RASTER OR RANDOM? LET THE DATA DECIDE

Anonymous authors
Paper under double-blind review

ABSTRACT

Autoregressive (AR) and masked diffusion models differ fundamentally in generation order, yet what determines the best order remains unclear. We show that the answer depends on the inner structure of the data. Prior comparisons contrast complete systems that differ in attention pattern, training objective, and decoding schedule at once, so the effect of order alone cannot be isolated. We introduce a unified framework that makes order the only variable: a shared causal-attention architecture takes the token order as an explicit input, holds everything else fixed, and decouples the training order from the inference order. Linear dynamical systems (LDS) and their shuffled counterpart further provide controlled proxies whose dependencies are aligned or misaligned with the token order. Across modalities, the AR order wins on language and ordinary LDS, whereas the random order wins on images and shuffled LDS. To reveal the underlying mechanisms, we decouple training and inference stages and reveal two distinct mechanisms: during training, order diversity acts as data augmentation on image-like data but not on language; during inference, scattered orders prevent the accumulation of shift along the generation. An information-theoretic analysis on LDS explains how the augmentation benefit depends on the data dependency structure. These results clarify when and why each order succeeds and guide the choice of order for a given modality.

1 INTRODUCTION

Autoregressive (AR) models and masked diffusion models are two leading paradigms for generative modeling, with strong performance in both image and language generation. However, when and why one outperforms the other remain open questions. Resolving these questions is essential for selecting models suited to different data modalities and identifying the design choices that drive their performance. Prior work has compared the two paradigms along several axes, including decoding efficiency (Feng et al., 2026; Li et al., 2024b; Zhang et al., 2026), sample efficiency, scaling (Fraij & Dauncey, 2025; Von Rütte et al., 2026), and downstream capabilities such as planning and reasoning (Ye et al.; Ni et al., 2026). We focus on a key distinction lying in generation order: conventional AR models follow a fixed left-to-right order, whereas masked diffusion models progressively reveal tokens in flexible, often random orders. Although both paradigms achieve excellent performance on images and language, some prior studies have observed that different modalities may favor different generation orders. This distinction raises a fundamental question:

> How does the order interact with the intrinsic dependency structure of data modality, and to what extent can this interaction explain the relative strengths of each paradigm?

Intuitively, the sequential structure of text may favor left-to-right generation, whereas the spatial dependencies of images, which lack a privileged one-dimensional ordering, may favor more flexible generation orders. However, addressing this question is challenging because the generation order is intertwined with many other choices in training and inference. Existing AR and masked diffusion methods differ in attention mechanisms, architectures, and training procedures, making it difficult to isolate the effect of the generation order. Besides, cross-modality comparisons introduce additional confounding factors, including data scale, dimensionality, and distribution. Therefore, empirically observed performance differences cannot be readily attributed to the alignment between the genera-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

tion order and data structure. These challenges call for a unified framework that isolates the role of orders while controlling other relevant factors.

Our contributions. To address these challenges, we introduce a unified framework for systematically studying the role of generation order in language and image generation, showing that the internal structure of the modality largely affects how the generation order behaves. First, we adopt a causal-attention network for both modalities that supports arbitrary token orderings while keeping the architecture and other modeling choices fixed. This design also decouples training order from inference order, allowing us to examine their individual contributions. Second, to stand out the internal order structure of data and control other factors, we introduce a linear dynamical system (LDS) framework with a predefined internal order, which provides a tractable while simplified setting for both empirical and theoretical analyses. Our main findings are summarized as follows:

- **The benefits of generation order critically depend on data structure.** For language modality, whose dependencies follow a natural sequential order, left-to-right autoregressive (AR) ordering beats random order. We use LDS as a synthetic data to model the language data and obtain the same trend. In contrast, for images, whose spatial dependencies lack a privileged left-to-right ordering, random ordering attains superior performance than raster order. The shuffled LDS data, which abstracts the unordered internal structure of images, validates the same phenomenon. Together, these controlled experiments imply that the generation order should align with the data structure (Section 4.1).
- **The training order and inference order play different roles.** We decouple the contributions of order during training and inference stages. We find that random order augments the training procedure, preventing overfitting and leading to lower test error, but it only works on image data. As for inference stage, random order can mitigate shift accumulation (Section 4.2).
- **Theory explains how the augmentation effect of random training depends on data structure.** We provide an information-theoretic analysis of training under the synthetic data framework. Random-order training exposes the model to more diverse token pairs. On shuffled LDS, this diversity can improve conditional estimation accuracy, whereas on ordinary LDS, it does not translate into better performance (Section 5).

2 PRELIMINARIES

Generative tasks. Given a sample $\boldsymbol{x} = (\boldsymbol{x}_0, \ldots, \boldsymbol{x}_{T-1})$ consisting of T tokens, generative models aim to approximate the joint data distribution $q(\boldsymbol{x}) = q(\boldsymbol{x}_0, \ldots, \boldsymbol{x}_{T-1})$ with a parameterized model $p_{\boldsymbol{\theta}}(\boldsymbol{x})$ trained on a finite dataset. In recent years, autoregressive and diffusion models have emerged as two leading paradigms for this task.

- **Autoregressive models.** If we let $\boldsymbol{x}_{<t} = (\boldsymbol{x}_0, \cdots, \boldsymbol{x}_{t-1})$, by factorizing the joint data distribution $q(\boldsymbol{x}) = \prod_{t=0}^{T-1} q(\boldsymbol{x}_t \mid \boldsymbol{x}_{<t})$ along the left-to-right order, an autoregressive (AR) model learns the conditional distribution $q(\boldsymbol{x}_t \mid \boldsymbol{x}_{<t})$ through next token prediction and generates tokens sequentially. Specifically, the AR model is learned by

$$\min_{\boldsymbol{\theta}} \ \mathcal{L}_{\mathrm{AR}}(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{x}} \sum_{t=0}^{T-1} -\log p_{\boldsymbol{\theta}}(\boldsymbol{x}_t \mid \boldsymbol{x}_{<t}). \tag{1}$$

  Equation (1) is the next-token prediction (NTP) loss that underlies large language models (Radford et al., 2019; Brown et al., 2020; Touvron et al., 2023). Beyond language, NTP has also become increasingly popular for image generation. By flattening the 2D grid in raster-scan order, AR models have been applied both directly to pixels (Van Den Oord et al., 2016b) and to discrete tokens produced by a learned tokenizer (Esser et al., 2021; Sun et al., 2024).
- **Masked diffusion models.** A masked diffusion model instead generates data by starting from a totally masked sequence and then gradually unmasking (predicting) a random subset of masked tokens until finally all mask tokens have been unmasked. Training corrupts a sample by masking a random subset $M \subseteq \{0, \ldots, T-1\}$ of its tokens and trains a bidirectional network to recover the masked tokens $\boldsymbol{x}_t$ with $t \in M$, from the visible ones $\boldsymbol{x}_{\bar{M}}$:

$$\mathcal{L}_{\mathrm{MD}}(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{x}} \, \mathbb{E}_M \, \frac{T}{|M|} \sum_{t \in M} -\log p_{\boldsymbol{\theta}}(\boldsymbol{x}_t \mid \boldsymbol{x}_{\bar{M}}), \qquad \bar{M} = \{0, \ldots, T-1\} \setminus M, \tag{2}$$

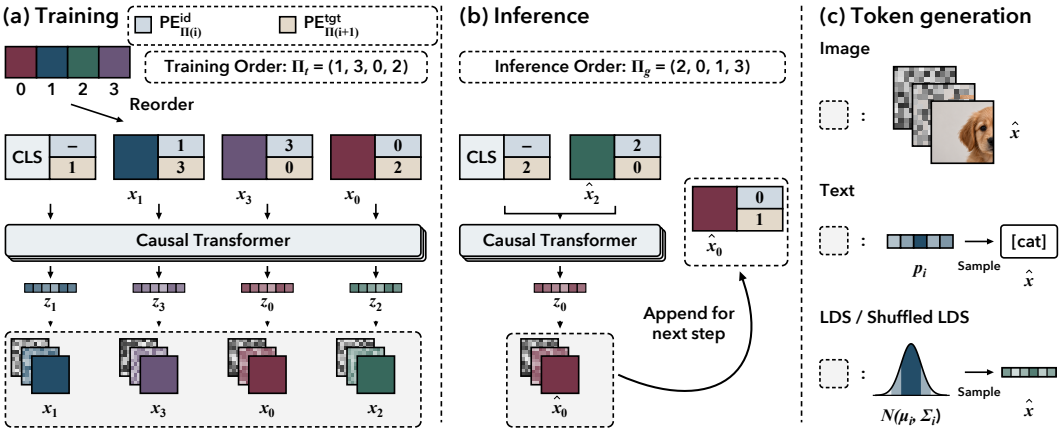

Figure 1: **An illustration of the unified architecture.** **(a)** Training: the tokens are permuted into the order Π_t, each carrying its identity and target embeddings, and the model aims to predict the next token. **(b)** Inference: the sequence grows one token per step following the inference order, and each sampled token is appended with its own two embeddings. **(c)** Token generation across modalities: image patches are generated by iterative denoising, text tokens are sampled from a categorical distribution, and LDS tokens from a Gaussian distribution.

where the masking ratio $|M|/T$ is drawn from a schedule; for a uniform schedule in Equation (2) is a variational bound on $-\log p_{\boldsymbol{\theta}}(\boldsymbol{x})$ (Austin et al., 2021; Sahoo et al., 2024). This paradigm was first established for image generation with discrete (Chang et al., 2022; 2023) and continuous tokens (Li et al., 2024a), and has been scaled to language (Lou et al., 2023; Nie et al., 2026).

Relation between AR and masked diffusion. When restricted to unmasking exactly *one position per step*, masked diffusion reduces to AR sampling along the unmasking order. To see this, let Π be a permutation of $\{0, \ldots, T-1\}$, where $\Pi(t)$ denotes the position generated at step t. Each token $\boldsymbol{x}_{\Pi(t)}$ is then generated conditioned on the previously generated tokens $\boldsymbol{x}_{\Pi(<t)}$. Under the uniform schedule, Equation (2) equals the AR objective averaged over uniformly random orders:

$$\mathcal{L}_{\mathrm{MD}}(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{x}}\, \mathbb{E}_{\Pi \sim \mathrm{Unif}(\mathcal{S}_T)} \sum_{t=0}^{T-1} -\log p_{\boldsymbol{\theta}}\big(\boldsymbol{x}_{\Pi(t)} \mid \boldsymbol{x}_{\Pi(<t)}\big), \tag{3}$$

where $\mathcal{S}_T$ is the set of all permutations of $\{0, \ldots, T-1\}$. Thus, single-token masked diffusion is equivalent to an any-order AR model (Uria et al., 2014; Yang et al., 2019), as formalized by Hoogeboom et al. (2021). Comparing Equations (1) and (3) reveals a key distinction: standard AR trains and generates in a fixed left-to-right order, whereas masked diffusion trains over all orders and, under uniform single-token unmasking, generates along a random one. More related work can be found in Section A.1.

In practice, AR and masked diffusion methods also differ in attention mechanisms (causal versus bidirectional), architectures, and training procedures. Therefore, their performance gap cannot be attributed to generation order alone. This motivates a unified framework that isolates the effect of order while holding other modeling and training choices fixed, which we introduce next.

3 A Unified Framework for Studying Generation Order

3.1 A Unified Framework

Our framework makes the order the only variable: a single causal architecture takes the order Π as an explicit input, holds everything else (attention pattern, parameter count, sequence length, and the number of generation steps) fixed, and decouples the training order from the inference order (see Figure 1; adapted from MAR (Li et al., 2024a)). Throughout, *AR* denotes the left-to-right order for text and the raster scan for images, and *random* denotes the any-order paradigm of Equation (3).

A unified training objective. For any order Π, the model $f_{\boldsymbol{\theta}}$ is trained to predict each token from the tokens that precede it in Π:

$$\mathcal{L}(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{x},\Pi} \sum_{t=0}^{T-1} \ell(\boldsymbol{z}_t, \boldsymbol{x}_{\Pi(t)}), \qquad \boldsymbol{z}_t = f_{\boldsymbol{\theta}}\big(\, \boldsymbol{x}_{\Pi(0)}, \ldots, \boldsymbol{x}_{\Pi(t-1)};\, \Pi\big), \tag{4}$$

where $\boldsymbol{z}_t$ is the prediction at step t and ℓ is a task-specific loss specified below. The two paradigms of Section 2 are two choices of the order distribution in Equation (4) and nothing else: with Π fixed to the identity (the AR order), Equation (4) is the AR objective of Equation (1); with $\Pi \sim \mathrm{Unif}(\mathcal{S}_T)$ resampled for every sample, it is the any-order objective of Equation (3) of single-token masked diffusion. The same recipe accepts any other order distribution, such as the reverse scan or a single random permutation $\bar{\Pi}$ shared by all samples, which we compare later. Throughout this paper, we refer this loss of Equation (4) as the teacher-forcing loss.

Implementation. The backbone $f_{\boldsymbol{\theta}}$ is a causal transformer that reads the tokens in the order Π, i.e. $(\boldsymbol{x}_{\Pi(0)}, \ldots, \boldsymbol{x}_{\Pi(T-1)})$, so that a single causal pass supervises all T terms of Equation (4). The order is injected through two learnable position embeddings indexed by the *original* token position: token $\boldsymbol{x}_{\Pi(t)}$ carries $\mathrm{PE}^{\mathrm{id}}_{\Pi(t)} + \mathrm{PE}^{\mathrm{tgt}}_{\Pi(t+1)}$, where the identity embedding identifies the position the token comes from and the target embedding the position the model must produce next. The loss ℓ follows the modality. For images, we follow MAR and place a per-token diffusion head over the backbone, which denoises the target token conditioned on $\boldsymbol{z}_t$: $\ell(\boldsymbol{z}_t, \boldsymbol{x}_{\Pi(t)}) = \mathbb{E}_{\tau,\boldsymbol{\epsilon}} \big\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}^{(\tau)}_{\Pi(t)}, \tau \,\big|\, \boldsymbol{z}_t) \big\|^2$, where τ is the diffusion timestep (distinct from the token index t), $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ is the injected Gaussian noise, and $\boldsymbol{x}^{(\tau)}_{\Pi(t)}$ is the target token noised to level τ. For language, $\boldsymbol{z}_t$ is a probability vector over $\mathcal{V}$ and ℓ is the cross-entropy. For (shuffled) LDS, $\boldsymbol{z}_t = (\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t)$ is a predicted mean and covariance and $\ell(\boldsymbol{z}_t, \boldsymbol{x}_{\Pi(t)}) = -\log \mathcal{N}(\boldsymbol{x}_{\Pi(t)}; \boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t)$.

Inference. Generation follows a given order Π one token per step: at step t the model reads the t tokens generated so far (plus a class token, if any), outputs $\boldsymbol{z}_t$, samples $\boldsymbol{x}_{\Pi(t)}$ from the task head, and appends the sampled token with its own two embeddings. Since the order is an explicit input, a model trained under one order can generate under any other. Training order and inference order are therefore decoupled, which allows us to report the full Training order $\times$ inference order matrix and attribute an effect to either stage.

3.2 Data Models and Their Intrinsic Orders

Using our unified framework, we investigate how generation order interacts with the data's underlying dependency structure in language, images, and a synthetic data family that isolates these structural differences.

Language and image modalities. For language, discrete tokens $\boldsymbol{x}_t \in \mathcal{V}$ are indexed in reading order, naturally aligning the index with sequential dependencies. For images, continuous patch tokens $\boldsymbol{x}_t \in \mathbb{R}^d$ are indexed in raster order over a $\sqrt{T} \times \sqrt{T}$ grid. This serialization need not align with visual dependencies arising from objects and scene geometry. For language, we train GPT-style decoders (0.68B parameters) on ClimbMix and measure end-to-end quality by MAUVE (Pillutla et al., 2021). For images, we train MAR-style causal Transformers (MAR-Base, 206M parameters) on CIFAR-10 in pixel space and use the FID score to measure image generation quality. Detailed setup is provided in Section A.2.

Synthetic data modeling real-world dependencies. To isolate the role of this alignment from differences in dimensionality and tokenization, we introduce a synthetic family that fixes the underlying dependency structure while varying its alignment with the scan order.

