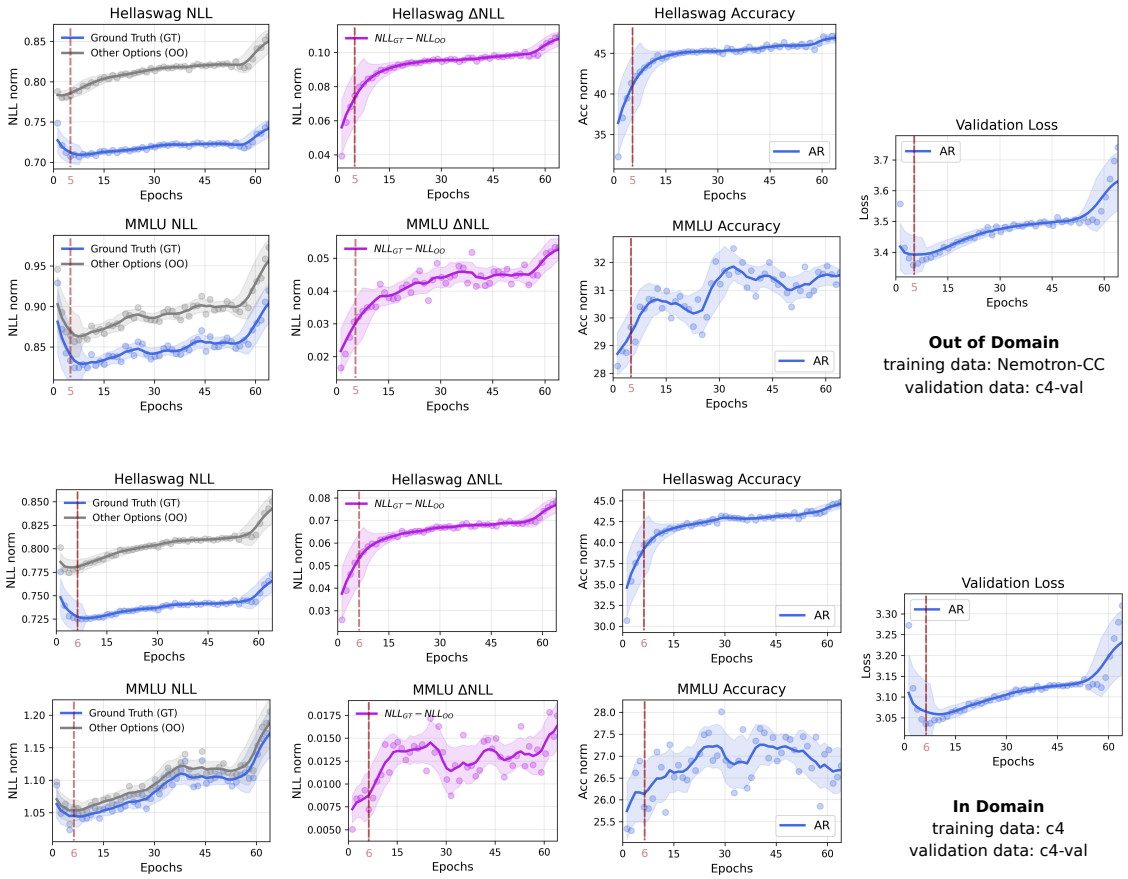

Figure 9 | **An illustration of why the models' performance keeps growing after they overfit on pre-training validation sets (indicated with dashed line).** NLL: Negative log-likelihood on the ground-truth and other options of multiple-choice evals (NLLs on other options are averaged). ΔNLL: The differences between the NLLs on ground-truth and other options, which keeps growing. This is a 1B autoregressive model trained on 1.5B unique tokens, 64 epochs, on both out-of-domain and in-domain pre-training data.

One plausible explanation is that repeated exposure to a limited set of training data causes the model to become excessively confident on certain text segments, amplifying NLL values for incorrect predictions. Nevertheless, the persistent growth in relative NLL differences between ground-truth and other options reflects continuous improvement in the model's discriminative power. A similar rationale applies to generative evaluations, where choices are made at the token rather than sentence level, and we hypothesize that being mistakenly overconfident on non-essential tokens has limited impact on the overall task. The same overfitting patterns on generative benchmarks is revealed in §4 on code generation benchmarks, where the AR models overfit early on validation loss while the benchmark degradation is typically delayed.

