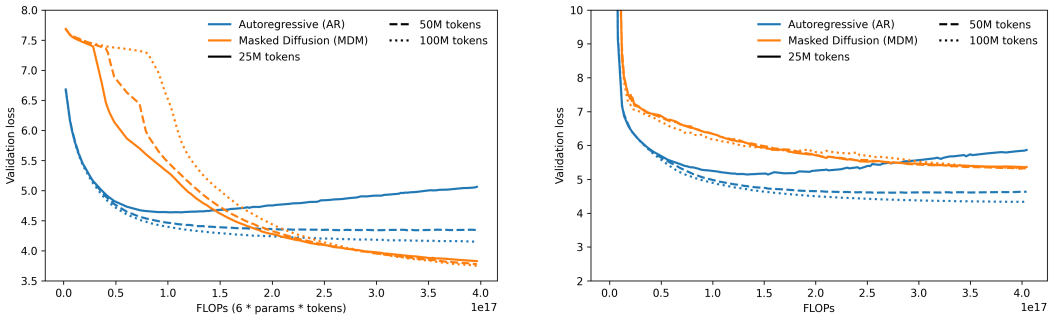
Recreating Results from Diffusion Beats Autoregressive in Data-Constrained Settings

Jared Weissberg
jared1@stanford.edu

Abstract

This is a time-constrained retest of two key questions in [Prabhudesai et al., 2025]: (i) Do Diffusion models outperform Autoregressive models in data-constrained settings, and (ii) why would Diffusion models outperform Autoregressive models in data-constrained settings? To investigate these questions, I construct a comparable architecture across both models and test these models' performance on varied token sizes and compute. I then attempt to isolate the reason for a Diffusion model's potentially superior performance in data-constrained settings by permuting the Autoregressive model. My findings are largely consistent with [Prabhudesai et al., 2025].



(a) Model A (baseline architecture)

(b) Model B (stronger architecture)

Figure 1: Validation loss versus training compute (FLOPs) for Autoregressive (AR) and masked Diffusion language models (MDM) under fixed data budgets. Each curve corresponds to training on a fixed corpus size (25M, 50M, or 100M tokens), with line style indicating dataset size and color indicating modeling paradigm. **(a)** Under the baseline Model A, MDM exhibits a clear crossover with AR across all data sizes, with intersection points shifting leftward and upward as data becomes more constrained. **(b)** Under the stronger Model B, MDM outperforms AR only in the most data-constrained regime (25M tokens), while AR continues to improve for larger datasets. The results suggest that Diffusion models gain an advantage primarily in regimes where repeated data reuse causes Autoregressive training to saturate or degrade, rather than providing uniform improvements across all data scales. These comparisons fix both architecture and data budget and do not claim Pareto optimality across all model or data scaling choices.

1 Introduction

Autoregressive (AR) models have long dominated language modeling, but Diffusion-based language models have emerged as a promising alternative. Prior work studies masked Diffusion models (MDMs) in data-constrained settings [Prabhudesai et al., 2025], where training involves repeated

Preprint. Under review.

passes over limited data, and finds that they outperform AR models when compute is abundant but data is scarce. The core observation is that AR models tend to improve quickly at low compute but saturate under repeated data, whereas masked Diffusion models continue improving beyond a critical compute threshold, consistent with new scaling laws and a closed-form crossover prediction. This advantage is attributed by [Prabhudesai et al., 2025] to the randomized masking objective, which implicitly trains over a rich distribution of token orderings and acts as an implicit data augmentation that fixed left-to-right factorization lacks.

In this report, I reproduce this comparison under a controlled setup that holds the tokenization, data pipeline, optimizer, and Transformer backbone fixed across model families. This is implemented without specialized Diffusion architectures, using a standard Transformer and differing only at the objective level. I report two experimental settings. The first uses a simplified, objective-only implementation that yields an early and clear crossover, directly illustrating the core phenomenon. The second uses a more faithful architectural recreation of prior work, providing a robustness check that more closely matches the original setup, but is limited by practical compute constraints.

I vary model size and compute under fixed unique-token budgets, and compare (i) a standard causal Transformer trained with next-token cross-entropy and (ii) a bidirectional Transformer trained with a masked-denoising objective with per-step corruption rate sampling. The goal is to isolate whether the Diffusion objective exhibits the same qualitative behavior under repeated data—early AR advantage, saturation, and a compute-dependent crossover—as model size and training compute scale.