- **Linear dynamical system (LDS).** To mirror how the reading order aligns with sequential dependencies in language, we use an *ordinary LDS* as a tractable proxy for the alignment between token order and sequential dependencies in language:

$$\boldsymbol{x}_0 \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}_d), \qquad \boldsymbol{x}_t = \boldsymbol{F}\boldsymbol{x}_{t-1} + \boldsymbol{\epsilon}_t, \qquad \boldsymbol{\epsilon}_t \overset{\text{iid}}{\sim} \mathcal{N}(\mathbf{0}, \sigma^2 \boldsymbol{I}_d), \qquad t = 1, \ldots, T-1, \tag{5}$$

  where $\rho(\boldsymbol{F})$ denotes the spectral radius of $\boldsymbol{F}$ and $\rho(\boldsymbol{F}) < 1$ ensures stable dynamics.

Training order	Inference order				Inference order			
	rand.	AR	reverse	fixed-rand.	rand.	AR	reverse	fixed-rand.
	ClimbMix (MAUVE↑)				**LDS** ($\mathcal{K}_{\text{rev}}$ ↓)			
rand.	0.653	0.315	0.661	0.764	0.044	0.037	0.046	0.035
AR	0.011	**0.868**	0.030	0.010	746.7	**0.031**	145.7	593.3
reverse	0.009	0.013	**0.810**	0.011	341.5	277.3	0.044	274.4
fixed-rand.	0.049	0.054	0.086	0.067	55.5	16.5	24.1	0.083
	CIFAR-10 (FID↓)				**Shuffled LDS** ($\mathcal{K}_{\text{rev}}$ ↓)			
rand.	**14.26**	40.33	28.45	**14.25**	**0.038**	0.040	0.047	**0.032**
AR	225.1	23.17	161.6	241.3	109.2	0.079	33.1	190.3
reverse	227.2	166.3	20.4	242.1	193.0	179.5	0.088	114.5
fixed-rand.	185.6	172.1	160.9	23.09	124.7	202.8	112.5	0.080

Table 1: **End-to-end generation quality on language, images, and (shuffled) LDS.** Top: ClimbMix MAUVE (↑) and ordinary-LDS reverse KL $\mathcal{K}_{\text{rev}}$ (↓). Bottom: FID (↓) on 50k generated CIFAR-10 images in pixel space and shuffled-LDS reverse KL (↓). Rows indicate training order; columns indicate inference order, with shared headers across both dataset blocks. Generation perplexity and sample entropy are given in Section A.7.

- **Shuffled linear dynamical system (shuffled LDS).** To capture the misalignment between raster order and visual dependencies in images, we construct a *shuffled LDS* using a fixed permutation π of $\{0, \dots, T-1\}$ shared across all samples:

$$\boldsymbol{x}_t^{\text{shuf}} = \boldsymbol{x}_{\pi(t)}, \qquad t = 0, \dots, T-1, \tag{6}$$

  where permutation π determines the inner data structure and is distinct from the generation order Π. Shuffling preserves the underlying dependencies while disrupting their alignment with token indices: directly linked tokens can become distant in the observed sequence. This mirrors a key property of rasterized images, where shared objects and scene geometry induce dependencies between patches that are far apart in scan order. Shuffled LDS captures this misalignment without modeling the full spatial structure of images. Appendix A.9 extends this construction to a finite pool of data permutations, instead of a single permutation.

Therefore, comparing the data dependency of LDS and shuffled LDS isolates generates order alignment while holding the underlying dynamics, sequence length, and token dimensionality fixed. In our experiments, we will report results for each modality alongside its synthetic proxy: ordinary LDS for language and shuffled LDS for images. Their known ground-truth densities q enable evaluation of generation quality using the per-token reverse KL, $\mathcal{K}_{\text{rev}} = \frac{1}{T}\text{KL}(p_{\boldsymbol{\theta}} \,\|\, q)$. The detailed setup is provided in Section A.2.

4 Empirical Observation

Within our unified framework, we first evaluate end-to-end performance with matching training and sampling orders in Section 4.1. We then decouple the two in Section 4.2, to isolate their individual effects on generation quality.

4.1 End-to-End Comparison: Generation Order Depends on Data Order

We begin with an end-to-end question: when training and inference orders align, *which generation order yields the best sample quality for each modality?* Our experiments in Table 1 on language, images, and their synthetic counterparts reveal a consistent pattern:

Observation 1 (**The best generation order depends on the data's dependency structure**). AR ordering wins for language and ordinary LDS with natural sequential dependencies; random ordering wins for images and shuffled LDS without a privileged one-dimensional order.

Training order	Train loss	Test loss by inference order				Train loss	Test loss by inference order			
		rand.	AR	reverse	fixed-rand.		rand.	AR	reverse	fixed-rand.
	CIFAR-10 (MSE↓)					**ClimbMix** (bpb↓)				
rand.	0.041	**0.054**	**0.052**	**0.052**	**0.054**	0.914	**0.916**	0.907	0.939	0.916
AR	0.032	0.237	0.064	0.210	0.244	0.698	2.382	**0.706**	1.898	2.383
reverse	0.032	0.239	0.209	0.064	0.239	0.723	2.388	1.887	**0.731**	2.389
fixed-rand.	0.033	0.150	0.158	0.167	0.066	0.900	2.262	2.170	2.190	**0.906**
	Shuffled LDS ($\widehat{\mathcal{K}}_{\text{fwd}}$ ↓)					**LDS** ($\widehat{\mathcal{K}}_{\text{fwd}}$ ↓)				
rand.	0.014	**0.037**	**0.039**	**0.044**	**0.030**	0.019	**0.041**	0.034	0.043	**0.033**
AR	0.004	∞	0.078	180.0	∞	−0.009	158.7	**0.031**	68.4	146.4
reverse	0.000	∞	∞	0.083	242.1	−0.001	158.3	283.7	**0.042**	121.3
fixed-rand.	−0.019	∞	∞	102.3	0.076	0.014	∞	144.4	58.4	0.079

Table 2: **Teacher-forcing training and test losses.** Top: diffusion MSE on CIFAR-10 and cross-entropy in bpb on ClimbMix. Bottom: empirical forward KL $\widehat{\mathcal{K}}_{\text{fwd}}$ on shuffled and ordinary LDS. Rows indicate training order; column headers apply to both dataset blocks. Training losses are evaluated using the corresponding training order on training data; test losses are reported under the indicated inference orders. Values above 1000 are displayed as ∞.

We establish this pattern within our controlled framework by comparing four token orders: two sequential orders, **AR** and **reverse AR**, and two scattered orders, **random** and **fixed-random**. Random ordering draws a fresh permutation Π independently for each sample, whereas fixed-random ordering shares a fixed random permutation across all samples, separating the effects of order *diversity* from scattered *geometry*. We name a model by its training order: the *random model* is the model trained with random orders, and the *AR*, *reverse*, and *fixed-random models* are defined likewise; we call the latter three *single-order models*. All models within a modality share the same architecture and capacity, the same training data, and the same training compute; all training and inference settings are held fixed except for token order.

- **AR order wins for language modality.** Table 1 reports MAUVE for every training-order vs. inference-order pair. Among the diagonal entries, where training and inference orders match, the AR model performs best, achieving 0.868 against a ceiling of 0.89. By contrast, the random model reaches at most 0.764 across all inference orders. Generation perplexity yields the same conclusion (Section A.7). Ordinary LDS, whose sequential dependencies align with token order, corroborates this result: AR training paired with AR generation achieves the lowest reverse KL.
- **Random order wins for image modality.** The corresponding comparison on CIFAR-10 reverses the ranking (Table 1): random-order training and inference achieve the best FID (14.26), outperforming every single-order model evaluated in its training order (20.4–23.2). Shuffled LDS, which disrupts the alignment between token order and sequential dependencies, reproduces this pattern: random-order training yields the lowest reverse KL, outperforming every single-order model even in its own training order.

To explain why random order helps in images but not in language, the next subsection decouples the training and inference stages. The motivation for this decoupling comes from reading Table 1 both by row and by column, which separates two effects that the end-to-end comparison conflates.

Specifically, holding the inference order (column) fixed at fixed-random, random-order training beats fixed-random training by 8.8 FID (14.25 vs. 23.09), indicating that a diverse training distribution acts as a regularizer during training. Holding the model (row) fixed at random-order, random inference beats AR inference by 26.1 FID (14.26 vs. 40.33), while fixed-random inference matches random inference; the gain at inference therefore comes from the scattered geometry of the order, not from its diversity. Order diversity helps at training time and is irrelevant at inference time, and the end-to-end preference is the sum of these two stage-specific effects.

4.2 Decoupled Comparison: Effects of Training and Inference Order

In this section, by disentangling training and inference orders, we observe the following patterns:

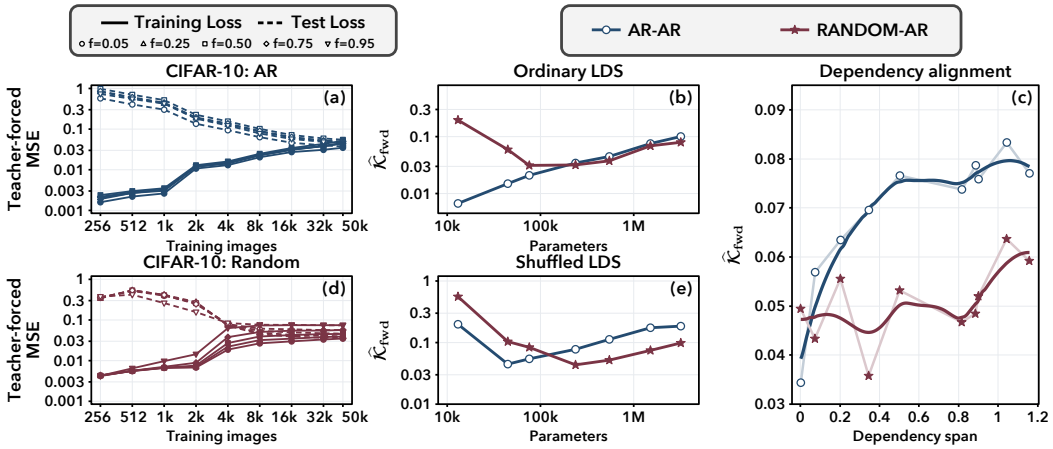

Figure 2: **(a, d) Generalization gap** on CIFAR-10, MAR-Tiny (36M parameters): teacher-forcing diffusion MSE on training (solid) and test (dashed) images versus training-set size, grouped by masked fraction f, for the AR model **(a)** and the random model **(d)**. **(b, e) Test** $\widehat{\mathcal{K}}_{\mathrm{fwd}}$ versus model capacity on ordinary LDS **(b)** and shuffled LDS **(e)**. **(c) Test** $\widehat{\mathcal{K}}_{\mathrm{fwd}}$ versus the dependency span $\nu(\pi)$ of partially shuffled LDS at fixed training-set size and model capacity. **Legend labels** in (b, c, e) read training order–inference order: Random–AR is the Random model evaluated under AR order.

Observation 2 (Effects of Training and Inference Orders)**.**

- **Training (Section 4.2.1):** On image-style data, random-order training acts as data augmentation: it diversifies the training distribution and thereby improves generalization. However, it hurts the performance on language-style data.
- **Inference (Section 4.2.2):** Random-order sampling mitigate the accumulation of token-wise shift along the generation trajectory, preventing samples from collapsing into near-constant images. This points toward new sampling strategies at inference time.

4.2.1 Training Stage: Data Augmentation

This section provides multiple perspectives to demonstrate that training with random order works as a data augmentation on image and shuffled LDS, while it hurts performance on language modality. The metrics we report in Table 2 (i.e., MSE, bpb, and $\widehat{\mathcal{K}}_{\mathrm{fwd}}$) are all teacher-forcing loss of Equation (4) in different modalities.

Image modality: random-order training acts as data augmentation. The claim is supported by the following observations.

- **Lower test error.** In Table 2, the model trained with diverse random orders achieves the lowest test MSE under every inference order (rand. row, ≈ 0.052), outperforming each single-order model even on its own training order (diagonal, ≈ 0.064). The fixed-random model, trained on one scattered permutation, behaves like the other single-order models. The gain therefore stems from the diversity of training orders, not from scattered geometry.
- **Smaller generalization gap and less memorization.** Single-order models show a large gap between training error ("Train error" columns) and test error (diagonal), a sign of severe overfitting. As Figure 2 (a,d) shows, the gap shrinks with more training images, and random-order training reaches generalization with far fewer samples than AR training (see Section A.4 for model size). A memorization score (Section A.5) confirms that, with limited data, the AR model replicates training images more often than the random-order model.

Language modality: no clear data augmentation effect. The right panel of Table 2 shows no such effect on text. The AR model attains a better test bpb than the random model (0.706 vs. 0.907). There is also no generalization gap to regularize since language models are usually trained on large scale data with few epochs and do not overfit. In this case, additional factorizations that random order

must learn only bring burden requiring larger models and more training computation. Prabhudesai et al. (2026) observes that diffusion beats AR when a large model is trained on limited data with sufficient compute. However, Figure 2 (b) shows that, in ordinary LDS data, random model never catches up the best AR model, even with enough capacity and compute.

LDS and shuffled LDS: similar data augmentation effects. Table 2 reproduces real data results on synthetic data. On shuffled LDS, random training has the same augmentation effect as on images: the random model attains the lowest test $\widehat{\mathcal{K}}_{\mathrm{fwd}}$ under every inference order, surpassing each single-order model even on its own order. On ordinary LDS, as on language, random training does not help and the AR model attains the best test KL. Figure 2(b, e) varies the model capacity. Random training, which learns every factorization, requires more capacity and compute than AR training: on shuffled LDS it overtakes AR once the capacity is sufficient, whereas on ordinary LDS it never reaches the best AR model at any tested capacity. To interpolate between the two settings, we construct partially shuffled datasets and quantify the degree of disorder: $\nu(\pi) = \frac{3}{T-2}\big[\frac{1}{T-1}\sum_{t=1}^{T-1}|\pi^{-1}(t+1) - \pi^{-1}(t)| - 1\big]$, where $\pi^{-1}(t)$ is the observed position of latent state t, so that ordinary LDS has $\nu = 0$ and a uniformly random permutation has $\mathbb{E}[\nu] = 1$. Figure 2(c) recovers the two endpoints and shows that the advantage switches in between. Section 5 provides a theoretical analysis of this pattern.

4.2.2 Inference Stage: Shift Accumulation

In this subsection, we fix the training order and investigate the effects of order during sampling. We show that, compared with sequential orders, random-order sampling mitigates shift accumulation and prevents samples from collapsing into near-constant images.

Gap between teacher-forcing loss and FID. Table 2 shows that the random model attains nearly the same teacher-forcing MSE under every inference order, yet its FID varies from 14 to 40 (first row of Table 1). The teacher-forcing loss scores each token given a ground-truth prefix, whereas at inference the model conditions on its own previous outputs, so a shift in an early token is passed on to the tokens that follow. Intuitively, under AR order every new token is an adjacent extension of the tokens just generated and therefore inherits and amplifies their pixel shift, whereas under a scattered order the tokens do not shift along one direction due the disorder. This is visible in the samples (Figure 7): images generated in random order are more detailed and diverse, while AR-generated images of the same model are more monotonous and lack variation.

Measuring shift accumulation. To quantify this effect, we measure the extent to which a shift in a token generated early affects the tokens generated later. We perturb the token sampled at one step by a small kick and track the deviation of every later token from an unperturbed rollout that shares the same order and sampling noise, so that any deviation is caused by the kick alone. $\bar{R}$ is the mean deviation per downstream step in units of the kick (definition and full statistics in Section A.6). Table 3 reports the random model under the four inference orders. AR inference attains the highest $\bar{R} = 6.5$, four times that of the scattered orders, and the ordering of $\bar{R}$ across inference orders coincides exactly with the ordering of FID. The inference-stage benefit of random order on images is thus the mitigation of shift accumulation. The same injection experiment on language, where scan orders amplify a single error as they do on images, is given in Section A.8.

Order	$\bar{R}$	FID
random	1.52	14.26
fixed-rand.	1.48	14.25
reverse	5.40	28.45
AR	6.50	40.33

Table 3: Shift accumulation $\bar{R}$ and FID of the random-order CIFAR-10 model under four inference orders.

5 Why Alignment Changes the Benefit of Random Order

Section 4.2.1 shows that the random model learns better conditional distributions than the AR model on shuffled LDS but not on ordinary LDS. This section theoretically analyzes the mechanism.