6. Diffusion Language Models also Overfit the Data

To study the full potential of tokens in DLM training, we launched an additional run in which the same 1B-token dataset was repeated for 480 epochs, yielding a total of 480B training tokens. Notably, it achieves 56% accuracy on HellaSwag and 33% on MMLU, significantly outperforming AR's 41% and 29%, respectively. Surprisingly, even under such extreme repetition, performance did not saturate,

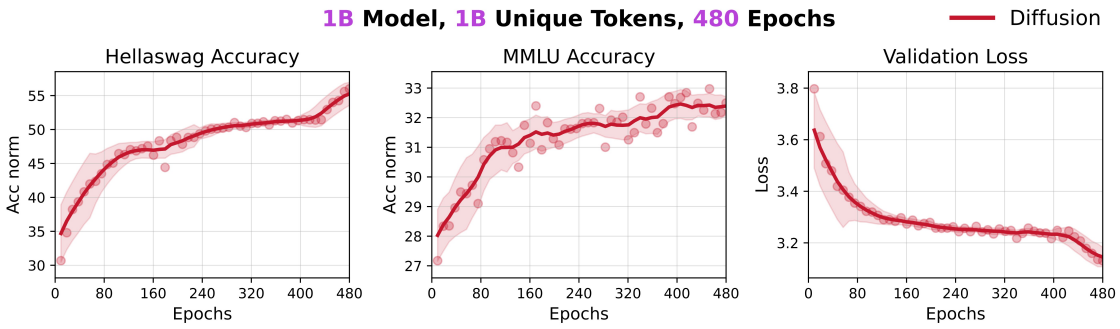

Figure 10 | The 1B-parameter DLM—trained solely on the original 1B pre-training tokens for 480 epochs—achieves 56% accuracy on HellaSwag and 33% on MMLU.

suggesting that DLMs can extract substantially more signal from a fixed 1B-token corpus. This leads us to investigate: Do DLMs eventually overfit given sufficient training?

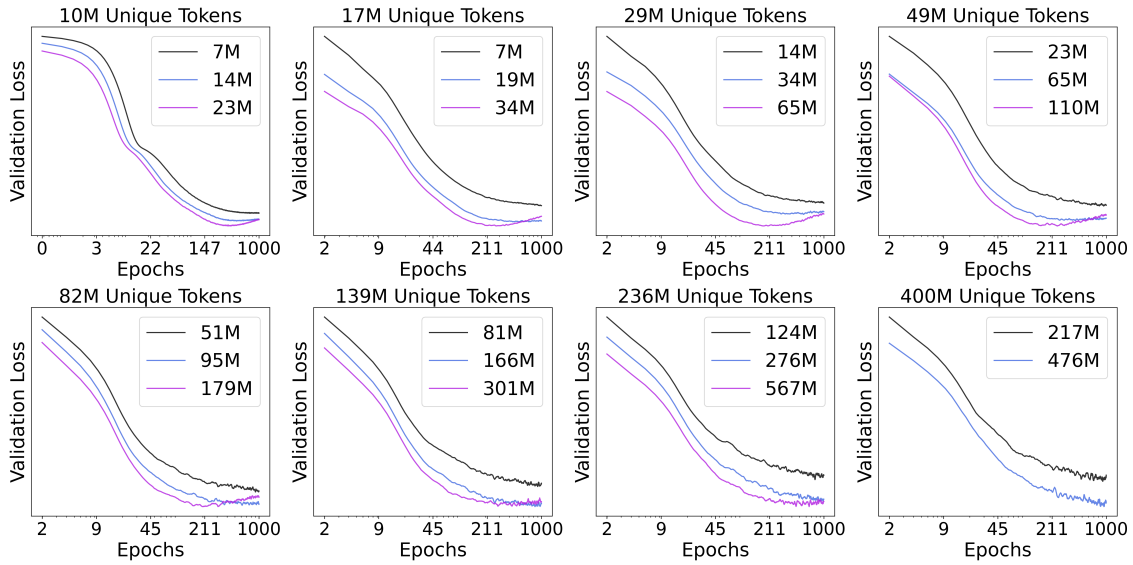

Figure 11 | **Validation loss curves for models trained with various sizes and unique data budgets, repeating up to 1000 epochs**. Diffusion language models will also overfit. The more unique data we train on, the later it overfits; the larger model we train, the earlier it overfits.

We trained models across various sizes and unique data budgets, extending training up to 1000 epochs. As illustrated in Figure 11, when the unique data is sufficiently small and the model size sufficiently large, overfitting eventually emerges after prolonged training. Specifically, we observe that the epoch at which a model begins to overfit positively correlates with the unique data size and negatively correlates with model size. In other words, larger unique data size delay overfitting, while larger models accelerate its onset. It is important to note that validation loss overfitting does not immediately imply a decline in model capability—actual performance degradation typically occurs much later (e.g., as seen in Figure 1 at 0.5B tokens and 192 epochs).