2 Methods

I compare Autoregressive and masked Diffusion models in a strictly compute-limited setting. Because time and compute are limited, I fix the model architecture and training setup for all experiments. I do not sweep model sizes or training schedules, and I do not attempt to recover a full Pareto frontier.

Instead, I vary the amount of unique training data. This directly changes how many times the same data is reused during training. By keeping the model and optimizer fixed, I isolate the effect of data reuse without mixing it with changes in model capacity or compute allocation.

I report validation loss as a function of cumulative training FLOPs within a single training run. In this setting, the FLOPs axis represents compute spent by a fixed model over time, not the best achievable loss across different training strategies. These curves should therefore be interpreted as training dynamics rather than compute-optimal frontiers.

This setup is not meant to estimate critical compute thresholds or scaling laws. Its purpose is narrower: to test whether Diffusion models continue to improve with repeated data while Autoregressive models saturate or overfit under matched compute. Even in this constrained setting, the observed behavior matches the qualitative trends reported in prior work.

2.1 Data

I train on pre-tokenized C4 shards released by *Diffusion Beats Autoregressive in Data-Constrained Settings*. Tokens are stored as contiguous `uint16` streams without explicit document-boundary handling in my loader. Training and validation use disjoint binary files.

I define *unique data* as the number of distinct tokens available in a contiguous slice of the token stream. Data-constrained regimes are enforced by selecting a fixed slice using an offset and a maximum token count. During training, batches are formed by sampling random contiguous windows of fixed length from this slice with replacement, allowing repeated passes over a small corpus. Data reuse is quantified as the ratio of total tokens processed to unique tokens available.

2.2 Models

I use a single Transformer backbone shared across all experiments. Autoregressive and masked Diffusion models are architecturally identical within each model variant and have the same parameter count; they differ only in attention masking and training objective.

2.2.1 Model A: Baseline Transformer

The baseline model follows a standard GPT-2–style Transformer architecture. It uses absolute positional embeddings, post-activation LayerNorm, and a GELU-based feed-forward network. The model does not use rotary positional embeddings, gated MLPs, or RMSNorm. This configuration closely matches a conventional Autoregressive language model and serves as a minimal, objective-level comparison between AR and masked Diffusion training.

2.2.2 Model B: Modernized Transformer

The main model uses a modernized Transformer architecture. The model uses a vocabulary of size 50,258, with token IDs 0–50,256 corresponding to the base tokenizer vocabulary and a dedicated `[MASK]` token with ID 50,257 used for masked Diffusion training. Sequence length is fixed to 512.

The backbone consists of a stack of pre-norm Transformer blocks with RMSNorm, multi-head self-attention, and SwiGLU feed-forward layers. Rotary positional embeddings (RoPE) are applied to queries and keys in every attention layer. Token embeddings are weight-tied with the output projection. No dropout is used. In AR mode, attention is causally masked; in MDM mode, attention is fully bidirectional. All other components are shared.

2.3 Objectives

For AR training, I minimize standard next-token cross-entropy over all positions using a causal attention mask.

For MDM training, at each step I sample a corruption rate $r \sim \mathcal{U}(0, 1)$, clamped to $[0.01, 0.99]$. Each token is independently masked with probability r, with at least one masked token enforced per sequence. The model predicts the original tokens at masked positions using bidirectional attention. The training loss is the masked cross-entropy scaled by $1/r$:

$$\mathcal{L}_{MDM} = \frac{1}{r} \, \text{CrossEntropy}_{masked}.$$

For evaluation, AR models report standard validation cross-entropy. MDM models report unscaled masked cross-entropy at a fixed corruption rate $r = 0.3$. The masking process during evaluation is seeded to ensure deterministic and comparable validation metrics.

2.4 Optimization

I optimize all models using AdamW with $\beta_1 = 0.9$, $\beta_2 = 0.95$, weight decay 0.1, and mixed-precision training with bfloat16. Learning rates follow a warmup plus cosine decay schedule, with a warmup of 2% of total steps and decay to 10% of the peak learning rate. Batch size is defined per device, with global batch size scaling linearly with the number of distributed workers.

2.5 Compute Accounting

Total compute is measured in terms of tokens processed:

$$tokensprocessed = steps \times globalbatchsize \times sequencelength.$$

When needed, I estimate floating-point operations using a standard dense Transformer proxy proportional to parameter count times tokens processed. Data reuse is computed as total tokens processed divided by the number of unique tokens available.