Lag-k conditionals. Under AR inference on shuffled LDS, step t predicts $\boldsymbol{x}_{\pi(t)}$ from $\boldsymbol{x}_{\pi(t-1)}$, a latent-time gap $k = \pi(t) - \pi(t-1)$ that varies with t and π; on ordinary LDS every step has $k = 1$. The elementary object is therefore the lag-k conditional distribution of $\boldsymbol{x}_{t+k}$ given $\boldsymbol{x}_t$, for $1 \le k < T$ on shuffled LDS and $k = 1$ on ordinary LDS. Since conditioning on the full prefix

yields a distribution that changes with the prefix and is intractable to analyze, we assume that the next token depends only on the immediately preceding token in the training order. Under the LDS of Section 3.2, this conditional, denoted $q_k(\cdot \mid \boldsymbol{x}_t)$, is Gaussian:

$$\boldsymbol{x}_{t+k} \mid \boldsymbol{x}_t \sim \mathcal{N}(\boldsymbol{B}_k \boldsymbol{x}_t, \boldsymbol{S}_k), \qquad \boldsymbol{B}_k = \boldsymbol{F}^k, \qquad \boldsymbol{S}_k = \sigma^2 \sum_{r=0}^{k-1} \boldsymbol{F}^r (\boldsymbol{F}^r)^\top.$$

It depends on k but not on t, so estimation decomposes into one problem per lag, with $(\boldsymbol{B}_k, \boldsymbol{S}_k)$ estimated only from lag-k pairs that appear consecutively during training (in the order $(\boldsymbol{x}_1, \boldsymbol{x}_4, \boldsymbol{x}_6)$, $(\boldsymbol{x}_1, \boldsymbol{x}_4)$ is an observation for lag 3). Random training exposes pairs at every lag, whereas AR training on shuffled LDS exposes only the pairs adjacent under π.

Error measure. We fit each lag-k conditional with a Gaussian $\widehat{p}_k(\cdot \mid \boldsymbol{x}) = \mathcal{N}(\widehat{\boldsymbol{B}}_k \boldsymbol{x}, \widehat{\boldsymbol{S}}_k)$ and measure the expected KL between the true and fitted conditionals, $\mathcal{K}^k_{\mathrm{fwd}} = \mathbb{E}_{\mathrm{train},\boldsymbol{x}} \left[\mathrm{KL}(q_k(\cdot \mid \boldsymbol{x}) \,\|\, \widehat{p}_k(\cdot \mid \boldsymbol{x})) \right]$, where the expectation covers training data, training orders, and an independent input $\boldsymbol{x}$. Since the Gaussian KL is nonlinear in the fitted parameters, we Taylor-expand it around $(\boldsymbol{B}_k, \boldsymbol{S}_k)$; the zeroth- and first-order terms vanish, leaving a quadratic term $\mathcal{Q}_k$.

Lemma 1 (Quadratic expansion of conditional KL)**.** *Let $\mathcal{Q}_k$ denote the quadratic Taylor term of $\mathbb{E}_{\boldsymbol{x}} \left[\mathrm{KL}(q_k(\cdot \mid \boldsymbol{x}) \,\|\, \widehat{p}_k(\cdot \mid \boldsymbol{x})) \right]$, if the fitted parameters lie in a fixed small neighborhood of the true parameters, then*

$$\mathcal{K}^k_{\mathrm{fwd}} = \mathbb{E}_{\mathrm{train}} \mathcal{Q}_k + O\left(\mathbb{E}_{\mathrm{train}} \left[\left(\|\widehat{\boldsymbol{B}}_k - \boldsymbol{B}_k\|_F + \|\widehat{\boldsymbol{S}}_k - \boldsymbol{S}_k\|_F \right)^3 \right] \right). \tag{7}$$

The explicit form of $\mathcal{Q}_k$ and the derivation are given in Appendix B.1.3; conditions for controlling the expected remainder are discussed in Appendix B.1.7.

We further apply the (biased) Cramér–Rao inequality to $\mathbb{E}_{\mathrm{train}} \mathcal{Q}_k$ and obtain a lower bound, which we denote by $\mathcal{B}^o_{k,s}(E)$, where $s \in \{\mathrm{ordinary}, \mathrm{shuf}\}$ denotes the dataset, $o \in \{\mathrm{AR}, \mathrm{rand}\}$ the training protocol, and E the number of training epochs (Section B.1.4). The following theorem compares the lower bounds of the two training protocols under the assumptions stated in the appendix.

Theorem 2. *Let training epoch $1 < E < \infty$. In random-order training, a uniformly random order is sampled independently for each sequence at every epoch. We have*

- ***For shuffled LDS** and any $1 \le k < T$,*

$$\mathcal{B}^{\mathrm{rand}}_{k,\mathrm{shuf}}(E) < \mathcal{B}^{\mathrm{AR}}_{k,\mathrm{shuf}}. \tag{8}$$

- ***For ordinary LDS** ($k = 1$).*

$$\mathcal{B}^{\mathrm{AR}}_{1,\mathrm{ordinary}} < \mathcal{B}^{\mathrm{rand}}_{1,\mathrm{ordinary}}(E). \tag{9}$$

Remarks. To validate the theory in the single-token conditioning setting, Section B.2 trains models that only take one token as input and reproduces its two cases: random training lowers the error on shuffled LDS but not on ordinary LDS. Overall, the theory provides a statistical insight of the augmentation effect: random orders expose token pairs at every lag, which raises the Fisher information available for estimating each conditional and tightens the error bound when the data's dependencies are not aligned with the token order. Proofs are given in Section B.1.

6 Practical Implications

We build a unified and fair framework for analyzing how the generation order should depend on the data structure. For other modalities such as video, audio, weather dynamics, and proteins, the same framework can be used to propose improved training strategies. We also quantify how different inference orders affect generation through the closed-loop response $\bar{R}$, which motivates future work on sampling strategies that improve this metric.

AI USE STATEMENT

In this work, we used generative AI tools to implement and debug parts of the experimental, evaluation, and plotting code, to reformat experimental results into tables, and to assist with translation during writing. We have not used generative AI tools to develop the theoretical model or conceptual framework, formulate mathematical claims, provide ingredients for or assist in writing proofs, propose or refine hypotheses, design or provide feedback on the research methodology or experiments, or interpret results; the synthetic datasets follow definitions specified by the authors. The remaining required-disclosure tasks (cleaning and reformatting datasets and supporting qualitative and thematic data analysis) are not applicable to this work. Additionally, we used generative AI tools to edit the manuscript for readability, draft and revise figure and table captions, and search for relevant literature. We have reviewed all AI-assisted work: AI-generated code was checked and tested by the authors, every figure and table was verified against the underlying data, and all AI-edited text was read and corrected by the authors. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work is a methodological study of generation order in autoregressive and masked diffusion models. It involves no human subjects and collects no personal data. All experiments use publicly available datasets under their respective licenses, namely CIFAR-10, the official 32×32 FFHQ thumbnails (used only for aggregate quality and error-propagation measurements), and the ClimbMix text corpus, together with synthetic data generated from linear dynamical systems. Web-scale text corpora such as ClimbMix may contain biased or offensive content, and language models trained on them can reproduce it; our models are small research models trained for controlled comparisons, and we do not release them as general-purpose generators. We do not foresee negative societal impacts beyond those common to generative modeling research, and we have read and adhere to the ICLR Code of Ethics.

REPRODUCIBILITY STATEMENT

The unified framework, including the training objective, the order-injection mechanism, and the inference procedure, is specified in Section 3, and the synthetic LDS and shuffled-LDS data are defined in Section 3.2. Section A.2 lists the model architectures, parameter counts, and training recipes for the text, image, and synthetic experiments, together with the datasets and their preprocessing; all real datasets (CIFAR-10, FFHQ, ClimbMix) are publicly available. The definitions and measurement protocols of every reported metric are given in Section A.7 (MAUVE, generation perplexity, sample entropy), Section A.6 (closed-loop injection and $\bar{R}$), and Section A.5 (memorization score), and Section A.3 reports a full replication of the image experiments on a second dataset. For the theoretical results, Section 5 states the assumptions and Section B contains the complete proofs. Code, training configurations, and evaluation scripts will be released upon publication.

REFERENCES

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.

Zalán Borsos, Matt Sharifi, Damien Vincent, Eugene Kharitonov, Neil Zeghidour, and Marco Tagliasacchi. Soundstorm: Efficient parallel audio generation. *arXiv preprint arXiv:2305.09636*, 2023.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T Freeman. Maskgit: Masked generative image transformer. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11305–11315. IEEE, 2022.

Huiwen Chang, Han Zhang, Jarred Barber, AJ Maschinot, Jose Lezama, Lu Jiang, Ming-Hsuan Yang, Kevin Murphy, William T Freeman, Michael Rubinstein, et al. Muse: Text-to-image generation via masked generative transformers. *arXiv preprint arXiv:2301.00704*, 2023.

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.

Jade Copet, Felix Kreuk, Itai Gat, Tal Remez, David Kant, Gabriel Synnaeve, Yossi Adi, and Alexandre Défossez. Simple and controllable music generation. *Advances in neural information processing systems*, 36:47704–47720, 2023.

Justas Dauparas, Ivan Anishchenko, Nathaniel Bennett, Hua Bai, Robert J Ragotte, Lukas F Milles, Basile IM Wicky, Alexis Courbet, Rob J de Haas, Neville Bethel, et al. Robust deep learning–based protein sequence design using proteinmpnn. *Science*, 378(6615):49–56, 2022.

Shizhe Diao, Yu Yang, Yonggan Fu, Xin Dong, Dan Su, Markus Kliegl, Zijia Chen, Peter Belcak, Yoshi Suhara, Hongxu Yin, et al. Climb: Clustering-based iterative data mixture bootstrapping for language model pre-training. arxiv preprint, 2025. *URL https://arxiv. org/abs/2504.13161*.

Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12873–12883, 2021.

Guhao Feng, Yihan Geng, Jian Guan, Wei Wu, Liwei Wang, and Di He. Theoretical benefit and limitation of diffusion language model. *Advances in Neural Information Processing Systems*, 38: 24415–24459, 2026.

Ahmad Fraij and Sam Dauncey. Double descent as a lens for sample efficiency in autoregressive vs. discrete diffusion models. *arXiv preprint arXiv:2509.24974*, 2025.

Yonggan Fu, Lexington Whalen, Abhinav Garg, Chengyue Wu, Maksim Khadkevich, Nicolai Oswald, Enze Xie, Daniel Egert, Sharath Turuvekere Sreenivas, Shizhe Diao, et al. Nemotron-labs-diffusion: A tri-mode language model unifying autoregressive, diffusion, and self-speculation decoding. *arXiv preprint arXiv:2607.05722*, 2026.

Shansan Gong, Shivam Agarwal, Yizhe Zhang, Jiacheng Ye, Lin Zheng, Mukai Li, Chenxin An, Peilin Zhao, Wei Bi, Jiawei Han, et al. Scaling diffusion language models via adaptation from autoregressive models. In *International Conference on Learning Representations*, volume 2025, pp. 5046–5073, 2025.

Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *The journal of machine learning research*, 13:723–773, 2012.

Chunsan Hong, Sanghyun Lee, and Jong Chul Ye. Unifying masked diffusion models with various generation orders and beyond. *arXiv preprint arXiv:2602.02112*, 2026.

Emiel Hoogeboom, Alexey A Gritsenko, Jasmijn Bastings, Ben Poole, Rianne van den Berg, and Tim Salimans. Autoregressive diffusion models. *arXiv preprint arXiv:2110.02037*, 2021.

Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *2019 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 4396–4405. IEEE, 2019.

Tianhong Li, Yonglong Tian, He Li, Mingyang Deng, and Kaiming He. Autoregressive image generation without vector quantization. *Advances in Neural Information Processing Systems*, 37: 56424–56445, 2024a.

Yuchen Li, Alexandre Kirchmeyer, Aashay Mehta, Yilong Qin, Boris Dadachev, Kishore Papineni, Sanjiv Kumar, and Andrej Risteski. Promises and pitfalls of generative masked language modeling: Theoretical framework and practical guidelines. *arXiv preprint arXiv:2407.21046*, 2024b.

Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion modeling by estimating the ratios of the data distribution. *arXiv preprint arXiv:2310.16834*, 2023.

Zanlin Ni, Shenzhi Wang, Yang Yue, Tianyu Yu, Weilin Zhao, Yeguo Hua, Tianyi Chen, Jun Song, Cheng Yu, Bo Zheng, et al. The flexibility trap: Rethinking the value of arbitrary order in diffusion language models. *arXiv preprint arXiv:2601.15165*, 2026.

Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, Jun Zhou, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. Large language diffusion models. *Advances in Neural Information Processing Systems*, 38:50608–50646, 2026.

Krishna Pillutla, Swabha Swayamdipta, Rowan Zellers, John Thickstun, Sean Welleck, Yejin Choi, and Zaid Harchaoui. Mauve: Measuring the gap between neural text and human text using divergence frontiers. *Advances in Neural Information Processing Systems*, 34:4816–4828, 2021.

Mihir Prabhudesai, Mengning Wu, Amir Zadeh, Katerina Fragkiadaki, and Deepak Pathak. Diffusion beats autoregressive in data-constrained settings. *Advances in Neural Information Processing Systems*, 38:10581–10606, 2026.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Sucheng Ren, Qihang Yu, Ju He, Xiaohui Shen, Alan Yuille, and Liang-Chieh Chen. Beyond next-token: Next-x prediction for autoregressive visual generation. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 15781–15791. IEEE, 2025.

Subham S Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin T Chiu, Alexander Rush, and Volodymyr Kuleshov. Simple and effective masked diffusion language models. *Advances in Neural Information Processing Systems*, 37:130136–130184, 2024.

Yawar Siddiqui, Antonio Alliegro, Alexey Artemov, Tatiana Tommasi, Daniele Sirigatti, Vladislav Rosov, Angela Dai, and Matthias Nießner. Meshgpt: Generating triangle meshes with decoder-only transformers. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19615–19625. IEEE, 2024.

Peize Sun, Yi Jiang, Shoufa Chen, Shilong Zhang, Bingyue Peng, Ping Luo, and Zehuan Yuan. Autoregressive model beats diffusion: Llama for scalable image generation. *arXiv preprint arXiv:2406.06525*, 2024.

Keyu Tian, Yi Jiang, Zehuan Yuan, Bingyue Peng, and Liwei Wang. Visual autoregressive modeling: Scalable image generation via next-scale prediction. *Advances in neural information processing systems*, 37:84839–84865, 2024.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models, 2023. URL https://arxiv.org/abs/2302.13971.

Benigno Uria, Iain Murray, and Hugo Larochelle. A deep and tractable density estimator. In *International Conference on Machine Learning*, pp. 467–475. PMLR, 2014.

Aaron Van Den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, Koray Kavukcuoglu, et al. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 12(1), 2016a.

Aäron Van Den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. In *International conference on machine learning*, pp. 1747–1756. PMLR, 2016b.

Dimitri Von Rütte, Janis Fluri, Omead Pooladzandi, Bernhard Schölkopf, Thomas Hofmann, and Antonio Orvieto. Scaling behavior of discrete diffusion language models. In *International Conference on Learning Representations*, volume 2026, pp. 36638–36663, 2026.

Tong Wu, Zhihao Fan, Xiao Liu, Hai-Tao Zheng, Yeyun Gong, Jian Jiao, Juntao Li, Jian Guo, Nan Duan, Weizhu Chen, et al. Ar-diffusion: Auto-regressive diffusion model for text generation. *Advances in Neural Information Processing Systems*, 36:39957–39974, 2023.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32, 2019.

Jiacheng Ye, Jiahui Gao, Shansan Gong, Lin Zheng, Xin Jiang, Zhenguo Li, and Lingpeng Kong. Beyond autoregression: Discrete diffusion for complex reasoning and planning, 2025. *URL https://arxiv. org/abs/2410.14157*.

Shaorong Zhang, Longxuan Yu, Rob Brekelmans, Luhan Tang, Salman Asif, and Greg Ver Steeg. Generation order and parallel decoding in masked diffusion models: An information-theoretic perspective. *arXiv preprint arXiv:2602.00286*, 2026.