7. Discussions

7.1. What is the Real Advantage of Diffusion Language Models?

Reduced inductive bias via any-order modeling Autoregressive language models impose a strict causal inductive bias on textual data modeling, where each token prediction is conditioned solely on preceding tokens. While natural language exhibits inherent left-to-right causality from a human perspective, evidence indicates that modeling language in reverse or arbitrary order remains feasible (Xue et al., 2025). Moreover, numerous non-causal data types, such as source code, database entries, symbolic notations, and biological sequences, etc., frequently appear online. Thus, enforcing a purely causal inductive bias significantly restricts capturing the rich patterns embedded in diverse textual distributions. DLMs remove this inductive bias with the diffusion objective and the bi-directional attention, enabling such any-order modeling, fully squeezing the value of every single data point.

Super-Density: more training and test time FLOPs per task As illustrated above, diffusion models outperform AR counterparts by repeatedly processing some portion of unique data during training, effectively scaling FLOPs along the temporal dimension. The continuous-time objective utilized by masked language models is particularly advantageous, enabling high granularity in temporal FLOPs scaling. Similarly, at inference, diffusion models iteratively refine predictions, further amplifying computational density per task. Notably, bidirectional attention implies each token is computed up to N times to generate a sequence of length N, contrasting with AR models using KV cache, which compute each token only once. The effectiveness of super-density in data-constrained settings has been validated in the strictly controlled experiments in §3.5.

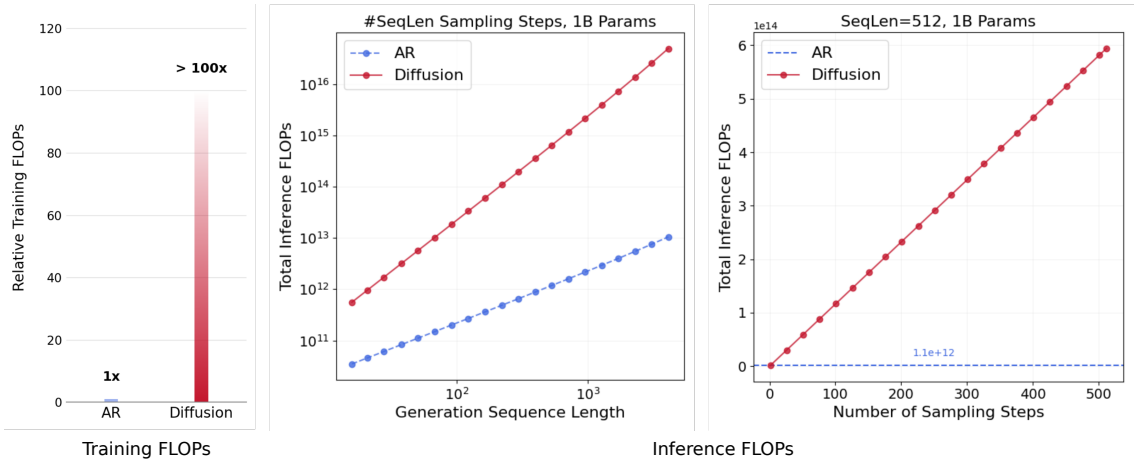

Figure 12 | (**left**) The diffusion language models are approximated to consume >100 time more FLOPs than AR counterparts to achieve their full potential in training (where the peak performance is usually much greater than AR). (**middle**) The theoretical inference FLOPs controlling the sampling steps to be equal to the sequence length. The total inference FLOPs have a power-law relationship with the generation sequence length for both. (**right**) The theoretical inference FLOPs controlling the generation sequence length, where sampling 512 steps from an AR model with KV cache ≈ sampling 1 step from the masked diffusion model.

Figure 12 compares the FLOPs of autoregressive (AR) and masked diffusion models during training and inference. At training time, our preliminary experiments indicate diffusion models require at least about two orders of magnitude (>100×) more FLOPs than AR models to reach optimal performance, with the exact figure varying depending on model size, data budget, etc. During inference, given a

fixed number of sampling steps, masked diffusion models consume between 16× and 4700× more FLOPs per task, with this gap widening as the target sequence length increases from 16 to 4096 (Figure 12 middle). Moreover, for a constant sequence length, FLOPs consumed by diffusion models scale linearly with the sampling steps. In theory, diffusion models can generate an N-token sequence within a single step, whereas AR models inherently require N sequential steps. However, due to the KV-cache mechanism, the computational cost for AR models generating N tokens is roughly equivalent to that of diffusion models performing a single sampling step. It's worth noting that a significant portion of diffusion model computations can be parallelized. Thus, in practice, before GPU compute bound is reached, the inference speed gap between diffusion and AR models at the same number of sampling steps remains acceptable. Additionally, advances in GPU architectures specifically optimized for compute-intensive workloads may further mitigate this performance gap in the near future. Reflecting on the LLM history, many recent intelligence leaps, such as T5 (Raffel et al., 2020), GPT-3 (Brown et al., 2020), and o1 (OpenAI, 2024), are the direct results of FLOPs scaling.

