The Flexibility Trap: Rethinking the Value of Arbitrary Order in Diffusion Language Models

Zanlin Ni[1], Shenzhi Wang[1], Yang Yue[1], Tianyu Yu[2], Weilin Zhao[2], Yeguo Hua[3], Tianyi Chen[3], Jun Song[4], Cheng Yu[4], Bo Zheng[4], Gao Huang

[1]LeapLab, Tsinghua University [2]NLPLab, Tsinghua University
[3]Tsinghua University [4]Alibaba Group [5]BNRist, Tsinghua University

ICML 2026 Outstanding Paper Award

Preliminaries

- Diffusion Large Language Models
  - Diffusion Large Language Models (dLLMs), particularly Masked Diffusion Models (MDMs), generate sequences by iteratively denoising a masked state x_t initialized from fully masked tokens. The process is indexed by a continuous time variable $t \in [0, 1]$, representing the masking ratio. Given a clean sequence x_0, the forward process independently masks each token with probability t:

$$q(x_{t,k} \mid x_{0,k}) = \begin{cases} \texttt{[MASK]}, & \text{with prob } t, \\ x_{0,k}, & \text{with prob } 1 - t, \end{cases}$$

    where k is the token index. A neural network $p\theta(x_0 \mid x_t)$ estimates the original token distribution at masked positions. During inference, generation starts from x_1 (all [MASK]) and iteratively unmasks a subset of tokens based on heuristics such as confidence scores, updating $x_t \rightarrow x_t - \Delta t$ until completion.

[1] Jinjie Ni et al., Diffusion Language Models are Super Data Learners, arXiv'25
[2] Keya Hu et al., ELF: Embedded Language Flows, arXiv'26

Preliminaries

- Diffusion Large Language Models
  - As a special case, dLLMs can also perform generation autoregressively by always unmasking the leftmost token. The model is trained by minimizing the Negative Evidence Lower Bound, which reduces to a weighted cross entropy loss over masked tokens:

$$\mathcal{L}_{\text{MDM}}(\theta) = -\mathbb{E}_{t\sim\mathcal{U}[0,1],\, x_t\sim q(x_t|x_0)} \left[\frac{1}{t} \sum_{k=1}^{L} \mathbf{1}[x_{t,k} = \texttt{[MASK]}] \log p_\theta(x_{0,k} \mid x_t) \right]$$

Preliminaries

- Group Relative Policy Optimization
  - Group Relative Policy Optimization (GRPO) is a reinforcement learning algorithm that avoids value function estimation by using group level reward normalization. It is typically applied to autoregressive policies $\pi_\theta(o_k \mid o<k, q)$. For each query q, GRPO samples a group of G outputs $\{o_1, \ldots, o_G\}$ from the old policy π_θ. An advantage $\hat{A}_{i,k}$ is computed by standardizing the reward $r(o_i)$ against the group old statistics: $\hat{A}_{i,k} = (r(o_i) - \mu_G)/\sigma_G$, where μ_G and σ_G are the group mean and standard deviation. The GRPO objective maximizes a clipped surrogate function with a KL regularization term:

$$\mathcal{J}(\theta) = \mathbb{E}_{q\sim P(Q),\{o_i\}_{i=1}^{G}\sim\pi_{\theta_{\text{old}}}} \left[\frac{1}{G}\sum_{i=1}^{G} \frac{1}{|o_i|}\sum_{k=1}^{|o_i|} \left(\min\left(\rho_{i,k}\hat{A}_{i,k}, \text{clip}(\rho_{i,k}, 1-\epsilon, 1+\epsilon)\hat{A}_{i,k}\right) - \beta\mathbb{D}_{\text{KL}} \right) \right]$$

  with the token level importance ratio $\rho_{i,k} = \pi_\theta(o_{i,k}|o_{i,<k},q)$.