A Appendix

A.1 Related Work

Masked diffusion and autoregressive generative models. *Masked diffusion for images.* MaskGIT generates from a fully masked canvas by iterative parallel decoding (Chang et al., 2022), Muse scales the same recipe to text-to-image (Chang et al., 2023), and MAR replaces the discrete tokenizer with a per-token diffusion head so that the same masked prediction works on continuous tokens (Li et al., 2024a); MAR is the architecture our framework builds on. *Masked diffusion for language.* D3PM defines diffusion in discrete state spaces (Austin et al., 2021), SEDD learns the ratios of the data distribution instead of the transitions (Lou et al., 2023), MDLM reduces the masked case to a simple likelihood bound (Sahoo et al., 2024), and LLaDA carries it to the scale of a modern language model (Nie et al., 2026). *Autoregressive image generation.* PixelRNN generates pixels in raster order (Van Den Oord et al., 2016b), VQGAN puts a transformer over discrete image tokens (Esser et al., 2021), LlamaGen shows that a plain language-model architecture over such tokens rivals diffusion (Sun et al., 2024), and VAR keeps the autoregressive factorization but predicts the next scale rather than the next token (Tian et al., 2024).

Comparing autoregressive and diffusion generation. Which paradigm is better turns out to depend on the axis of comparison. On efficiency, masked diffusion reaches near-optimal perplexity in a constant number of steps but needs a step count linear in sequence length to get sequences right, which erases its advantage over autoregressive decoding (Feng et al., 2026); the culprit is parallel decoding, whose tokens are drawn from their marginals and ignore one another (Li et al., 2024b; Zhang et al., 2026). On sample efficiency and scaling the evidence is mixed, with discrete diffusion reported as both less efficient (Fraij & Dauncey, 2025) and competitive (Von Rütte et al., 2026). On capability the sign flips with the task: diffusion wins on problems demanding global consistency and long-horizon planning (Ye et al.), yet its arbitrary-order freedom can weaken reasoning (Ni et al., 2026).

All of these compare *systems*, which differ at once in attention pattern, training objective, supervised positions per step and number of decoding steps, so an end-to-end difference cannot be attributed to any one of them. We hold all of them fixed and vary only the token order (Section 3), and we separate the training stage from the inference stage. That is what lets us show that the same order choice is right on images and wrong on text for *different* reasons on the two stages, and it recasts the parallel-decoding inconsistency above as a question about the reveal schedule rather than about diffusion versus autoregression.

Interpolating between autoregression and diffusion. A separate line combines the two rather than choosing, mostly by controlling when information becomes available: adapting an autoregressive checkpoint into a diffusion language model (Gong et al., 2025), making the noise level position-dependent so earlier tokens denoise first (Wu et al., 2023; Chen et al., 2024), generalizing what is predicted per step from one token to an arbitrary subset or a learned mask schedule (Ren et al., 2025; Hong et al., 2026), and training both objectives jointly to switch modes at inference (Fu et al., 2026). These occupy the interior of the axis we study; we measure its two endpoints under one architecture, separately on each stage.

Generation order in other modalities. What counts as the natural order varies with the modality. It is physically given in raw audio, where WaveNet generates samples forward in time (Van Den Oord et al., 2016a); it is a design variable over neural audio codec tokens, where MusicGen offsets successive codebooks with a delay pattern (Copet et al., 2023) and SoundStorm fills them by confidence-based parallel decoding (Borsos et al., 2023); it is imposed by convention in 3D geometry, where MeshGPT sorts vertices and faces canonically before tokenizing (Siddiqui et al., 2024); and it is deliberately randomized in protein design, where ProteinMPNN samples a uniform decoding permutation (Dauparas et al., 2022).

A.2 Experimental Setup

Language. The text models are GPT-style decoders with 24 layers, width 1536, 12 attention heads of dimension 128, MLP ratio 4, rotary position embeddings, a $32{,}768$-token vocabulary, and a 2048-token context (0.68B non-embedding parameters; 1.39B in total including the token, output, and per-layer value embeddings). They are trained on 8.8B tokens of ClimbMix (Diao et al.) with a

batch of 2^{20} tokens, one model per order with an identical recipe; one pass over our local shards yields 7.75B tokens, so the last 13% of training revisits data, identically for all orders. The order enters the model through two learned position embeddings as in Section 3, so every model can be evaluated and can generate under every order. End-to-end quality is measured by *MAUVE* (Pillutla et al., 2021), a two-sample distance between generated and real text in GPT-2-large feature space (higher is better); two disjoint real sets score 0.89 against each other, which is the ceiling at our sample size. We also report the *generation perplexity* of the samples under an independently trained oracle model (lower is better). Sampling protocol and the definitions of both metrics are given in Section A.7, which also reports a replication with 12-layer, 768-wide models (6 heads; 85M non-embedding, 0.29B total parameters).

Images. The image model is a causal MAR: a 24-layer, 768-wide Transformer with 12 heads (MAR-Base, the encoder and decoder of MAR-B merged into a single causal stack) followed by a per-token diffusion head, an MLP of 6 blocks and width 1024 that uses 100 denoising steps at inference; 206M parameters in total. A buffer of 16 class tokens precedes the image tokens (class-conditional, label dropout 0.1 during training, no classifier-free guidance at inference). Images are modeled directly in pixel space at 32×32 resolution and partitioned into 4×4 patches, yielding a sequence of $8\times8=64$ tokens per image. Every model is trained for 800 epochs on CIFAR-10 with batch size 256, base learning rate 2×10^{-4}, and EMA decay 0.9999; evaluation uses the EMA weights. MAR-Tiny, used for the training-set-size sweep of Figure 2, has 12 layers, width 384, 6 heads, and a 4-block, 768-wide diffusion head (36M parameters). End-to-end quality is measured by FID on 50k samples against the training set. Section A.3 replicates the image experiments on FFHQ and reaches the same conclusion.

Synthetic data. Each LDS sequence has $T=16$ tokens of dimension $d=12$; $\boldsymbol{F}$ is a random matrix rescaled to spectral radius 0.9, $\boldsymbol{x}_0\sim\mathcal{N}(\mathbf{0},\boldsymbol{I}_d)$, and the noise standard deviation is $\sigma=0.3$. Shuffled LDS applies one fixed permutation π to every sequence. The models are causal Transformers with the order-controlled design of Section 3, whose head outputs the mean and full covariance of a Gaussian and which are trained by the Gaussian negative log-likelihood of Equation (4); the capacity sweep of Figure 2(b, e) varies depth and width from 10^4 to 2×10^7 parameters. For the tabulated results on both ordinary and shuffled LDS, we use 3-layer Transformers with hidden dimension 96. Both AR and random-order models are trained on 50,000 sequences for 2,000 epochs. Their known ground-truth densities allow us to evaluate free generation using the per-token reverse KL:

$$\mathcal{K}_{\mathrm{rev}}=\frac{1}{T}\mathrm{KL}(p_{\boldsymbol{\theta}}\|q)=\frac{1}{T}\mathbb{E}_{\boldsymbol{y}\sim p_{\boldsymbol{\theta}}}\left[\log\frac{p_{\boldsymbol{\theta}}(\boldsymbol{y})}{q(\boldsymbol{y})}\right]. \tag{10}$$

Here, $\boldsymbol{y}=(\boldsymbol{y}_0,\ldots,\boldsymbol{y}_{T-1})$ is a generated sequence of T tokens, $p_{\boldsymbol{\theta}}$ is the model distribution, and q is the true data distribution. This metric measures the discrepancy between the generated and true distributions.

A.3 FFHQ-32: Replication of Sections 4.1 and 4.2

Setup. We repeat the whole of Sections 4.1 and 4.2 on FFHQ (Karras et al., 2019) at 32×32, using the identical recipe: causal MAR-Base with a per-token diffusion head, pixel space, 4×4 patches ($8\times8=64$ tokens), 800 epochs, one model per token order, and the same four orders. Images are the official 128×128 thumbnails box-downsampled to 32×32; we keep the official split, training on images 00000–59999 (60k) and holding out 60000–69999 (10k) for the test MSE. FFHQ carries no labels, so the models are unconditional (one class token) — the only deviation from the CIFAR-10 setup. FID uses 50k samples against the 60k training images; the reference distribution differs from CIFAR-10, so FID values are comparable only within this section. The closed-loop measurement uses the same $\delta=0.05$ kick, the same 8 injection steps and 256 trajectory pairs, and the $\delta=0$ control is again bit-identical.

Results. All three conclusions of Sections 4.1 and 4.2 carry over. In Section A.3 the random model attains the lowest test MSE under every order, the single-order models' train-set advantage inverts at test time (AR $+52\%$, reverse $+51\%$, fixed-random $+89\%$, versus $+23\%$ for random order), and fixed-random behaves like the other single-order models. In Section A.3 scan orders amplify the injected error along the generation while scattered orders do not, and the ordering of the propagation totals A again matches the ordering of FID. In Section A.3 order diversity alone is worth 7.9 FID at a

fixed fixed-random generation geometry (5.62 vs. 13.55), scattered generation beats scan generation for the random model, and every single-order model collapses off its training order.

The one quantitative difference is the strength of the propagation: the scattered rows of Section A.3 match CIFAR-10 almost exactly ($A = 80.4$ vs. 84.8), but the scan rows are roughly halved (148.7 vs. 281 for AR). Amplification relative to scattered generation drops from $3.3\times$ on CIFAR-10 to $1.8\times$ here, and the FID ratio moves with it ($2.83\times \to 2.30\times$): on aligned faces the random model is correspondingly more robust to the inference order (worst-order FID 12.74 versus 40.33 on CIFAR-10). The excess inter-token correlation is weaker as well: AR generation raises the correlation between tokens by $+0.07$ over real FFHQ-32 images, about twice the value of the scattered orders ($+0.04$), compared with $+0.29$ versus $+0.07$–$+0.08$ on CIFAR-10; reverse generation shows no excess beyond the scattered orders ($+0.03$) even though it still amplifies the injected error in R and A.

	random	AR	reverse	fixed-random
random	**0.056** (0.045)	**0.055**	**0.055**	**0.056**
AR	0.213	0.062 (0.041)	0.226	0.217
reverse	0.212	0.228	0.062 (0.041)	0.208
fixed-random	0.158	0.161	0.164	0.068 (0.036)

Table 4: Teacher-forcing diffusion MSE on the FFHQ-32 *test* split (rows: training order; columns: inference order). Diagonal entries additionally show the *training-set* MSE in parentheses. Compare Table 2.

			FFHQ-32 (CIFAR-10)			
inference order	R (first 8)	R (final 8)	A	$\bar{R}$	excess corr.	FID
fixed-random	1.32	1.62	78.2 (82.9)	1.36	$+0.04$ ($+0.07$)	5.62 (14.25)
random	1.25	1.68	80.4 (84.8)	1.40	$+0.04$ ($+0.08$)	5.55 (14.26)
reverse	1.59	4.82	131.8 (236)	2.80	$+0.03$ ($+0.19$)	10.33 (28.45)
AR	1.66	**5.66**	**148.7 (281)**	**3.19**	$\mathbf{+0.07}$ ($\mathbf{+0.29}$)	**12.74 (40.33)**

Table 5: Closed-loop injection on the FFHQ-32 random model under four inference orders ($\delta = 0.05$), with the CIFAR-10 values of Table 3 repeated for reference. R: response in units of the kick, averaged over $\Delta = 1..8$ and over the final 8 generation steps; A: total response; $\bar{R}$: mean response per downstream step; excess corr.: mean off-diagonal inter-token correlation of the generated samples minus that of real test images (FFHQ-32: 0.21; CIFAR-10: 0.29), computed as in Section 4.2.2.

	random	AR	reverse	fixed-random
random	**5.55**	12.74	10.33	**5.62**
AR	282.0	11.08	242.6	298.3
reverse	270.6	247.0	11.90	283.0
fixed-random	251.3	230.2	223.3	13.55

Table 6: FID (50k samples) on FFHQ-32, pixel space (rows: training order; columns: inference order). Compare Table 1.

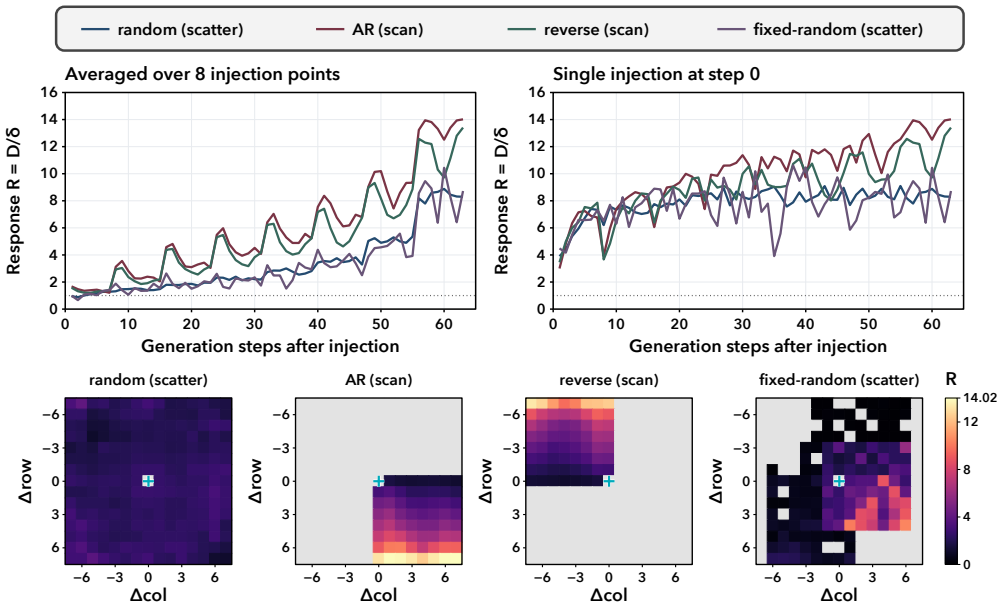

Figure 3: Closed-loop error injection on the FFHQ-32 random model, the counterpart of Figure 8. **Top:** response $R(\Delta)$ versus steps after injection. **Bottom:** response as a function of the grid offset from the injected cell (grey cells are unreachable under the given order).

A.4 Additional Training-Stage Results on Images

Figure 4 illustrates the two stage-specific measurements used in Sections 4.2.1 and 4.2.2: the teacher-forcing MSE, which scores every token from its ground-truth prefix without any rollout, and closed-loop error injection, which perturbs a single sampled token and tracks its effect on the rest of the generation.

Effect of model size. Figure 5 repeats Figure 2 at MAR-Base (206M), and the same ordering holds one octave to the right. At MAR-Tiny the random model's train and test curves have merged by 16k images (test/train ratio 1.0–1.3 at every f), whereas the AR model needs the full 50k to reach the same state and is still a factor 1.7–2.1 apart at 16k; at MAR-Base the random model has merged at $f = 0.95$ by 32k and is within 1.4–1.9 at 50k, while the AR model is still a factor 1.8–2.2 apart at 50k. Dividing the parameter count by six thus shifts the transition of both models to the left by roughly a factor of four in data, whereas at either capacity order diversity is worth a factor of about three to four in training images: capacity sets where the transition happens, and order sets how much data it takes to cross it.

Two details of Figures 2 and 5 are specific to order. The random model's residual train–test gap grows as the masked fraction f decreases (at MAR-Base and 50k images it is 1.0 at $f = 0.95$ and 1.9 at $f = 0.05$), so its memorization is conditional on having context, as one expects when the conditioning pattern itself is randomized during training; the AR model shows no such dependence because its context is always the same AR prefix. In the small-data regime the AR model's test MSE is *higher* with more context ($f = 0.5$) than with less ($f = 0.95$): its conditionals are valid only on training prefixes, and an unseen prefix takes it off the learned manifold.