$$\mathcal{L} \triangleq \mathbb{E}_{t,q(\boldsymbol{x}_t|\boldsymbol{x}_0)}\left[\frac{\alpha_t'}{1-\alpha_t}\sum_{\{i|\boldsymbol{x}_t^i=m\}}\log p_\theta(\boldsymbol{x}_0^i \mid \boldsymbol{x}_t)\right] \tag{4}$$

Learning From Richer Monte Carlo Sampling When conducting multi-epoch training for masked DLMs, we effectively transform each unique data point into multiple noisy variants. Specifically, the loss function for masked diffusion models includes an expectation term $\mathbb{E}_{t,q(\boldsymbol{x}_t|\boldsymbol{x}_0)}$, placed outside the negative log-likelihood component. Here, $q(\boldsymbol{x}_t|\boldsymbol{x}_0)$ represents the distribution of masked sequences $\boldsymbol{x}_t$ conditioned on the clean input $\boldsymbol{x}_0$ at diffusion timestep t, as determined by the forward corruption process. Intuitively, this means averaging the loss across all possible masking configurations $\boldsymbol{x}_t$ at each time step $t \in [0, 1]$.

In other words, the objective function explicitly requires each data point in the pre-training dataset to be corrupted at multiple masking ratios and combinations for more effective training, by estimating a more precise expectation. Thus, data repetition emerges inherently from the diffusion model's objective rather than from an arbitrary source. Open-source models, such as LLaDA, typically corrupt each data point only once, likely due to computational limitations, approximating the expectation term using a single-sample Monte Carlo estimator.

In general, masked DLMs corrupt an input data with length L into 2^L different sequences, while AR learns causal mapping with L different sequences from short to long for a given input of the same length. Such drastic expansion of the learning space makes DLM's learning upper limit much higher than AR given a limited portion of data (though not every corruption produces a significant difference), requiring larger model size to fully fit it. There have been a lot of frontier works trying to augmenting the AR models' data with model rewriting–rewriting an input sequence into multiple variants, which also demonstrated fruitful gains for AR models (Team et al., 2025). However, we note that such data rewriting requires expert design of the whole complicated pipeline and even though it's improving the models' performance, there's not guarantee that the improvement is well-rounded. In contrast, for diffusion language models, injecting noise to the input is all you need to do the data augmentation.

7.2. Autoregressive Models are Trading Data Potential for Compute Efficiency

The autoregressive (AR) modeling methodology (decoder-only transformer architecture with teacher-forcing and causal masking) is a legendary local optimum in the AI history. Its success can be broken down into two factors:

Optimal utilization of modern GPU architectures AR achieves an exceptionally high signal-to-FLOPs ratio during training and a high Model FLOPs Utilization (MFU) during batched inference. During training, each token in a batch consistently receives gradient signals, approximately twice the expected signals compared to masked diffusion models with linear schedules. Indeed, it is challenging to identify alternative methodologies surpassing AR in terms of signal-to-FLOPs efficiency. At inference, token-by-token generation naturally facilitates throughput optimization techniques such as continuous batching, maximizing MFU and KV cache, minimizing computation. Thus, AR stands as an exceedingly robust and efficient baseline method.

Natural language can be causally modeled with low loss Empirically, pre-training on web-scale corpora demonstrates that left-to-right modeling consistently attains lower loss compared to alternative sequence orders (see Figure 2 of Xue et al. (2025)). If one must select a single sequence order for language modeling, the left-to-right order is empirically optimal (Eq. 1), as it effectively captures natural language patterns. It's also easy to interpret this: most text data are generated by humans, and humans are RNNs. However, as previously discussed, purely left-to-right modeling inherently misses certain contextual dependencies, indicating room for improvement in data potential.

Currently, there is an emerging trend in which computational resources are becoming increasingly affordable, shifting the primary constraint for scaling intelligence towards data availability. Consequently, for researchers targeting advanced intelligence, the previous emphasis on maximizing GPU utilization has diminished. The inherent modeling limitations imposed by causal masking are now becoming unacceptable. This motivates our exploration of DLMs, which intentionally sacrifice computational efficiency to achieve higher data efficiency—representing an approach diametrically opposed to autoregressive methods.

To strike a favorable balance between these two extremes, a natural strategy is interpolation, as exemplified by block diffusion methods (Arriola et al., 2025). However, achieving comparable training efficiency remains challenging: block diffusion inherently conditions each generated block on a clean context, significantly constraining training efficiency compared to the highly efficient teacher-forcing paradigm employed in autoregressive training.