Preliminaries

- Pass@k as a Proxy for Reasoning Potential
  - In the context of Reinforcement Learning with Verifiable Rewards (RLVR), which is currently the dominant paradigm for enhancing reasoning capabilities, exploration is a prerequisite for improvement. **An RL agent can only reinforce correct reasoning paths if it is capable of sampling them during the exploration phase to obtain a positive reward signal.**
  - It measures the probability that at least one correct solution is generated within k independent samples, effectively representing the upper bound of the solution space accessible to RL optimization. A high Pass@k indicates that the correct reasoning trajectory lies within the model's sampling distribution and is thus learnable via RL optimization.

$$\text{Pass@}k = \mathbb{E}\left[1 - \frac{\binom{n-c}{k}}{\binom{n}{k}}\right]$$

Motivation

- Diffusion Large Language Models (dLLMs) challenges the dominant autoregressive (AR) paradigm by **treating sequence generation as a discrete denoising process**. Central to the appeal of dLLMs is their theoretical flexibility, which offers two distinct advantages over the strict left-to-right causal chain of AR models: **efficient parallel decoding and the capability for arbitrary-order generation**.
- While the efficiency gains of parallel decoding have been well-established, the implications of **arbitrary-order generation** remain relatively less explored. Theoretically, the unconstrained generation order constitutes a superset of the fixed autoregressive trajectory and has the potential to unlock more advanced problem-solving abilities.

Motivation

- In this work, we present a **counter-intuitive observation**: for these general reasoning tasks, arbitrary-order generation may in fact narrow rather than expand the reasoning potential elicitable by RL.
- To rigorously assess this, we employ Pass@k, which measures the coverage of solution space. Under this metric, we compare the reasoning potential of arbitrary-order generation against standard AR decoding using LLaDA-Instruct.

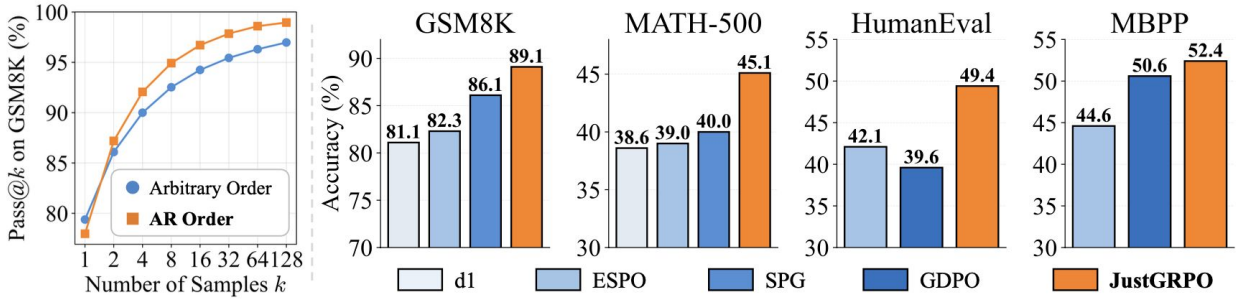


Figure 1: **Less flexibility unlocks better reasoning potential.** *Left*: We observe a counter-intuitive

Motivation

- The counter-intuitive observation is closely related to how different orders handle uncertainty. **Reasoning is inherently non-uniform**: it hinges on sparse “forking tokens”, typically connectives like “Therefore” or “Since”, which do not merely continue a sentence but steer the logical trajectory into distinct branches.