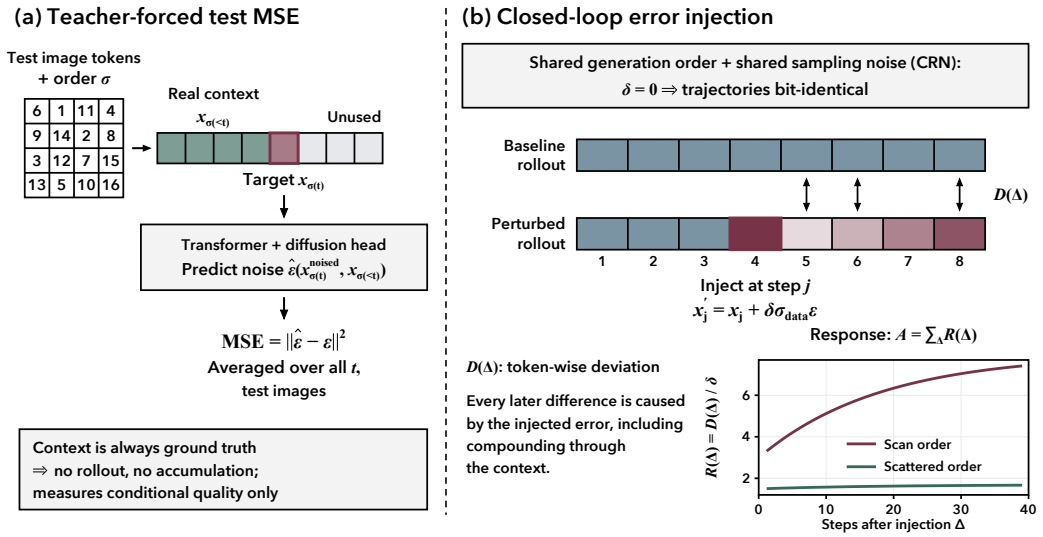

Figure 4: How the two stage-specific quantities are measured. **(a)** Teacher-forcing test MSE: the tokens of a test image are arranged in the inference order Π; every position is predicted from its *ground-truth* prefix and the diffusion noise-prediction MSE is averaged over positions and images. No rollout is involved, so this isolates the quality of the learned conditionals. **(b)** Closed-loop error injection: two rollouts share the inference order and all diffusion sampling noise; only the token sampled at step t_0 receives a small kick $\delta\,\sigma_{\text{data}}\,\varepsilon$. The token-wise deviation $D(\Delta)$ at every later step is caused solely by the injected error, including its compounding through the context; $R(\Delta) = D(\Delta)/\delta$ and $A = \sum_{\Delta} R(\Delta)$ summarize the propagation.

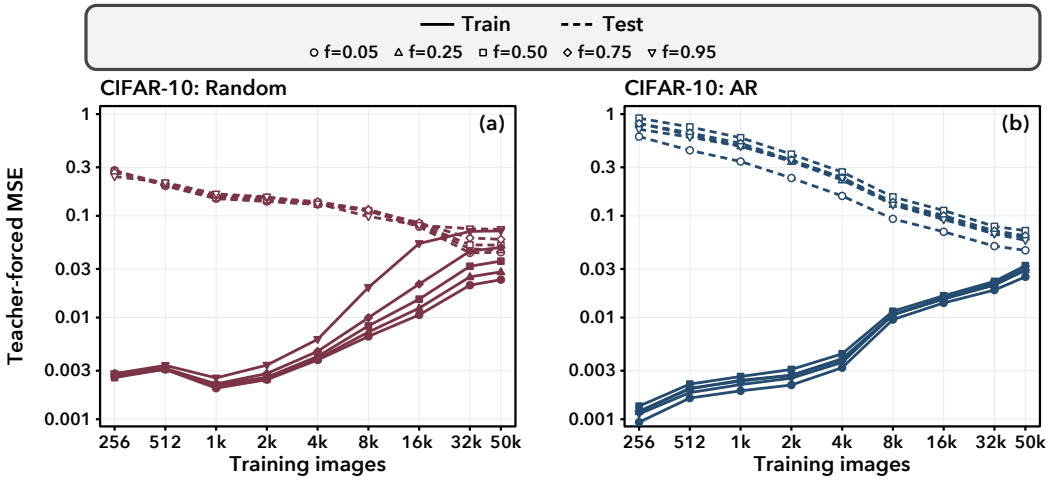

Figure 5: Counterpart of Figure 2(a, d) on CIFAR-10 at MAR-Base (206M): teacher-forcing diffusion MSE on training images (solid) and test images (dashed) as a function of the training-set size, split by the masked fraction f, for **(a)** the AR model and **(b)** the random model. Same update budget and vertical scale as Figure 2.

A.5 Sample-Level Memorization on Images

Memorization score. For each model we draw 10,000 class-balanced samples with the settings used for FID (EMA weights of the best checkpoint, cfg 1.0, inference order equal to training order), embed them together with the training images by DINOv2 ViT-B/14 (CLS token, $32 \to 224$ bicubic), and take for every sample the Euclidean distance to its nearest training image. The memorization score is the mean of these distances, so lower means closer to the training set. Its scale is calibrated

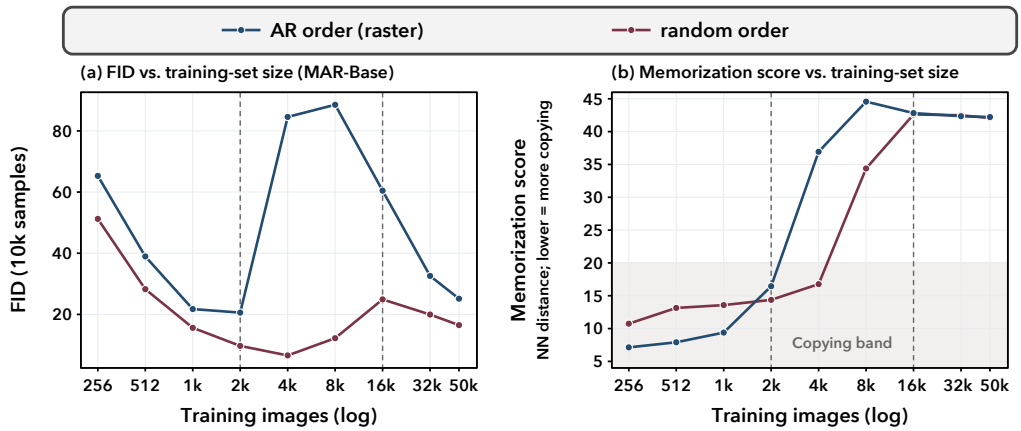

Figure 6: Data-regime sweep on CIFAR-10 at fixed capacity (MAR-Base, 156,000 gradient updates per run): **(a)** FID and **(b)** memorization score for the AR and random models as the training set grows from 256 to 50,000 images. The two dashed lines mark the boundaries between the three data regimes discussed in the text; the shaded band is the copying regime calibrated by the small-data runs.

by the runs themselves: models trained on a few hundred images, whose samples are visibly copies, score 7–17, whereas models that generalize saturate near 42; the shaded band in Figure 6 marks this copying regime. The score is a population statistic and must be read together with FID: a low score with good FID is copying, a high score with good FID is generalization, and a high score with bad FID is drift.

Three data regimes. Figure 6 sweeps the training-set size from 256 to 50,000 images at a fixed update budget (MAR-Base, 156,000 updates per run); the dashed lines at 2k and 16k separate three regimes. (i) *Copying* ($n \leq$ 2k): both models sit inside the copying band, but the AR model sits deeper in it (score 7.1–9.4 against 10.7–13.6 for the random model at $n \leq$ 1k), i.e. it replicates its training images more directly. FID does not measure generalization here: it improves with n simply because a larger training subset, reproduced, lies closer to the full training set. (ii) *Transition* (2k$< n \leq$ 16k): the AR model leaves the copying band abruptly between 2k and 4k (score 16.5 $\rightarrow$ 36.9) and its FID collapses at the same point (20.6 $\rightarrow$ 84.6, and 88.5 at 8k), so its samples are neither copies nor good images. The random model stays in the band until 4k, where its FID of 6.6 equals the FID of those 4096 real images against the full training set (it reproduces its training subset rather than generalizing better), and then leaves the band between 4k and 8k with only a mild FID bump (peak 24.9 at 16k). (iii) *Generalization* ($n \geq$ 16k): the scores of both models saturate near 42 and are indistinguishable from each other, while FID improves monotonically with data and the random model stays ahead (16.5 vs. 25.1 at 50k). Random-order training therefore reduces sample-level replication when data are scarce and removes the unstable transition between memorizing and generalizing; on the full training set neither model copies, and the conditional-level overfitting of Table 2 shows up as drift rather than as replication.

A.6 Image Inference Stage: Response Curves, Spatial Maps, and Caveats

Closed-loop injection. Two generation trajectories share the same inference order and all diffusion sampling noise (a $\delta = 0$ control yields bit-identical trajectories); only the token sampled at one step t_0 is perturbed by a small kick $\delta\,\sigma_{\text{data}}\,\boldsymbol{\varepsilon}$ with $\delta = 0.05$ and $\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$, so any later difference between the two trajectories is caused by the injected shift. The response $R(\Delta)$ is the deviation of the token generated Δ steps after the injection in units of the kick, $A = \sum_{\Delta \geq 1} R(\Delta)$ its total and $\bar{R}$ its mean per downstream step, averaged over 8 injection steps and 256 trajectory pairs. All measurements use the random model, whose conditional quality is order-independent (Table 2), so differences across inference orders reflect propagation alone.

Figure 8 shows the closed-loop response of Section 4.2.2 as a function of the number of steps after the injection and of the grid offset from the injected cell. Table 7 lists the full statistics: under scan



Figure 7: Samples of the random-order CIFAR-10 model under **(a)** AR and **(b)** random inference order; each column shares the class label and sampling seed. Red frames mark collapsed samples (near-constant images, pixel standard deviation below 25 on the 0–255 scale).

Table 7: Full closed-loop injection statistics for the CIFAR-10 random model (MAR-Base) under four inference orders ($\delta = 0.05$). R: response in units of the kick, averaged over $\Delta = 1..8$ (immediately after injection) and over the final 8 generation steps; A: total response; $\bar{R}$: mean response per downstream step; excess corr.: mean off-diagonal inter-token correlation of the generated samples minus that of real CIFAR-10 test images (0.29). The last column repeats the FID of the same model under the same inference order (from Table 1): stronger propagation goes with worse FID.

inference order	R (first 8)	R (final 8)	A	$\bar{R}$	excess corr.	FID
random	1.36	1.83	84.8	1.52	+0.08	14.26
fixed-random	1.41	1.76	82.9	1.48	+0.07	14.25
reverse	3.18	6.57	236	5.40	+0.19	28.45
AR	3.11	**8.45**	**281**	**6.50**	**+0.29**	**40.33**

orders the response *grows* along the generation, from about 3 times the kick immediately after the injection to 6.6–8.5 times in the final steps, whereas under scattered orders it stays flat at 1.4–1.8; the total A orders the four inference orders exactly as FID does. Two caveats apply to these statistics. The response is sublinear in the kick size δ, so A is comparable only at a fixed δ; and it is comparable only across inference orders of the same model, since across models it is confounded with the per-step error magnitude. The excess correlation is computed by standardizing every generated token by the per-position, per-channel mean and standard deviation of the real test images, computing for each channel the 64×64 Pearson correlation matrix between token positions, averaging it over channels, and taking the mean of its off-diagonal entries; the value for real test images is 0.29. Under AR generation the excess is present at every grid distance, including the most distant token pairs.

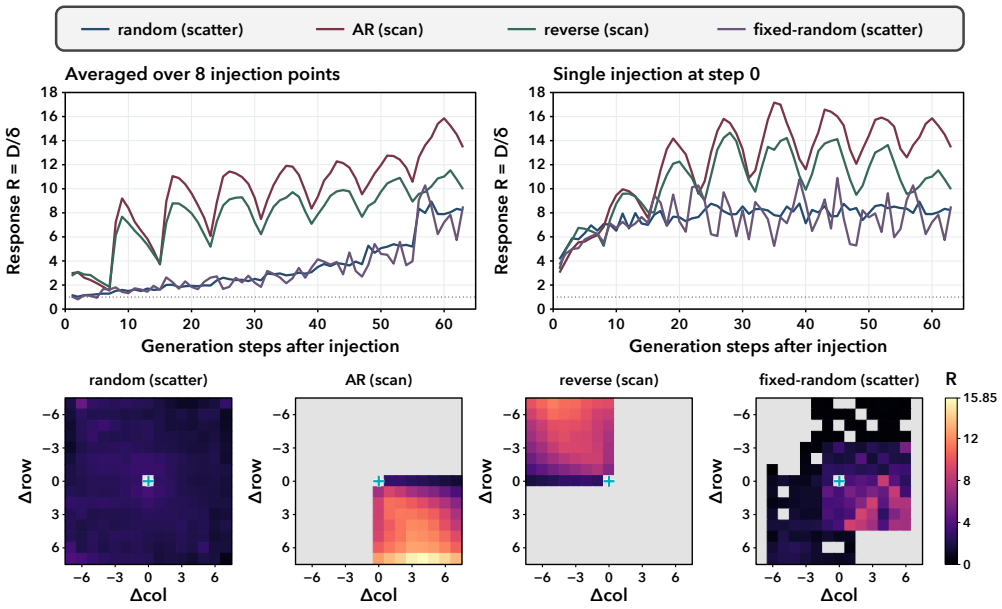

Figure 8: Closed-loop error injection on the CIFAR-10 random model (MAR-Base). **Top:** response $R(\Delta)$ versus steps after injection — scan orders amplify the injected error along the generation, scattered orders do not. **Bottom:** response as a function of the grid offset from the injected cell — under AR the influence grows along the scan direction, under scattered orders it is uniform and isotropic (grey cells are unreachable under the given order).

A.7 Text Metrics: Generation Perplexity, Sample Entropy, and MAUVE

Setup details. test bpb is measured on 21M validation tokens; the training-set value uses a random sample of the training documents with the same protocol. For end-to-end quality, every cell draws 128 unconditional samples of 256 tokens at temperature 1 under the given inference order, read back in natural order. The same pattern of Table 2 holds for 12-layer models trained with the same recipe (AR model 0.848 vs. random model 1.156 under AR; train/val gaps $\leq 1.4\%$).

Generation perplexity is $\mathbb{E}_{p_\theta}[-\log p_{\text{oracle}}]$ of the samples under an independently trained model of the same architecture and tokenizer, under which real validation text scores 16.0. Because it is a reverse cross-entropy, $H(p_\theta) + \text{KL}(p_\theta \| p_{\text{oracle}})$, it pays for the model's own entropy and rewards degeneration, and the literature pairs it with the *sample entropy*, the mean unigram Shannon entropy per sample (real text 4.77). Section A.7 reports both for every cell of the training-order $\times$ generation-order matrix. MAUVE is computed by embedding both sample sets with GPT-2-large, jointly quantizing them into 25 clusters, and scoring the area under the divergence frontier of the two cluster histograms, averaged over 5 k-means seeds; the reference is the same 128 held-out chunks throughout, and two further disjoint real sets of that size score 0.893 and 0.885 against it. Under a paired 96-of-128 subsampling test, the MAUVE gap between fixed-random and AR generation for the random model is positive in 100% of replicates, and the gap between random and AR in 97%. The samples themselves show the failure mode that generation perplexity rewards: the AR samples of the random model have entropy 4.18 and a 19% 4-gram repetition rate, against 4.85–4.86 and 0.2% under the scattered orders, and the fixed-random model has entropy 1.2–1.5 under every order.

	random	AR	reverse	fixed-random
random	152.3 / 4.85	69.7 / 4.18	137.6 / 4.55	163.5 / 4.86
AR	2252 / 4.81	**35.1** / 4.81	1171 / 4.80	2323 / 4.82
reverse	2402 / 4.81	1230 / 4.81	**48.6** / 4.83	2572 / 4.83
fixed-random	14.5† / 1.51	7.8† / 1.19	8.4† / 1.30	12.6† / 1.40

Table 8: Generation perplexity / sample entropy of text samples from the 24-layer ClimbMix models (rows: training order; columns: inference order). Perplexity is measured under an independent oracle; entropy is the mean per-sample unigram entropy in nats. Real validation text: 16.0 / 4.77. †The fixed-random model degenerates into repetition under every order (entropy 1.2–1.5), so its low perplexity does not indicate quality.

A.8 Inference Stage on Text: Error Accumulation and a Second Two-Sample Distance

Section 4.2.2 measures shift accumulation on images. This section runs the same injection experiment on the 24-layer text models and asks what accounts for the MAUVE ordering of Table 1. Throughout, the model is the random-order one, whose teacher-forcing bpb is flat across inference orders (0.907–0.939, first row of Table 2), so its per-step conditional quality is order-independent and any terminal difference must come from what happens to an error after it is made. In brief, the experiment reproduces the two-tier split of the image result: one flipped token changes 82–86 of the 256 tokens under a scan order against 24–25 under a scattered order. On text, however, the AR order is the causal order of the data itself, so part of this amplification is intrinsic to language and the number of changed tokens no longer orders the end-to-end metric; the damage of an error, scored by an oracle, does. The rest of this section gives the details.

Closed-loop injection. Discreteness replaces the shared diffusion noise by a maximal coupling: at every step both chains' conditionals are known, so the same token is drawn with probability $1 - \mathrm{TV}$ and the two chains otherwise draw from the residuals. Identical histories therefore give identical tokens — a no-injection control is bit-identical, the analogue of the $\delta = 0$ check — and $R(\Delta) = \mathbb{E}[\mathrm{TV}]$ is exactly the probability that the token generated Δ steps after the injection differs. At step t_0 the perturbed chain resamples from the full conditional. We report $A = \sum_{\Delta \geq 1} R(\Delta)$, the expected number of the 256 tokens changed by one injected error, over 8 injection positions.