7.3. Limitations

DLMs trade *compute* for *data efficiency*. Our crossovers rely on higher train- and test-time FLOPs with bidirectional attention and iterative refinement, which raises energy cost, processing time, and memory pressure compared to AR with KV cache. Whereas under the multi-token prediction scheme it can achieve much lower inference latency than AR models. Second, practical inference performance is sensitive to choices that are still under-explored at scale—masking schedules, denoising weights, step counts, and decoding policies—which complicates fair, apples-to-apples comparisons with heavily optimized AR pipelines. Third, evaluation confounds remain: diffusion objectives optimize a variational bound rather than a normalized left-to-right likelihood, so perplexity is not directly comparable, and benchmark gains may not uniformly translate to streaming, tool-use, or long-horizon generation. Fourth, heavy data reuse heightens contamination and memorization risk if deduplication and auditing are imperfect; safety and privacy audits should therefore be stricter under super-dense training. Finally, the systems stack for practical deployment (continuous batching, speculative sampling, retrieval/tool integration, RL/RLAIF for policy shaping) is less mature for DLMs than for AR, and our experiments focus on English-centric pretraining; broader multilingual, multimodal, and long-context regimes merit dedicated study.

8. Related Work

Diffusion language models Building on the theoretical foundations of DLMs (Lou et al., 2023; Ou et al., 2024; Sahoo et al., 2024; Shi et al., 2024), Nie et al. (2025) trained the first large-scale DLM from scratch, achieving performance competitive with leading open-source AR models (Dubey et al., 2024). In parallel, several commercial DLMs have emerged, demonstrating strong coding and math capabilities while offering significantly lower generation latency (Google DeepMind, 2025; Khanna et al., 2025; Song et al., 2025). Efforts have also explored hybrid approaches bridging AR and diffusion. Block diffusion (Arriola et al., 2025) performs block-wise diffusion, with block size 1 reducing to AR modeling without shift. Dream (Ye et al., 2025) initialized DLMs with AR priors and employed a "shift-by-one" diffusion strategy to better retain AR knowledge, offering another effective training paradigm. Recent work has also advanced DLM coders (Gong et al., 2025; Xie et al., 2025), DLM RL scaling (Zhu et al., 2025), accelerated inference techniques (Wu et al., 2025), pushing DLMs toward greater practicality and competitiveness.

Mitigating data constraints and the "token crisis." A complementary line of work asks how to keep scaling when *unique* tokens are scarce. Muennighoff et al. (2025) formalize data-constrained scaling laws, showing that repeating data for up to ~4 epochs yields little loss penalty at fixed compute but exhibits sharply diminishing returns thereafter; they also find that mixing in code or relaxing aggressive filters can substitute for some unique text. Xue et al. (2023) analyze multi-epoch degradation, identifying dropout as a robust regularizer; this echoes earlier evidence that small repeated fractions can disproportionately harm generalization (Hernández et al., 2022). In parallel, large open corpora aim to expand high-quality supply through better curation and deduplication: RefinedWeb (5T tokens) (Penedo et al., 2023), FineWeb (15T; with FineWeb-Edu) (Penedo et al., 2024), and Dolma (3T) (Soldaini et al., 2024).

A second strategy manufactures or selects *higher-utility* tokens. Rephrasing/rewriting pipelines improve pretraining efficiency by paraphrasing web documents (WRAP) (Maini et al., 2024), extending to multilingual rephrasing (Pieler et al., 2024) and targeted math/code rewriting (SwallowCode/SwallowMath) (Fujii et al., 2025). Large applied systems report related practices at scale; for example, Kimi K2 describes large agentic data synthesis, and independent commentary attributes part of its gains to rephrasing public corpora (Breunig, 2025; Kimi Team et al., 2025). Orthogonal levers include retrieval-time scaling with trillion-token datastores (Shao et al., 2024), perplexity-based data pruning (Ankner et al., 2024), and end-of-training domain upsampling (Blakeney et al., 2024). Our results are complementary: diffusion objectives further amplify value per *unique* token when repetition is unavoidable.

References

A. N. Amin, N. Gruver, and A. G. Wilson. Why masking diffusion works: Condition on the jump schedule for improved discrete diffusion. *arXiv preprint arXiv:2506.08316*, 2025.

Z. Ankner, C. Blakeney, K. Sreenivasan, M. Marion, M. L. Leavitt, and M. Paul. Perplexed by perplexity: Perplexity-based data pruning with small reference models. *arXiv preprint arXiv:2405.20541*, 2024. URL https://arxiv.org/abs/2405.20541.

M. Arriola, A. Gokaslan, J. T. Chiu, Z. Yang, Z. Qi, J. Han, S. S. Sahoo, and V. Kuleshov. Block diffusion: Interpolating between autoregressive and diffusion language models. *arXiv preprint arXiv:2503.09573*, 2025.

S. Black, S. Biderman, E. Hallahan, Q. Anthony, L. Gao, L. Golding, H. He, C. Leahy, K. McDonell, J. Phang, et al. Gpt-neox-20b: An open-source autoregressive language model. *arXiv preprint arXiv:2204.06745*, 2022.