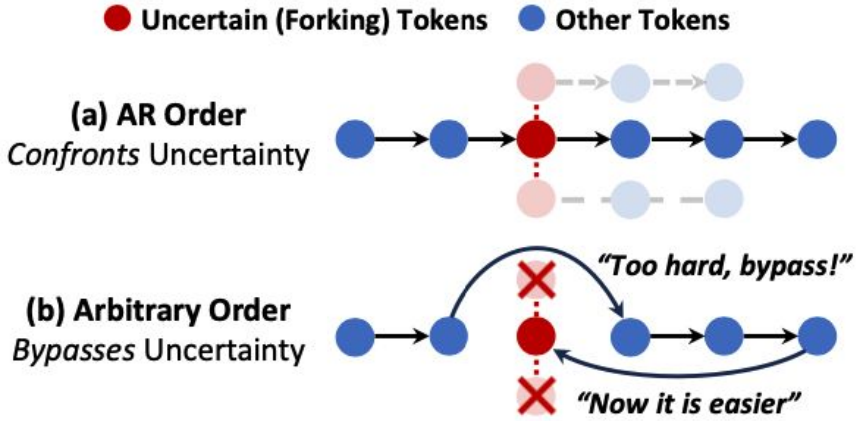
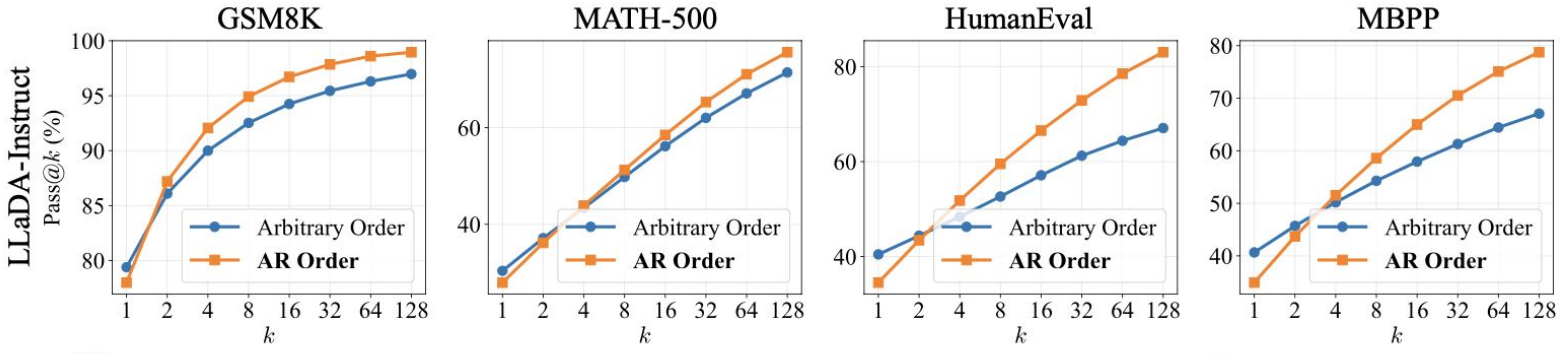


Figure 2: **Confronting *vs.* bypassing uncertainty.** (a) **AR order** preserves reasoning space by forcing decisions at uncertain tokens. (b) **Arbitrary order** bypasses uncertainty and resolves easier tokens first. Once future context is established, the original branching possibilities are effectively pruned.

Method - Arbitrary order can limit reasoning potential

- Pass@k analysis.
  - We first evaluate the reasoning potential using the Pass@k metric on three representative dLLMs: LLaDA-Instruct, Dream-Instruct, and LLaDA 1.5 on four reasoning benchmarks: GSM8K, MATH-500, HumanEval, and MBPP. As shown in Figure 3, while arbitrary order often achieves competitive and in many cases better performance at k = 1, it exhibits a notably flatter scaling curve compared to the AR mode. As k increases, the AR mode demonstrates a stronger capability to uncover more correct solutions.



Method - Arbitrary order can limit reasoning potential

- Solution space coverage.
  - One might wonder whether arbitrary order explores a different solution space, albeit less efficiently, which could account for its lower reasoning potential. We test this by analyzing solution coverage at k = 1024 using LLaDA-Instruct. As shown in Figure 4, we find that solutions obtained via arbitrary-order generation largely overlap with those discovered by AR decoding, and in practice form a smaller subset.

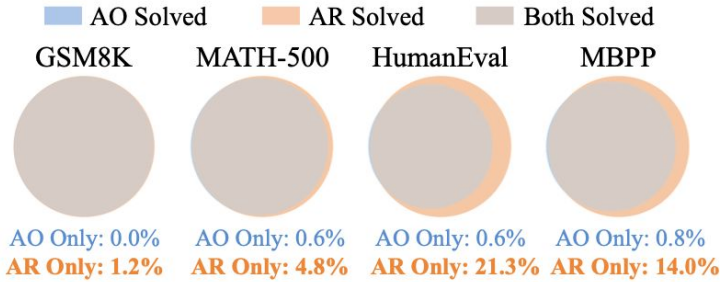


Figure 4: **Solution space coverage** measured by Pass@1024. The problems solvable by arbitrary order (AO) are largely a subset of those solved by AR Order. Results measured with LLaDA-Instruct.