Propagation reproduces the image result, but A is the wrong summary on text. Section A.9 shows the same two-tier split as Table 3: one flipped token changes 82–86 tokens under a scan order against 24–25 under a scattered order, and under a scan order the two chains diverge at the very next step (median lag 1) whereas under a scattered order they stay identical for 15–20 steps and 14% of pairs never diverge at all. But A does not order the four cells the way the end-to-end metric does: it ranks reverse (85.9) marginally *worse* than AR (82.3). This is expected on text and is not a defect of the measurement. A is a property of the sampler's dynamics under the given order, and the AR order is the causal order of the data, so a large response there is a property of language itself — an exact model would propagate under AR too. On images, where no scan direction is causal, the two coincide and A alone orders the FID column; on text they must be separated.

The damage of an error, not the number of tokens it changes. We therefore score the *cost* of a propagated error rather than its extent. Both completed texts are read back in natural order and scored by the same independent oracle used for generation perplexity, and we sum the per-position NLL difference over the whole text:

$$\Delta\mathrm{NLL} = \sum_{\text{natural positions } t} \Big[-\log p_{\mathrm{oracle}}(\boldsymbol{x}^+_t \mid \boldsymbol{x}^+_{<t}) + \log p_{\mathrm{oracle}}(\boldsymbol{x}_t \mid \boldsymbol{x}_{<t}) \Big],$$

averaged over injection positions and trajectory pairs, where $\boldsymbol{x}^+$ is the perturbed rollout. A counts how many tokens one error changes; $\Delta\mathrm{NLL}$ measures how much worse it makes the sample. It separates AR from reverse by a factor of two and spans a factor of 16 across the row, and its ordering — random 0.54, fixed-random 2.40, reverse 4.25, AR 8.72 — is the ordering of the MAUVE column of Table 1 up to the two scattered cells, which MAUVE cannot separate at $n = 128$ (their paired difference is positive in only 85% of subsampling replicates). The damage is visible in the samples

inference order	A	median lag	never	ΔNLL	rep-4	$\mathrm{MMD}^2 \times 10^3$	MAUVE	gen ppl
random	24.4	15	0.15	**0.54**	0.002	**10.0**	0.653	152.3
fixed-random	25.3	20	0.14	2.40	0.001	10.6	**0.764**	163.5
reverse	85.9	1	0.05	4.25	0.073	14.5	0.661	137.6
AR	82.3	1	0.10	8.72	0.191	17.6	0.315	**69.7**

Table 9: Closed-loop injection on the 24-layer random-order text model under four inference orders. A: expected number of the 256 tokens changed by one injected error; median lag: generation steps until the two chains first differ; never: fraction of pairs that never diverge; ΔNLL: total oracle NLL damage of one injected error, in nats; rep-4: fraction of repeated 4-grams in the samples of that cell. The last three columns repeat the MMD^2 of Section A.9, the MAUVE of Table 1 and the generation perplexity of Section A.7 for the same cells. ΔNLL orders the two two-sample distances; A does not, and generation perplexity runs the other way.

as degeneration: the 4-gram repetition rate rises from 0.2% under the scattered orders to 7% under reverse and 19% under AR.

A second two-sample distance. Because MAUVE quantises the feature space before taking a divergence, we check the end-to-end ordering with a metric that does not: the unbiased MMD^2 (Gretton et al., 2012) on the same GPT-2-large features, with an RBF kernel whose bandwidth is the median pairwise distance *within the real set* and is therefore fixed across cells. Unlike an f-divergence it is an integral probability metric, so it charges absolute displacement of the generated distribution instead of whitening it by the data covariance; two disjoint real sets score 0.6 and 1.2 against the reference, which is the floor at this sample size. Its ordering of the four cells is exactly that of ΔNLL, including the two scattered cells that MAUVE leaves tied. Section A.9 gives the full matrix, which agrees with Table 1 everywhere: single-order models near the floor on their own order (4.3 and 2.1) and collapsed off it (53–66), and the fixed-random model degenerate under every order (28–32).

Reading the two modalities together: scan orders amplify a single error in both, and this is what the framework of Section 3 predicts. What differs is the terminal cost. On images the amplification is itself the whole cost and A alone orders the FID column. On text part of the amplification is the data's own causal structure, and only the damaging part reaches the sample, where it shows up as degeneration into repetition loops — which a likelihood-type score rewards and a two-sample distance penalises. This is why Section A.7 and Table 1 disagree on the random model's row.

A.9 Shuffled LDS with multiple data permutations

Setup. The shuffled LDS of Section 3.2 uses one data permutation shared across all sequences. Here we fix a pool of M permutations, with $M \in \{1, 5, 10, 100, 1000, 10000, 50000\}$. When constructing each data split, each sequence independently draws π uniformly from the pool and is reordered as $\boldsymbol{x}_t^{\mathrm{shuf}} = \boldsymbol{x}_{\pi(t)}$. These assignments remain fixed during training. The data permutation π is hidden from the model and is distinct from the training and inference order Π. We use $T = 16$, $d = 12$, 50,000 training sequences, and 8,192 sequences for each of validation and test. We sweep ten model capacities from 13k to 21.1M parameters and train for up to 5,000 epochs, with checkpoints selected by validation NLL under each model's training order and early stopping after 100 epochs without improvement. Both models are evaluated under the AR order. We report teacher-forcing test NLL per token, since the observed distribution is a mixture over data permutations and its entropy is not evaluated exactly.

Results. Figure 9 shows that random-order training attains lower test NLL at larger capacities for every pool size, while AR training can be better for smaller models. At 3.2M parameters and $M = 100$, random-order training achieves 4.744 versus 6.525 for AR; for $M \geq 1000$, its advantage at this capacity is about 1.3 nats per token. For $M = 1$, however, the lowest test NLL across the tested capacities is nearly identical for the two methods (3.527 for AR and 3.526 for random order). The training benefit therefore extends beyond a single shared data permutation, but remains dependent on model capacity and the permutation pool.

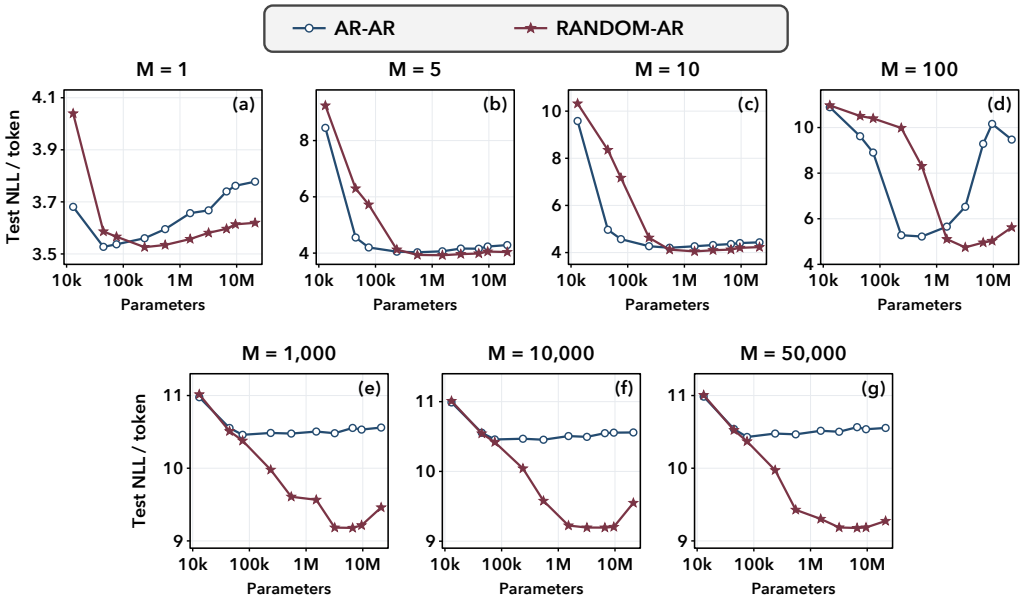

Figure 9: Teacher-forcing test NLL per token versus model capacity on shuffled LDS with M data permutations. Both models are evaluated under the AR order: AR–AR denotes AR training and RANDOM–AR denotes random-order training. Each panel corresponds to one pool size and uses its own vertical scale. Each configuration is a single run.

	random	AR	reverse	fixed-random
random	**10.0**	17.6	14.5	10.6
AR	65.6	**4.3**	53.5	63.1
reverse	63.9	58.4	**2.1**	60.0
fixed-random	28.3	31.9	30.1	30.9

Table 10: Unbiased MMD^2 ($\times 10^3$, lower is better) between generated and real ClimbMix text (24-layer models) (rows: training order; columns: inference order), the check on Table 1. GPT-2-large features, RBF kernel with a bandwidth fixed across all cells. Two disjoint real sets of the same size score 0.6 and 1.2 against the reference, so that range is the floor at $n = 128$, not 0.

B Proofs for the controlled LDS analysis

This section compares AR and random training under the same conditional distribution, error measure, and bias constraint. We first establish a common error lower bound determined by the information in the observations, then compare the information induced by the two orders, and finally discuss extensions to exact KL and joint KL.

B.1 Proof of Theorem 2

B.1.1 Definition of conditional KL

Both training methods estimate the same q_k and are therefore evaluated under the same true input distribution. Let $\boldsymbol{C}_t = \mathrm{Cov}(\boldsymbol{x}_t)$. Then

$$\boldsymbol{C}_0 = \boldsymbol{I}_d, \qquad \boldsymbol{C}_t = \boldsymbol{F}\boldsymbol{C}_{t-1}\boldsymbol{F}^\top + \sigma^2 \boldsymbol{I}_d. \tag{11}$$

For the fitted distribution $\widehat{p}_k(\cdot \mid \boldsymbol{x}) = \mathcal{N}(\widehat{\boldsymbol{B}}_k \boldsymbol{x}, \widehat{\boldsymbol{S}}_k)$, where $\widehat{\boldsymbol{S}}_k \succ 0$, define

$$K_k = \frac{1}{T-k} \sum_{t=0}^{T-k-1} \mathbb{E}_{\boldsymbol{x}\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{C}_t)} \mathrm{KL}(q_k(\cdot \mid \boldsymbol{x}) \,\|\, \widehat{p}_k(\cdot \mid \boldsymbol{x})) \,. \tag{12}$$

The evaluation inputs are independent of the training set. The expected conditional forward KL is

$$\mathcal{K}_{\mathrm{fwd}}^{k} = \mathbb{E}_{\mathrm{train}}[K_k],$$

where the expectation includes the training data and training orders; for shuffled LDS, it also averages over the uniformly sampled data permutation π, held fixed across sequences and epochs.

B.1.2 Bounded bias derivative

To state the bias condition, arrange all entries of $\boldsymbol{B}_k$ and the distinct entries of the symmetric matrix $\boldsymbol{S}_k$ into a parameter vector $\boldsymbol{\theta}$, with true value $\boldsymbol{\theta}_0$ and estimator $\widehat{\boldsymbol{\theta}}$. The vector $\boldsymbol{\theta}$ represents the conditional distribution parameters to be estimated, rather than neural network weights. Define

$$\boldsymbol{b}(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{\theta}}[\widehat{\boldsymbol{\theta}}] - \boldsymbol{\theta}, \qquad \dot{\boldsymbol{b}} = \left.\frac{\partial \boldsymbol{b}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right|_{\boldsymbol{\theta}=\boldsymbol{\theta}_0}. \tag{13}$$

Here, $\boldsymbol{b}$ is the bias averaged over the training data and orders, and $\dot{\boldsymbol{b}}$ is its derivative with respect to the true parameter. For the conditional forward KL K_k defined in Section B.1.1, write

$$\boldsymbol{G}_k = \left.\nabla_{\widehat{\boldsymbol{\theta}}}^{2} K_k(\widehat{\boldsymbol{\theta}})\right|_{\widehat{\boldsymbol{\theta}}=\boldsymbol{\theta}_0}, \qquad \|\boldsymbol{v}\|_{\boldsymbol{G}_k} = \sqrt{\boldsymbol{v}^\top \boldsymbol{G}_k \boldsymbol{v}}. \tag{14}$$

The matrix $\boldsymbol{G}_k$ is the Hessian of KL with respect to the fitted parameters; its positive definiteness is established in Section B.1.3. Assume that both training methods satisfy the same relative bound $0 \le \eta < 1$:

$$\|\dot{\boldsymbol{b}}\boldsymbol{v}\|_{\boldsymbol{G}_k} \le \eta \|\boldsymbol{v}\|_{\boldsymbol{G}_k} \qquad \text{for all } \boldsymbol{v}. \tag{15}$$

This condition limits the rate of change of the bias while allowing $\boldsymbol{b}(\boldsymbol{\theta}_0) \neq \boldsymbol{0}$.

B.1.3 The quadratic term of KL

To express the conditional KL in terms of parameter errors, write $\Delta\boldsymbol{B} = \widehat{\boldsymbol{B}}_k - \boldsymbol{B}_k$ and $\Delta\boldsymbol{S} = \widehat{\boldsymbol{S}}_k - \boldsymbol{S}_k$. By the Gaussian KL formula, for a fixed input $\boldsymbol{x}$,

$$\begin{aligned} &\mathrm{KL}(q_k(\cdot \mid \boldsymbol{x}) \,\|\, \widehat{p}_k(\cdot \mid \boldsymbol{x})) \\ &\quad = \frac{1}{2}\left[\log \frac{\det \widehat{\boldsymbol{S}}_k}{\det \boldsymbol{S}_k} - d + \mathrm{tr}(\widehat{\boldsymbol{S}}_k^{-1}\boldsymbol{S}_k) + \boldsymbol{x}^\top \Delta\boldsymbol{B}^\top \widehat{\boldsymbol{S}}_k^{-1} \Delta\boldsymbol{B}\boldsymbol{x}\right]. \end{aligned} \tag{16}$$

Average the input covariances $\boldsymbol{C}_t$ over all available starting times, and denote the result by

$$\overline{\boldsymbol{C}}_k = \frac{1}{T-k} \sum_{t=0}^{T-k-1} \boldsymbol{C}_t. \tag{17}$$

Averaging Eq. equation 16 gives

$$K_k = \frac{1}{2}\left[\log \frac{\det \widehat{\boldsymbol{S}}_k}{\det \boldsymbol{S}_k} - d + \mathrm{tr}(\widehat{\boldsymbol{S}}_k^{-1}\boldsymbol{S}_k) + \mathrm{tr}\Big(\widehat{\boldsymbol{S}}_k^{-1}\Delta\boldsymbol{B}\overline{\boldsymbol{C}}_k\Delta\boldsymbol{B}^\top\Big)\right]. \tag{18}$$

Expand Eq. equation 18 to second order in the fitted parameters $\widehat{\boldsymbol{\theta}}$ around the true value $\boldsymbol{\theta}_0$. Both KL and its first derivative vanish at the true value, and the quadratic term is

$$\begin{aligned} \mathcal{Q}_k &= \frac{1}{2}\,\mathrm{tr}\big(\boldsymbol{S}_k^{-1}\Delta\boldsymbol{B}\overline{\boldsymbol{C}}_k\Delta\boldsymbol{B}^\top\big) + \frac{1}{4}\,\mathrm{tr}\big(\boldsymbol{S}_k^{-1}\Delta\boldsymbol{S}\boldsymbol{S}_k^{-1}\Delta\boldsymbol{S}\big) \\ &= \frac{1}{2}(\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}_0)^\top \boldsymbol{G}_k (\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}_0). \end{aligned} \tag{19}$$

Since $\boldsymbol{S}_k, \overline{\boldsymbol{C}}_k \succ 0$, we have $\boldsymbol{G}_k \succ 0$. Near the true parameter,

$$K_k = \mathcal{Q}_k + O(\|\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}_0\|^3). \tag{20}$$

In a sufficiently small neighborhood, the remainder is bounded by a constant times the cube of the parameter error. We next establish a finite-sample lower bound on $\mathbb{E}_{\mathrm{train}}\mathcal{Q}_k$, then connect it to exact KL in Section B.1.7.

B.1.4 JOINT FISHER INFORMATION AND A COMMON LOWER BOUND

The training methods differ in the observations available to them. We represent these observations through joint Fisher information and derive a common lower bound.