3 Experiments

3.1 Model Variants and Data Scales

I run experiments at three data scales: 25M, 50M, and 100M unique training tokens. These scales are chosen to lie firmly in the data-constrained regime, where models make many repeated passes over the same corpus as training compute increases. This allows me to study how Autoregressive and

masked Diffusion objectives differ in their ability to reuse limited data under increasing optimization budgets.

For each data scale, I train both AR and MDM models using the same Transformer architecture and optimizer settings. The only differences between the two model classes are the attention mask (causal vs. bidirectional) and the training objective. This ensures that any observed differences arise from the learning objective rather than architectural or optimization confounds.

3.2 Autoregressive Models with Limited Ordering Augmentation on Model B

To examine whether Diffusion's advantage arises from exposure to diverse conditional prediction tasks, I augment standard Autoregressive training with a small set of fixed token orderings following the algorithm in Section 12 of [Prabhudesai et al., 2025].

For sequences of length L, I precompute a set of N orderings. The first ordering ($N = 1$) is the standard left-to-right order. Additional orderings are generated by adding Gaussian noise to the left-to-right position indices and sorting, yielding progressively more perturbed permutations. During training, each sequence independently samples one of the N predefined orderings. The first token is kept fixed, tokens in positions $[1, L]$ are permuted according to the sampled ordering, and targets are shifted accordingly to preserve the Autoregressive objective. Rotary positional embeddings (RoPE) are applied using the permuted position IDs, with position 0 anchored to the first token. Evaluation is always performed using the standard left-to-right ordering.

I evaluate $N \in \{1, 2, 4, 6\}$, where $N = 1$ corresponds to standard Autoregressive training and larger N increases the diversity of conditional prediction orders while keeping the model architecture, optimizer, and data pipeline fixed.

4 Results

4.1 AR Versus Diffusion

Figure 1 shows validation loss as a function of training compute for Autoregressive and masked Diffusion language models trained on fixed-size datasets.

Under the baseline Model A (Figure 1a), MDM eventually outperforms AR for all three dataset sizes. As the dataset size decreases, the crossover between AR and MDM occurs at lower compute and at higher loss. This coincides with AR validation loss increasing at higher compute under smaller data budgets, while MDM continues to improve.

Under the stronger Model B (Figure 1b), the same pattern appears only in the most data-constrained setting (25M tokens). In this case, AR performance degrades at higher compute, whereas MDM continues to reduce validation loss. For larger datasets (50M and 100M tokens), AR does not show signs of overfitting within the explored compute range, and no crossover is observed.

For the 100M-token setting under Model B, I also ran MDM for approximately three times the compute of the matched-compute experiments. In this extended regime, MDM continues to improve, while AR remains largely flat. Although this comparison is not compute-matched, it suggests that Diffusion models can continue to benefit from additional compute even after AR performance has saturated.

4.2 Ordering Diversity in Autoregressive Training

As demonstrated in Figure 2, a small amount of ordering diversity improves performance: $N = 2$ reduces validation loss, and $N = 4$ achieves the lowest loss overall. Increasing the number of orderings further to $N = 6$ degrades performance relative to $N = 4$, though it remains better than the baseline. This behavior is consistent with the aggressive permutation noise used in this setup, where larger N produces heavily permuted sequences that are harder to optimize.

These results suggest that part of Diffusion's advantage may stem from exposure to diverse token orderings, though this experiment isolates only ordering diversity and does not capture other perturbations applied during Autoregressive training in prior work, nor does it include a direct Diffusion baseline.