C. Blakeney, M. Paul, B. W. Larsen, S. Owen, and J. Frankle. Does your data spark joy? performance gains from domain upsampling at the end of training. *arXiv preprint arXiv:2406.03476*, 2024. URL https://arxiv.org/abs/2406.03476.

D. Breunig. How rewriting training data improved kimi k2's performance. https://www.dbreunig.com/2025/07/27/kimi-applies-rephrasing-to-pre-training-data.html, 2025. Blog post.

T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Common Crawl. Statistics of common crawl monthly archives: Crawl size. 2025. URL https://commoncrawl.github.io/cc-crawl-statistics/plots/crawlsize.

A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Yang, A. Fan, et al. The llama 3 herd of models. *arXiv e-prints*, pages arXiv–2407, 2024.

K. Fujii, Y. Tajima, S. Mizuki, H. Shimada, T. Shiotani, K. Saito, M. Ohi, M. Kawamura, T. Nakamura, T. Okamoto, S. Ishida, K. Hattori, Y. Ma, H. Takamura, R. Yokota, and N. Okazaki. Rewriting pre-training data boosts llm performance in math and code. *arXiv preprint arXiv:2505.02881*, 2025. URL https://arxiv.org/abs/2505.02881.

S. Gong, R. Zhang, H. Zheng, J. Gu, N. Jaitly, L. Kong, and Y. Zhang. Diffucoder: Understanding and improving masked diffusion models for code generation. *arXiv preprint arXiv:2506.20639*, 2025.

Google DeepMind. Gemini diffusion: Our state-of-the-art, experimental text diffusion model. https://deepmind.google/models/gemini-diffusion/, 2025. Accessed: 2025-09-23.

R. Han, Y. Chen, Z. CuiZhu, L. Miculicich, G. Sun, Y. Bi, W. Wen, H. Wan, C. Wen, S. Maître, et al. Deep researcher with test-time diffusion. *arXiv preprint arXiv:2507.16075*, 2025.

D. Hernández, T. Brown, T. Conerly, N. DasSarma, D. Drain, S. El-Showk, N. Elhage, Z. Hatfield-Dodds, T. Henighan, T. Hume, S. Johnston, B. Mann, C. Olah, C. Olsson, D. Amodei, N. Joseph, J. Kaplan, and S. McCandlish. Scaling laws and interpretability of learning from repeated data. *arXiv preprint arXiv:2205.10487*, 2022. URL https://arxiv.org/abs/2205.10487.

J. Hoffmann, S. Borgeaud, A. Mensch, E. Buchatskaya, T. Cai, E. Rutherford, D. d. L. Casas, L. A. Hendricks, J. Welbl, A. Clark, et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022.

S. Huang, T. Cheng, J. K. Liu, J. Hao, L. Song, Y. Xu, J. Yang, J. Liu, C. Zhang, L. Chai, et al. Opencoder: The open cookbook for top-tier code large language models. *arXiv preprint arXiv:2411.04905*, 2024.

S. Khanna, S. Kharbanda, S. Li, H. Varma, E. Wang, S. Birnbaum, Z. Luo, Y. Miraoui, A. Palrecha, S. Ermon, et al. Mercury: Ultra-fast language models based on diffusion. *arXiv preprint arXiv:2506.17298*, 2025.

Kimi Team, Y. Bai, Y. Bao, G. Chen, et al. Kimi k2: Open agentic intelligence. *arXiv preprint arXiv:2507.20534*, 2025. URL https://arxiv.org/abs/2507.20534.

A. Lou, C. Meng, and S. Ermon. Discrete diffusion modeling by estimating the ratios of the data distribution. *arXiv preprint arXiv:2310.16834*, 2023.

I. Magnusson, N. Tai, B. Bogin, D. Heineman, J. D. Hwang, L. Soldaini, A. Bhagia, J. Liu, D. Groeneveld, O. Tafjord, et al. Datadecide: How to predict best pretraining data with small experiments. *arXiv preprint arXiv:2504.11393*, 2025.

P. Maini, S. Seto, R. Bai, D. Grangier, Y. Zhang, and N. Jaitly. Rephrasing the web: A recipe for compute and data-efficient language modeling. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14044–14072, Bangkok, Thailand, Aug. 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.757. URL https://aclanthology.org/2024.acl-long.757/.

N. Muennighoff, A. Rush, B. Barak, T. Le Scao, N. Tazi, A. Piktus, S. Pyysalo, T. Wolf, and C. A. Raffel. Scaling data-constrained language models. *Advances in Neural Information Processing Systems*, 36: 50358–50376, 2023.

N. Muennighoff, L. Soldaini, D. Groeneveld, K. Lo, J. Morrison, S. Min, W. Shi, P. Walsh, O. Tafjord, N. Lambert, et al. Olmoe: Open mixture-of-experts language models. *arXiv preprint arXiv:2409.02060*, 2024.