Parameter space and regularity conditions. To define derivatives in all directions of $\boldsymbol{\theta} = (\boldsymbol{B}_k, \boldsymbol{S}_k)$, allow the noise covariance near the true value to vary from $\sigma^2 \boldsymbol{I}_d$ to a general positive definite matrix $\boldsymbol{\Omega}$, and require the mapping

$$(\boldsymbol{F}, \boldsymbol{\Omega}) \longmapsto \left(\boldsymbol{F}^k, \sum_{r=0}^{k-1} \boldsymbol{F}^r \boldsymbol{\Omega} (\boldsymbol{F}^r)^\top \right) = (\boldsymbol{B}_k, \boldsymbol{S}_k) \tag{21}$$

to have an invertible derivative matrix at the true value. Thus, nearby conditional parameters uniquely determine a nearby joint LDS distribution. For $k = 1$, we have $\boldsymbol{B}_1 = \boldsymbol{F}$ and $\boldsymbol{S}_1 = \boldsymbol{\Omega}$, so this condition holds automatically; for general k, the results require it as an additional condition. For example, when $\boldsymbol{F} = a\boldsymbol{I}_d$ and $0 < |a| < 1$, the derivative matrix is block lower triangular, with diagonal blocks $\boldsymbol{H} \mapsto k a^{k-1} \boldsymbol{H}$ and $\boldsymbol{L} \mapsto (\sum_{r=0}^{k-1} a^{2r}) \boldsymbol{L}$, both of which are invertible. Here, $\boldsymbol{H}$ is a perturbation direction of $\boldsymbol{F}$, and $\boldsymbol{L}$ is a symmetric perturbation direction of $\boldsymbol{\Omega}$.

Unlike the expansion in Section B.1.3 with respect to the fitted parameters, Fisher information differentiates along the true LDS parameter family above. We assume that the estimator is a fixed function of the training observations, does not depend on the unknown true value, has finite second moments, and permits differentiation under the integral. Assumption equation 15 is imposed on this parameter space; all information matrices and bias derivatives are evaluated at $\boldsymbol{\theta}_0$. If the estimator uses additional restrictions, such as a known noise covariance, the lower bound must be recomputed in the corresponding parameter space.

Exposure sets and their information. We retain the single-token conditioning and fixed-lag observation assumptions from the main text. Fix k and E, let $\boldsymbol{x}_t^{(n)}$ denote token t of sequence n, and define

$$C_n = \bigcup_{e=1}^{E} \big\{ (t, t+k) : \ \boldsymbol{x}_t^{(n)} \to \boldsymbol{x}_{t+k}^{(n)} \text{ are adjacent in epoch } e \big\}, \tag{22}$$

where $0 \le t < T - k$ and the arrow requires the indicated forward direction. The set C_n contains the pairs of original indices exposed in sequence n, with repeated pairs recorded only once. Below, C denotes such a set for a single sequence; it is distinct from the covariance matrix $\boldsymbol{C}_t$. Concatenate the values at the endpoints in C in original time order, removing duplicates, to form $\boldsymbol{w}_C$; let $\boldsymbol{w}_n$ denote the values selected by C_n from sequence n. Even when different sequences have the same C_n, their values remain distinct samples. Only these indices and endpoint values are used to estimate q_k; no observations of other tokens are additionally used.

Write the joint density of a nonempty $\boldsymbol{w}_C$ as $f_{\boldsymbol{\theta}}(\boldsymbol{w}_C)$. Its Gaussian covariance is determined by

$$\operatorname{Cov}(\boldsymbol{x}_t, \boldsymbol{x}_{t'}) = \boldsymbol{C}_t (\boldsymbol{F}^{t'-t})^\top \quad (t \le t'), \qquad \boldsymbol{C}_t = \boldsymbol{F}^t (\boldsymbol{F}^t)^\top + \sum_{r=0}^{t-1} \boldsymbol{F}^r \boldsymbol{\Omega} (\boldsymbol{F}^r)^\top \tag{23}$$

and thus retains shared tokens and correlations between different pairs. Define the score and joint Fisher information as

$$\boldsymbol{u}_C = \nabla_{\boldsymbol{\theta}} \log f_{\boldsymbol{\theta}}(\boldsymbol{w}_C), \qquad \boldsymbol{J}_k(C) = \mathbb{E}[\boldsymbol{u}_C \boldsymbol{u}_C^\top], \qquad \boldsymbol{u}_\varnothing = \boldsymbol{0}, \quad \boldsymbol{J}_k(\varnothing) = \boldsymbol{0}. \tag{24}$$

The score is the derivative of the observed log density with respect to the true parameter; $\boldsymbol{J}_k(C)$ is its second moment. Under a random design, the average information from one sequence is $\boldsymbol{J} = \mathbb{E}_C[\boldsymbol{J}_k(C)]$.

The biased Cramér–Rao inequality and a common lower bound. Let $s \in \{\mathrm{ordinary}, \mathrm{shuf}\}$ denote the data setting and $o \in \{\mathrm{AR}, \mathrm{rand}\}$ the training method. For $\boldsymbol{J}$ and $\mathcal{B}$, subscripts indicate the lag and data setting, and superscripts indicate the training method.

Proposition 1 (Common error lower bound). For N independent training sequences, under the assumptions and regularity conditions above, if $\boldsymbol{J}^o_{k,s}(E) \succ 0$, then

$$\mathbb{E}_{\text{train}}\mathcal{Q}_{k,s,o}(E) \geq \mathcal{B}^o_{k,s}(E), \qquad \mathcal{B}^o_{k,s}(E) = \frac{(1-\eta)^2}{2N}\,\mathrm{tr}\big(\boldsymbol{G}_k \boldsymbol{J}^o_{k,s}(E)^{-1}\big)\,. \tag{25}$$

The two training methods share $\boldsymbol{G}_k$ and η; their lower bounds differ through the average joint information $\boldsymbol{J}^o_{k,s}(E)$.

Proof. For N independent training sequences, their observed values are conditionally independent given all exposure sets. The total score is therefore

$$\boldsymbol{u} = \nabla_{\boldsymbol{\theta}} \log \prod_{n=1}^{N} f_{\boldsymbol{\theta}}(\boldsymbol{w}_n \mid C_n) = \sum_{n=1}^{N} \boldsymbol{u}_n, \qquad \boldsymbol{u}_n = \nabla_{\boldsymbol{\theta}} \log f_{\boldsymbol{\theta}}(\boldsymbol{w}_n \mid C_n). \tag{26}$$

For any fixed observation set, the density integrates to 1, so

$$\mathbb{E}_{\boldsymbol{\theta}} \boldsymbol{u}_C = \int \nabla_{\boldsymbol{\theta}} \log f_{\boldsymbol{\theta}}(\boldsymbol{w}_C) f_{\boldsymbol{\theta}}(\boldsymbol{w}_C)\, d\boldsymbol{w}_C = \nabla_{\boldsymbol{\theta}} \int f_{\boldsymbol{\theta}}(\boldsymbol{w}_C)\, d\boldsymbol{w}_C = \boldsymbol{0}.$$

Thus, when $n \neq m$, we have $\mathbb{E}[\boldsymbol{u}_n \boldsymbol{u}_m^\top \mid C_1, \ldots, C_N] = \boldsymbol{0}$, and hence

$$\mathbb{E}\boldsymbol{u} = \boldsymbol{0}, \qquad \mathrm{Cov}(\boldsymbol{u}) = \sum_{n=1}^{N} \mathbb{E}[\boldsymbol{u}_n \boldsymbol{u}_n^\top] = N\boldsymbol{J}. \tag{27}$$

This identity also holds when all sequences share a random π: first condition on the design, then average over it. The number of training epochs E affects $\boldsymbol{J}$ through the exposure sets; it does not provide NE independent sequences.

Denote all observations and the design collectively by $\boldsymbol{w}$; the integrals below include summation over the discrete design. Since the estimation rule does not depend on the unknown true value,

$$\begin{aligned} \frac{\partial}{\partial \boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{\theta}}[\widehat{\boldsymbol{\theta}}] &= \int \widehat{\boldsymbol{\theta}}(\boldsymbol{w}) \big(\nabla_{\boldsymbol{\theta}} f_{\boldsymbol{\theta}}(\boldsymbol{w})\big)^\top d\boldsymbol{w} \\ &= \mathbb{E}_{\boldsymbol{\theta}}[\widehat{\boldsymbol{\theta}} \boldsymbol{u}^\top] = \mathrm{Cov}(\widehat{\boldsymbol{\theta}}, \boldsymbol{u}) = \boldsymbol{I} + \dot{\boldsymbol{b}}. \end{aligned} \tag{28}$$

The last step follows from $\mathbb{E}_{\boldsymbol{\theta}} \widehat{\boldsymbol{\theta}} = \boldsymbol{\theta} + \boldsymbol{b}(\boldsymbol{\theta})$. Let $\boldsymbol{A} = \boldsymbol{I} + \dot{\boldsymbol{b}}$ and temporarily assume $\boldsymbol{J} \succ 0$; positive definiteness of the information matrices being compared is verified in Section B.1.5. Taking $\boldsymbol{M} = \boldsymbol{A}(N\boldsymbol{J})^{-1}$ gives

$$\mathrm{Cov}\Big(\widehat{\boldsymbol{\theta}} - \mathbb{E}\widehat{\boldsymbol{\theta}} - \boldsymbol{M}\boldsymbol{u}, \boldsymbol{u}\Big) = \boldsymbol{A} - \boldsymbol{M}(N\boldsymbol{J}) = \boldsymbol{0}.$$

Define the residual vector

$$\boldsymbol{r} = \widehat{\boldsymbol{\theta}} - \mathbb{E}\widehat{\boldsymbol{\theta}} - \boldsymbol{A}(N\boldsymbol{J})^{-1}\boldsymbol{u}.$$

Using $\mathrm{Cov}(\widehat{\boldsymbol{\theta}}, \boldsymbol{u}) = \boldsymbol{A}$ and $\mathrm{Cov}(\boldsymbol{u}) = N\boldsymbol{J}$, the full expansion is

$$\begin{aligned} \mathrm{Cov}(\boldsymbol{r}) =\ & \mathrm{Cov}(\widehat{\boldsymbol{\theta}}) - \mathrm{Cov}(\widehat{\boldsymbol{\theta}}, \boldsymbol{u})(N\boldsymbol{J})^{-1}\boldsymbol{A}^\top \\ & - \boldsymbol{A}(N\boldsymbol{J})^{-1}\,\mathrm{Cov}(\boldsymbol{u}, \widehat{\boldsymbol{\theta}}) + \boldsymbol{A}(N\boldsymbol{J})^{-1}\,\mathrm{Cov}(\boldsymbol{u})(N\boldsymbol{J})^{-1}\boldsymbol{A}^\top \\ =\ & \mathrm{Cov}(\widehat{\boldsymbol{\theta}}) - \boldsymbol{A}(N\boldsymbol{J})^{-1}\boldsymbol{A}^\top - \boldsymbol{A}(N\boldsymbol{J})^{-1}\boldsymbol{A}^\top \\ & + \boldsymbol{A}(N\boldsymbol{J})^{-1}(N\boldsymbol{J})(N\boldsymbol{J})^{-1}\boldsymbol{A}^\top \\ =\ & \mathrm{Cov}(\widehat{\boldsymbol{\theta}}) - \frac{1}{N}\boldsymbol{A}\boldsymbol{J}^{-1}\boldsymbol{A}^\top \succeq \boldsymbol{0}. \end{aligned} \tag{29}$$

The last step uses the positive semidefiniteness of covariance matrices. Thus,

$$\mathrm{Cov}(\widehat{\boldsymbol{\theta}}) \succeq \frac{1}{N}(\boldsymbol{I} + \dot{\boldsymbol{b}})\boldsymbol{J}^{-1}(\boldsymbol{I} + \dot{\boldsymbol{b}})^\top.$$

By Assumption equation 15,

$$\begin{aligned} \|(\boldsymbol{I} + \dot{\boldsymbol{b}})\boldsymbol{v}\|_{\boldsymbol{G}_k} \geq \|\boldsymbol{v}\|_{\boldsymbol{G}_k} - \|\dot{\boldsymbol{b}}\boldsymbol{v}\|_{\boldsymbol{G}_k} \geq (1-\eta)\|\boldsymbol{v}\|_{\boldsymbol{G}_k}, \\ (\boldsymbol{I} + \dot{\boldsymbol{b}})^\top \boldsymbol{G}_k (\boldsymbol{I} + \dot{\boldsymbol{b}}) \succeq (1-\eta)^2 \boldsymbol{G}_k. \end{aligned} \tag{30}$$

Combining this with the bias–variance decomposition yields

$$\begin{aligned}\mathbb{E}_{\text{train}}\mathcal{Q}_k &= \frac{1}{2}\boldsymbol{b}(\boldsymbol{\theta}_0)^\top \boldsymbol{G}_k \boldsymbol{b}(\boldsymbol{\theta}_0) + \frac{1}{2}\operatorname{tr}\Big(\boldsymbol{G}_k \operatorname{Cov}(\widehat{\boldsymbol{\theta}})\Big) \\ &\geq \frac{1}{2N}\operatorname{tr}\Big[\boldsymbol{J}^{-1}(\boldsymbol{I}+\dot{\boldsymbol{b}})^\top \boldsymbol{G}_k (\boldsymbol{I}+\dot{\boldsymbol{b}})\Big] \\ &\geq \frac{(1-\eta)^2}{2N}\operatorname{tr}(\boldsymbol{G}_k \boldsymbol{J}^{-1}).\end{aligned} \tag{31}$$

□

B.1.5 Comparing information in LDS and shuffled LDS

Eq. equation 25 reduces the comparison of lower bounds to a comparison of information matrices. We first show that additional observations do not decrease information, then use random orders to show that the average information increases strictly.

Lemma 1 (Additional observations do not decrease information). If $C \subseteq C'$, then

$$\boldsymbol{J}_k(C') \succeq \boldsymbol{J}_k(C).$$

Proof. Denote the additional endpoint values by $\boldsymbol{v}$. By marginalization and interchange of differentiation and integration,

$$\begin{aligned}\boldsymbol{u}_C &= \frac{\nabla_{\boldsymbol{\theta}} \int f_{\boldsymbol{\theta}}(\boldsymbol{w}_C, \boldsymbol{v})\, d\boldsymbol{v}}{f_{\boldsymbol{\theta}}(\boldsymbol{w}_C)} \\ &= \int \nabla_{\boldsymbol{\theta}} \log f_{\boldsymbol{\theta}}(\boldsymbol{w}_C, \boldsymbol{v})\, f_{\boldsymbol{\theta}}(\boldsymbol{v} \mid \boldsymbol{w}_C)\, d\boldsymbol{v} = \mathbb{E}[\boldsymbol{u}_{C'} \mid \boldsymbol{w}_C].\end{aligned} \tag{32}$$

Thus $\mathbb{E}[\boldsymbol{u}_{C'} - \boldsymbol{u}_C \mid \boldsymbol{w}_C] = \boldsymbol{0}$, so the cross terms vanish and

$$\begin{aligned}\boldsymbol{J}_k(C') - \boldsymbol{J}_k(C) &= \mathbb{E}[(\boldsymbol{u}_{C'} - \boldsymbol{u}_C)(\boldsymbol{u}_{C'} - \boldsymbol{u}_C)^\top] \\ &= \mathbb{E}[\operatorname{Cov}(\boldsymbol{u}_{C'} \mid \boldsymbol{w}_C)] \succeq \boldsymbol{0}.\end{aligned} \tag{33}$$

□

This identity allows correlated observations and allows additional pairs to provide no additional information.