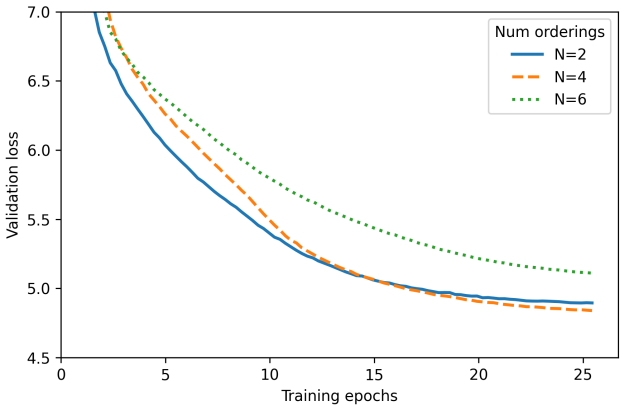
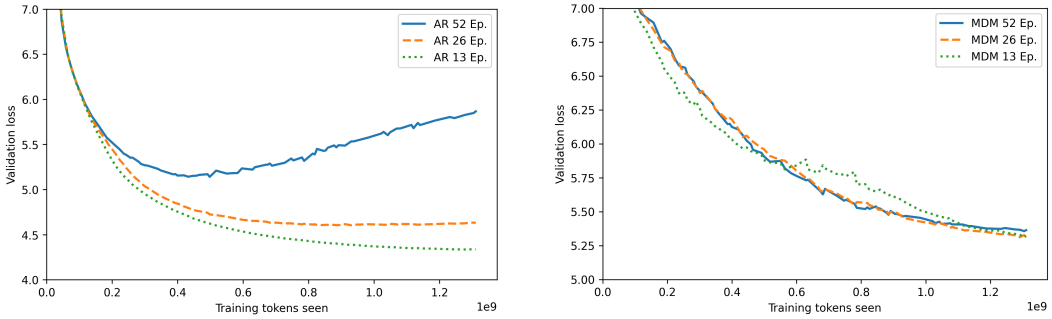


Figure 2: Validation loss versus training epochs for Autoregressive models with multiple token orderings. Moderate ordering diversity improves performance, with $N = 4$ achieving the lowest validation loss.

4.3 Robustness to Data Reuse



(a) Autoregressive (b) Masked Diffusion

Figure 3: **Validation loss versus training tokens for Model B under different levels of data reuse.** Each curve corresponds to a different effective number of epochs achieved by varying the amount of unique data while holding the architecture fixed. Autoregressive models exhibit increasing divergence and overfitting as the number of epochs increases, whereas Diffusion models produce nearly aligned training curves across epochs. This behavior is consistent with the robustness to data repetition reported in Figure 4 of [Prabhudesai et al., 2025].

I examine training dynamics under increasing data reuse by varying the effective number of epochs while holding the architecture fixed (Figure 3). For Autoregressive models, increasing the number of epochs leads to substantial divergence in validation loss, with higher data reuse resulting in worse performance. In contrast, Diffusion models exhibit nearly overlapping training curves across epoch counts, indicating little sensitivity to repeated exposure to the same data. This suggests that Diffusion models are more robust to data repetition during training. This qualitative behavior is consistent with prior findings on epoch scaling in data-constrained regimes.

5 Discussion

The goal of this paper was to recreate answers to the following questions: Do Diffusion models outperform Autoregressive models in data-constrained settings, and (ii) why would Diffusion models outperform Autoregressive models in data-constrained settings? This goal was limited by time and compute.

Results support Diffusion models outperforming AR models in data-constrained settings. The permutation experiment suggests that one way Diffusion models get this advantage is through exposure to diverse token orderings.

Thus, two other key questions in the paper went untested: When does this occur, and what is the interactive scaling law behind these constraints?

A key contribution of [Prabhudesai et al., 2025] is the determination of a critical compute curve and diminishing scaling law.

My study isolates the data-reuse regime but does not focus on what drives crossover. Two concrete areas of future research include (i) interpolating objectives that move continuously between next-token prediction and denoising while retaining efficient decoding via caching and block-parallel updates [Arriola et al., 2025], and (ii) AR baselines that randomize the factorization/order across epochs to test whether "more conditionals per sequence" explains gains without diffusion-specific training [Pannatier et al., 2024].

References

Marianne Arriola, Aaron Gokaslan, Justin T. Chiu, Zhihan Yang, Zhixuan Qi, Jiaqi Han, Subham Sekhar Sahoo, and Volodymyr Kuleshov. Block diffusion: Interpolating between autoregressive and diffusion language models, 2025. URL `https://arxiv.org/abs/2503.09573`.

Arnaud Pannatier, Evann Courdier, and François Fleuret. sigma-gpts: A new approach to autoregressive models, 2024. URL `https://arxiv.org/abs/2404.09562`.

Mihir Prabhudesai, Mengning Wu, Amir Zadeh, Katerina Fragkiadaki, and Deepak Pathak. Diffusion beats autoregressive in data-constrained settings. *arXiv preprint arXiv:2507.15857*, 2025.