N. Muennighoff, A. M. Rush, B. Barak, T. Le Scao, A. Piktus, N. Tazi, S. Pyysalo, T. Wolf, and C. Raffel. Scaling data-constrained language models. *Journal of Machine Learning Research*, 26(53):1–66, 2025. URL https://jmlr.org/papers/v26/24-1000.html.

S. Nie, F. Zhu, Z. You, X. Zhang, J. Ou, J. Hu, J. Zhou, Y. Lin, J.-R. Wen, and C. Li. Large language diffusion models. *arXiv preprint arXiv:2502.09992*, 2025.

OpenAI. Learning to reason with llms. https://openai.com/index/learning-to-reason-with-llms/, 2024.

J. Ou, S. Nie, K. Xue, F. Zhu, J. Sun, Z. Li, and C. Li. Your absorbing discrete diffusion secretly models the conditional distributions of clean data. *arXiv preprint arXiv:2406.03736*, 2024.

G. Penedo, Q. Malartic, D. Hesslow, R. Cojocaru, A. Cappelli, H. Alobeidli, B. Pannier, E. Almazrouei, and J. Launay. The refinedweb dataset for falcon llm: Outperforming curated corpora with web data, and web data only. *arXiv preprint arXiv:2306.01116*, 2023. URL https://arxiv.org/abs/2306.01116.

G. Penedo, H. Kydlíček, L. Ben Allal, A. Lozhkov, M. Mitchell, C. Raffel, L. Von Werra, and T. Wolf. The fineweb datasets: Decanting the web for the finest text data at scale. In *Advances in Neural Information Processing Systems 37 (NeurIPS 2024), Datasets and Benchmarks Track*, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/370df50ccfdf8bde18f8f9c2d9151bda-Abstract-Datasets_and_Benchmarks_Track.html.

M. Pieler, M. Bellagente, H. Teufel, D. Phung, N. Cooper, J. Tow, P. Rocha, R. Adithyan, Z. Alyafeai, N. Pinnaparaju, M. Zhuravinskyi, and C. Riquelme. Rephrasing natural text data with different languages and quality levels for large language model pre-training. *arXiv preprint arXiv:2410.20796*, 2024. URL https://arxiv.org/abs/2410.20796.

A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020.

S. Sahoo, M. Arriola, Y. Schiff, A. Gokaslan, E. Marroquin, J. Chiu, A. Rush, and V. Kuleshov. Simple and effective masked diffusion language models. *Advances in Neural Information Processing Systems*, 37:130136–130184, 2024.

J. Sevilla and E. Roldán. Training compute of frontier ai models grows by 4–5× per year. May 28 2024. URL https://epoch.ai/blog/training-compute-of-frontier-ai-models-grows-by-4-5x-per-year.

R. Shao, J. He, A. Asai, W. Shi, T. Dettmers, S. Min, L. Zettlemoyer, and P. W. Koh. Scaling retrieval-based language models with a trillion-token datastore. *arXiv preprint arXiv:2407.12854*, 2024. URL https://arxiv.org/abs/2407.12854.

N. Shazeer, A. Mirhoseini, K. Maziarz, A. Davis, Q. Le, G. Hinton, and J. Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*, 2017.

J. Shi, K. Han, Z. Wang, A. Doucet, and M. Titsias. Simplified and generalized masked diffusion for discrete data. *Advances in neural information processing systems*, 37:103131–103167, 2024.

L. Soldaini, R. Kinney, A. Bhagia, D. Schwenk, D. Atkinson, R. Authur, B. Bogin, K. Chandu, J. Dumas, Y. Elazar, V. Hofmann, A. H. Jha, S. Kumar, L. Lucy, X. Lyu, N. Lambert, I. Magnusson, J. Morrison, N. Muennighoff, A. Naik, C. Nam, M. E. Peters, A. Ravichander, K. Richardson, Z. Shen, E. Strubell, N. Subramani, O. Tafjord, P. Walsh, L. Zettlemoyer, N. A. Smith, H. Hajishirzi, I. Beltagy, D. Groeneveld, J. Dodge, and K. Lo. Dolma: an open corpus of three trillion tokens for language model pretraining research. *arXiv preprint arXiv:2402.00159*, 2024. URL https://arxiv.org/abs/2402.00159.

Y. Song, Z. Zhang, C. Luo, P. Gao, F. Xia, H. Luo, Z. Li, Y. Yang, H. Yu, X. Qu, et al. Seed diffusion: A large-scale diffusion language model with high-speed inference. *arXiv preprint arXiv:2508.02193*, 2025.