For a fixed directed pair $(t, t+k)$, the probability that a random order makes it adjacent in one round is $(T-1)!/T! = 1/T$. In particular, the joint information of the pair $(0, k)$ in a nonzero direction $(\boldsymbol{H}, \boldsymbol{L})$ is

$$\operatorname{tr}(\boldsymbol{S}_k^{-1}\boldsymbol{H}\boldsymbol{H}^\top) + \tfrac{1}{2}\operatorname{tr}(\boldsymbol{S}_k^{-1}\boldsymbol{L}\boldsymbol{S}_k^{-1}\boldsymbol{L}) > 0,$$

where $\boldsymbol{H}$ and the symmetric matrix $\boldsymbol{L}$ are perturbation directions of $\boldsymbol{B}_k, \boldsymbol{S}_k$, respectively, and the distribution of $\boldsymbol{x}_0$ does not depend on the parameters. By Eq. equation 33,

$$\boldsymbol{J}^{\text{rand}}_{k,\text{shuf}}(1) \succeq T^{-1}\boldsymbol{J}_k(\{(0,k)\}) \succ \boldsymbol{0}.$$

Let $r_k \in (0,1)$ be the probability that no forward lag-k pair is exposed in one round, C_E the exposure set from the first E rounds, and C^{new} the exposure set from the independent next round. The reverse chronological order ensures $r_k > 0$, while an order exposing $(0,k)$ ensures $r_k < 1$. The event $C_E = \varnothing$ has probability r_k^E; on this event, the conditional mean of the additional information equals the average information from one round. On all other events, the increment is nonnegative by Eq. equation 33, and hence

$$\begin{aligned}&\boldsymbol{J}^{\text{rand}}_{k,\text{shuf}}(E+1) - \boldsymbol{J}^{\text{rand}}_{k,\text{shuf}}(E) \\ &\quad = \mathbb{E}[\boldsymbol{J}_k(C_E \cup C^{\text{new}}) - \boldsymbol{J}_k(C_E)] \succeq r_k^E \boldsymbol{J}^{\text{rand}}_{k,\text{shuf}}(1) \succ \boldsymbol{0}.\end{aligned} \tag{34}$$

Here, the event of an empty exposure set only means that the lag-k restricted estimator has no available observations. Given a fixed π, the orders $\pi \circ \Pi_e$ remain independent and uniform, so the above information under random training is identical in the two data settings; shuffled AR always repeats the same order. Averaging over π drawn uniformly once and shared throughout gives

$$\boldsymbol{J}^{\text{AR}}_{k,\text{shuf}} = \boldsymbol{J}^{\text{rand}}_{k,\text{shuf}}(1).$$

The exposure set of AR does not change with the number of rounds, so E is omitted from its information and lower-bound notation.

B.1.6 THE TWO LOWER-BOUND COMPARISONS

We retain the observation restrictions, bias conditions, and regularity conditions above, with a common $\eta < 1$ for both training protocols.

Shuffled LDS case. Suppose $T \geq 3$, $E > 1$, $1 \leq k < T$, and the mapping in Eq. equation 21 is locally invertible near the true value; this condition holds automatically when $k = 1$. Averaging over π drawn uniformly once and fixed throughout,

$$\mathcal{B}^{\mathrm{rand}}_{k,\mathrm{shuf}}(E) < \mathcal{B}^{\mathrm{AR}}_{k,\mathrm{shuf}}.$$

Proof. By Eq. equation 34, for $E > 1$,

$$\boldsymbol{J}^{\mathrm{rand}}_{k,\mathrm{shuf}}(E) \succ \boldsymbol{J}^{\mathrm{AR}}_{k,\mathrm{shuf}} \quad \Longrightarrow \quad \boldsymbol{J}^{\mathrm{rand}}_{k,\mathrm{shuf}}(E)^{-1} \prec \left(\boldsymbol{J}^{\mathrm{AR}}_{k,\mathrm{shuf}}\right)^{-1}.$$

Taking the trace against the common $\boldsymbol{G}_k \succ 0$ and substituting into Eq. equation 25 proves the claim. □

Ordinary LDS case. Suppose $T \geq 3$. For lag-1 conditional estimation, every finite $E \geq 1$ satisfies

$$\mathcal{B}^{\mathrm{AR}}_{1,\mathrm{ordinary}} < \mathcal{B}^{\mathrm{rand}}_{1,\mathrm{ordinary}}(E).$$

Proof. Natural-order AR exposes all lag-1 pairs, whose endpoints cover the entire sequence. The observations available under random training form at most a subset of these and are empty with probability r_1^E. By Eq. equation 33,

$$\boldsymbol{J}^{\mathrm{rand}}_{1,\mathrm{ordinary}}(E) \preceq (1 - r_1^E)\boldsymbol{J}^{\mathrm{AR}}_{1,\mathrm{ordinary}} \prec \boldsymbol{J}^{\mathrm{AR}}_{1,\mathrm{ordinary}}. \tag{35}$$

Taking inverses and substituting into Eq. equation 25 proves the claim. As $E \to \infty$, random training almost surely covers all pairs, so its lower bound converges to that of AR. □

B.1.7 LOWER BOUNDS FOR THE EXACT CONDITIONAL KL

We retain the conditions of Proposition 1 and abbreviate its lower bound as $\mathcal{B}_k$.

Proposition 2 (Exact conditional KL: remainder control). Fix T, E. If, in addition, $\mathbb{E}_{\mathrm{train}}|K_k - \mathcal{Q}_k| = o(N^{-1})$, then

$$\mathcal{K}^k_{\mathrm{fwd}} \geq \mathcal{B}_k - o(N^{-1}).$$

Proof.

$$\mathcal{K}^k_{\mathrm{fwd}} \geq \mathbb{E}_{\mathrm{train}}\mathcal{Q}_k - \mathbb{E}_{\mathrm{train}}|K_k - \mathcal{Q}_k| \geq \mathcal{B}_k - o(N^{-1}). \tag{36}$$

By Eq. equation 20, a sufficient set of conditions is that the estimates lie in a fixed bounded region near the true value, the smallest eigenvalue of $\widehat{\boldsymbol{S}}_k$ has a uniform positive lower bound, and $\mathbb{E}\|\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}_0\|^3 = o(N^{-1})$. □

The shared π in shuffled AR exposes no lag-k observations with a fixed positive probability; increasing N therefore cannot guarantee the small-error conditions throughout a parameter neighborhood, and the above remainder control must be assumed separately.

Proposition 3 (Exact conditional KL: covariance control). Another sufficient set of conditions is that both training protocols satisfy, almost surely, for a common constant $\beta \geq 1$,

$$\mathbf{0} \prec \widehat{\boldsymbol{S}}_k \preceq \beta \boldsymbol{S}_k. \tag{37}$$

Then

$$K_k \geq \beta^{-2}\mathcal{Q}_k, \qquad \mathcal{K}^k_{\mathrm{fwd}} \geq \beta^{-2}\mathcal{B}_k. \tag{38}$$

Proof. Let λ_j be the eigenvalues of $\boldsymbol{S}_k^{-1/2}\widehat{\boldsymbol{S}}_k\boldsymbol{S}_k^{-1/2}$, so $0 < \lambda_j \le \beta$. The covariance terms in the exact KL and $\mathcal{Q}_k$ are $\frac{1}{2}\sum_j(\log\lambda_j + \lambda_j^{-1} - 1)$ and $\frac{1}{4}\sum_j(\lambda_j - 1)^2$, respectively. Since

$$\log\lambda + \lambda^{-1} - 1 = \begin{cases} \displaystyle\int_1^{\lambda} \frac{u-1}{u^2}\,du, & \lambda \ge 1, \\ \displaystyle\int_{\lambda}^{1} \frac{1-u}{u^2}\,du, & \lambda < 1, \end{cases} \qquad \ge \frac{(\lambda-1)^2}{2\beta^2}, \tag{39}$$

and $\widehat{\boldsymbol{S}}_k^{-1} \succeq \beta^{-1}\boldsymbol{S}_k^{-1}$, Eq. equation 18 and $\beta^{-1} \ge \beta^{-2}$ imply Eq. equation 38. □

This is a finite-sample inequality requiring no small-error approximation; the common factor β^{-2} preserves the ordering of lower bounds in both cases of Theorem 2.

B.1.8 Extension to joint KL

Given an inference order Π, let $\boldsymbol{z}_t = \boldsymbol{y}_{\Pi(t)}$, and let the single-token model be

$$\widehat{p}^{\Pi}(\boldsymbol{z}_{0:T-1}) = \widehat{p}_0(\boldsymbol{z}_0)\prod_{t=1}^{T-1}\widehat{p}_t(\boldsymbol{z}_t \mid \boldsymbol{z}_{t-1}).$$

Denote the true joint density by q. Decomposing each term as

$$\log\frac{q(\boldsymbol{z}_t \mid \boldsymbol{z}_{0:t-1})}{\widehat{p}_t(\boldsymbol{z}_t \mid \boldsymbol{z}_{t-1})} = \log\frac{q(\boldsymbol{z}_t \mid \boldsymbol{z}_{0:t-1})}{q(\boldsymbol{z}_t \mid \boldsymbol{z}_{t-1})} + \log\frac{q(\boldsymbol{z}_t \mid \boldsymbol{z}_{t-1})}{\widehat{p}_t(\boldsymbol{z}_t \mid \boldsymbol{z}_{t-1})},$$

and taking expectations under the true distribution yields

$$\begin{aligned}\mathrm{KL}(q\|\widehat{p}^{\Pi}) = \mathcal{A}_{\pi,\Pi} &+ \mathrm{KL}(q_{\boldsymbol{z}_0}\|\widehat{p}_0) \\ &+ \sum_{t=1}^{T-1}\mathbb{E}_{\boldsymbol{z}_{t-1}}\mathrm{KL}(q(\boldsymbol{z}_t \mid \boldsymbol{z}_{t-1})\|\widehat{p}_t(\boldsymbol{z}_t \mid \boldsymbol{z}_{t-1}))\,,\end{aligned} \tag{40}$$

where $\mathcal{A}_{\pi,\Pi}$ is the sum of the expected log ratios from the first term of the above decomposition. Each term can be written as a conditional KL and is therefore nonnegative and independent of the fitted parameters; $\boldsymbol{z}_{0:t-1}$ denotes the full prefix.

Let $\overline{\mathcal{A}} = \mathbb{E}_{\pi,\Pi}\mathcal{A}_{\pi,\Pi}$.

Corollary 1 (Joint-KL lower bounds). Assume that $\widehat{p}_{k,o}$ for the same forward lag is shared across positions and inference orders, that both training protocols use the same evaluation rule, and that Eq. equation 37 holds with a common β for all relevant lags. For ordinary LDS in natural order,

$$\mathbb{E}_{\text{train}}\,\mathrm{KL}(q\|\widehat{p}_o^{\mathrm{AR}}) \ge \frac{T-1}{\beta^2}\mathcal{B}^o_{1,\text{ordinary}}. \tag{41}$$

For shuffled LDS, Π is drawn independently and uniformly after training. If the above conditions hold for every lag, then

$$\mathbb{E}_{\text{train},\Pi}\,\mathrm{KL}(q\|\widehat{p}_o^{\Pi}) \ge \overline{\mathcal{A}} + \frac{1}{\beta^2}\sum_{k=1}^{T-1}\frac{T-k}{T}\mathcal{B}^o_{k,\text{shuf}}. \tag{42}$$

If the conditions hold only for $k = 1$, retaining this term gives

$$\mathbb{E}_{\text{train},\Pi}\,\mathrm{KL}(q\|\widehat{p}_o^{\Pi}) \ge \overline{\mathcal{A}} + \frac{T-1}{T\beta^2}\mathcal{B}^o_{1,\text{shuf}}.$$

Proof. For ordinary LDS in natural order, $\mathcal{A}_{\pi,\Pi} = 0$, and the expected sum of conditional errors is $(T-1)\mathcal{K}^1_{\text{fwd}}$, yielding Eq. equation 41. Given the training process and π, each directed pair $(t, t+k)$ appears with probability $1/T$, so, in expectation over training, the order average of its total lag-k contribution is

$$\mathbb{E}_{\text{train}}\frac{1}{T}\sum_{t=0}^{T-k-1}\mathbb{E}_{\boldsymbol{x}_t}\mathrm{KL}(q_k(\cdot \mid \boldsymbol{x}_t)\|\widehat{p}_{k,o}(\cdot \mid \boldsymbol{x}_t)) = \frac{T-k}{T}\mathcal{K}^k_{\text{fwd}}.$$

Dropping the nonnegative initial term and reverse conditional errors gives Eq. equation 42; if the conditions hold only for $k = 1$, retain that term. □

All weights are positive, so the corresponding common lower bounds preserve the orderings in Theorems 1 and 2. If Eq. equation 36 is used instead, with T, E fixed and the remainder conditions satisfied for every retained term, the above lower bounds omit β^{-2} and subtract $o(N^{-1})$; the joint KL lower bounds per token are further divided by T.

B.2 LDS WITH SINGLE-TOKEN CONDITIONING

Setup. We test the single-token simplification by restricting each prediction to the preceding token and the source and target positions; the first token uses BOS. Every layer processes each prediction independently, so earlier token values cannot enter through intermediate hidden states. We use ordinary and fixed-shuffled LDS with $T = d = 16$, $\sigma = 0.3$, $\rho(\boldsymbol{F}) = 0.9$, and $\boldsymbol{x}_0 \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_d)$. The training, validation, and test sets contain $5{,}000$, $1{,}000$, and $1{,}000$ sequences, respectively. We sweep ten capacities from 15.2k to 21.1M parameters, train each model for $5{,}000$ epochs, and select checkpoints by validation NLL under the training order. We use batch size 256 and learning rate 3×10^{-4}. Data, model, and permutation seeds are fixed at 0.

Results. Under common AR evaluation, AR training gives lower test joint KL on ordinary LDS at every tested capacity. On shuffled LDS, random training performs better except at the smallest capacity (Figure 10(a,b)). We also directly evaluate forward lag-k conditional KL using $4{,}096$ fresh inputs per time pair, averaged uniformly over starting positions. For this diagnostic, each protocol uses its validation-selected capacity. At $k = 1$, AR and random training give 0.048 and 0.116 on ordinary LDS, versus 8.929 and 0.104 on shuffled LDS. Random training has lower conditional KL at every tested lag on shuffled LDS. On ordinary LDS, AR is better at $k = 1$; the $k > 1$ pairs were not presented consecutively during AR training. On shuffled LDS, a separate held-out-input diagnostic gives similar conditional KL on the nine forward pairs exposed by AR training (0.061 for AR versus 0.064 for random), but a large gap on the 111 unexposed pairs (3.398 versus 0.054), averaging equally within each group. This ablation retains parameter sharing across lags, so it tests the single-token restriction rather than whether the theoretical lower bounds are attained.

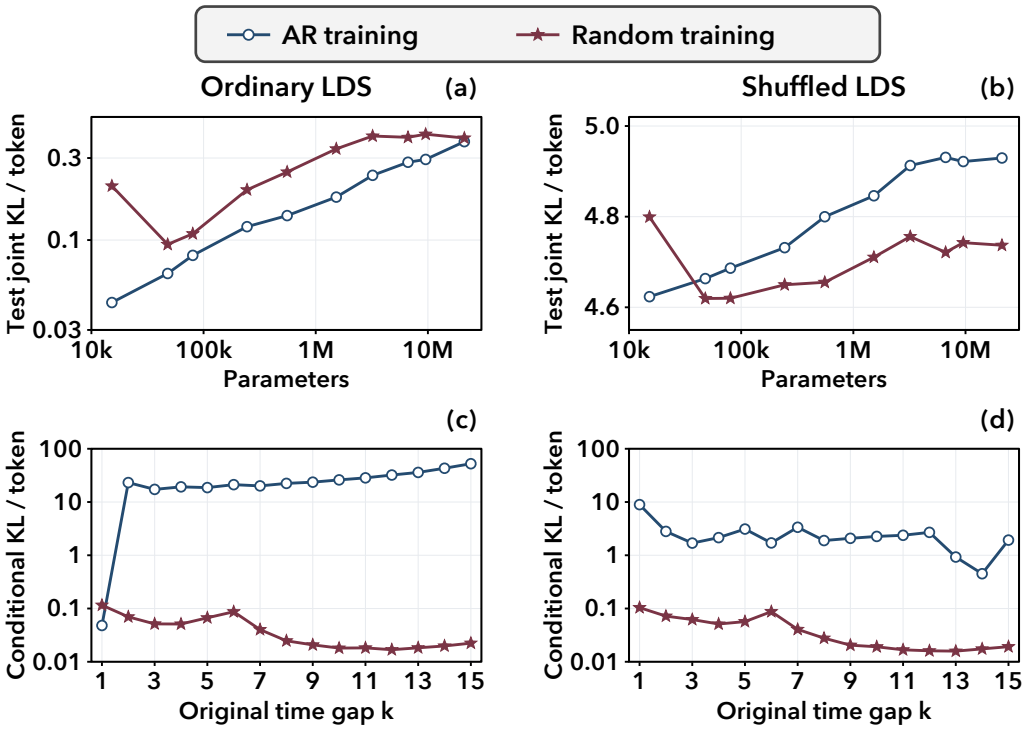

Figure 10: LDS with single-token conditioning. **(a,b)** Teacher-forced estimates of test joint forward KL per token under common AR evaluation, comparing matched capacities. **(c,d)** Direct lag-conditional forward KL for independently validation-selected models: 15.2k parameters for AR and 80.2k for random training on both datasets.