D. Su, K. Kong, Y. Lin, J. Jennings, B. Norick, M. Kliegl, M. Patwary, M. Shoeybi, and B. Catanzaro. Nemotron-cc: Transforming common crawl into a refined long-horizon pretraining dataset. *arXiv preprint arXiv:2412.02595*, 2024.

K. Team, Y. Bai, Y. Bao, G. Chen, J. Chen, N. Chen, R. Chen, Y. Chen, Y. Chen, Y. Chen, et al. Kimi k2: Open agentic intelligence. *arXiv preprint arXiv:2507.20534*, 2025.

C. Wu, H. Zhang, S. Xue, Z. Liu, S. Diao, L. Zhu, P. Luo, S. Han, and E. Xie. Fast-dllm: Training-free acceleration of diffusion llm by enabling kv cache and parallel decoding. *arXiv preprint arXiv:2505.22618*, 2025.

Z. Xie, J. Ye, L. Zheng, J. Gao, J. Dong, Z. Wu, X. Zhao, S. Gong, X. Jiang, Z. Li, et al. Dream-coder 7b: An open diffusion language model for code. *arXiv preprint arXiv:2509.01142*, 2025.

F. Xue, Y. Fu, W. Zhou, Z. Zheng, and Y. You. To repeat or not to repeat: Insights from scaling llm under token-crisis. In *Advances in Neural Information Processing Systems 36 (NeurIPS 2023)*, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/b9e472cd579c83e2f6aa3459f46aac28-Paper-Conference.pdf.

S. Xue, T. Xie, T. Hu, Z. Feng, J. Sun, K. Kawaguchi, Z. Li, and Z.-M. Ma. Any-order gpt as masked diffusion model: Decoupling formulation and architecture. *arXiv preprint arXiv:2506.19935*, 2025.

J. Ye, Z. Xie, L. Zheng, J. Gao, Z. Wu, X. Jiang, Z. Li, and L. Kong. Dream 7b: Diffusion large language models. *arXiv preprint arXiv:2508.15487*, 2025.

C. Zhou, P. Liu, P. Xu, S. Iyer, J. Sun, Y. Mao, X. Ma, A. Efrat, P. Yu, L. Yu, et al. Lima: Less is more for alignment. *Advances in Neural Information Processing Systems*, 36:55006–55021, 2023.

F. Zhu, R. Wang, S. Nie, X. Zhang, C. Wu, J. Hu, J. Zhou, J. Chen, Y. Lin, J.-R. Wen, et al. Llada 1.5: Variance-reduced preference optimization for large language diffusion models. *arXiv preprint arXiv:2505.19223*, 2025.

B. Zoph, I. Bello, S. Kumar, N. Du, Y. Huang, J. Dean, N. Shazeer, and W. Fedus. St-moe: Designing stable and transferable sparse expert models. *arXiv preprint arXiv:2202.08906*, 2022.

Table 1 | Summary of downstream evaluation used in this work. CF stands for Completion/Cloze formulation, Gen stands for generative, Char stands for per-character normalization. MMLU combines 0-5 shot results for stability (Muennighoff et al., 2024).

Dataset	Format	Shot	Norm	Split	Codebase
HellaSwag	CF	0	Char	Val	Megatron LM
MMLU	CF	0-5	Char	Val	Megatron LM
HumanEval	Gen	0	-	Test	lm-evaluation-harness
MBPP	Gen	3	-	Test	lm-evaluation-harness
HumanEval+	Gen	0	-	Test	lm-evaluation-harness
MBPP+	Gen	3	-	Test	lm-evaluation-harness

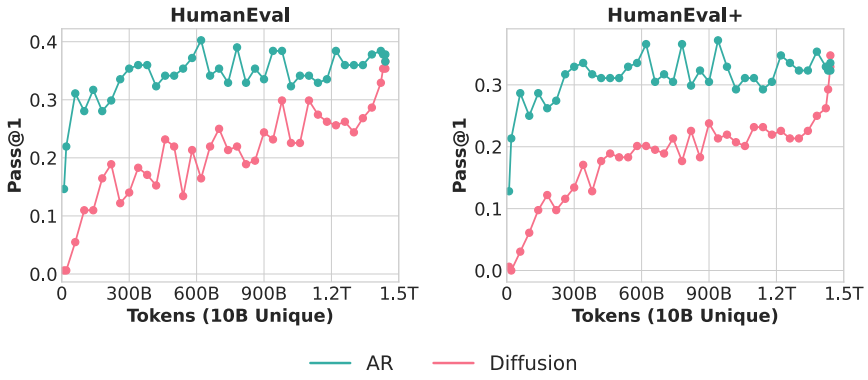

Figure 13 | **1.7B AR vs. DLM trained 10B unique code tokens for ≈ 150 epochs.** On HumanEval and HumanEval +, the crossover time is delayed and the two curves meet in the end, where the DLM didn't see diminish return. We suspect this is a direct result of the zero-shot setting.