Method

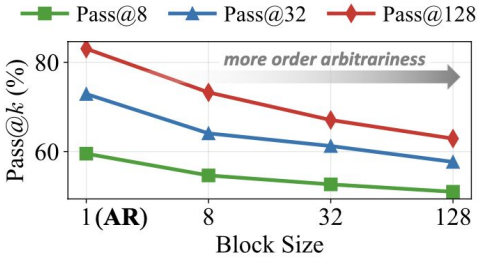


Figure 5: **More arbitrariness, less potential.**

- More arbitrariness, less potential.
  - So far we have contrasted arbitrary order with AR order as two extremes, but the degree of arbitrariness is not binary. The semi-autoregressive protocol in dLLMs controls it through the block size B: the sequence is split into blocks of size B, and within each block the model adaptively chooses tokens to unmask. Thus B sets the range within which the model can freely choose the next position, with B = 1 leaving no freedom (pure AR order) and larger B allowing more arbitrary decoding; our experiments above use B = 32. We now sweep B to examine how the degree of arbitrariness affects reasoning potential.
  - As shown in Figure 5, the trend is consistent and monotonic: across every k $\in$ {8,32,128}, Pass@k decreases as B grows. This suggests the effect is consistent rather than tied to the two extreme settings: in this setting, less arbitrariness is consistently associated with higher reasoning potential.

Method - Mechanism: the entropy degradation

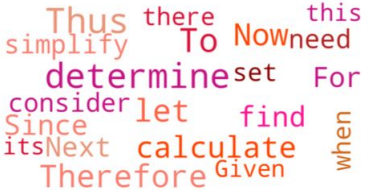
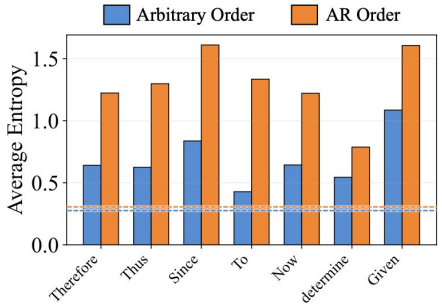


Figure 6: **Frequently bypassed tokens** in arbitrary or-

- Adaptive decoding bypasses logical forks.
  - To understand why the theoretically larger solution space of dLLMs appears to collapse in practice, we examine more closely how the two decoding modes behave. In AR order, the model is constrained to strictly resolve the left-most unknown token at each step, **forcing the model to confront uncertainty as it arises**. In contrast, arbitrary order adaptively selects tokens to update based on model confidence, preferentially generating “easy” tokens with high certainty while bypassing “hard” ones. Inspecting the frequently bypassed tokens reveals a clear pattern: As shown in the figure, the diffusion sampler disproportionately defers logical connectives and transition markers such as “Therefore”, “Thus”, and “Since”. Prior work has shown that **such tokens often have high entropy**, and act as “reasoning sparks” or “logical forks”, functioning as branching points that determine subsequent reasoning directions. In conventional language models, keeping these tokens in high-entropy state is critical for effective exploration of the reasoning space.

Method - Mechanism: the entropy degradation



- The “entropy degradation” phenomenon.
  - How does the adaptive behavior of arbitrary order affect these logical forking tokens? We measure the entropy of them when they are decoded in the figure. In AR order, these tokens generally maintain high entropy when decoded, indicating a relatively fruitful branching point where multiple reasoning paths remain viable. In contrast, arbitrary order exhibits a sharp decrease in entropy. By deferring the logical connectors, the model prioritizes generating easier future tokens before deciding the logic connections that lead to them. When the model eventually returns to fill in the bypassed connectors, the presence of future context significantly reduces uncertainty. As a result, the model is no longer making an open-ended navigational decision at a fork, but more like a retrospective alignment to bridge the gap to a pre-determined conclusion. We term this phenomenon entropy degradation.

Method - “Just GRPO” for dLLMs

- The Flexibility Tax in dLLMs’ RL
  - **Ambiguity in token-level decomposition.** In dLLMs, a generation state s_t is a noisy sequence conditioned on a stochastic unmasking trajectory τ. Unlike AR models, dLLMs do not admit a unique, index-aligned conditional probability of the form $\pi(o_t|s_t)$, making token-level credit assignment ambiguous and rendering the standard importance ratio $\rho_t=\pi_\theta(o_t|s_t)/\pi_{old}(o_t|s_t)$ hard to define.
  - **Intractable sequence likelihood.**
  - **Sampler-learner mismatch.** Even with an accurate likelihood approximation, a more subtle issue persists. In practice, rollout samples are typically produced by confidence-based generation $o \sim \pi_\theta^{conf}(o|q)$ to navigate the combinatorial space. However, the ELBO (Evidence Lower Bound) objective still targets the likelihood of the original model distribution $\pi_\theta(o \mid q)$, rather than that of the actual sampling policy $\pi_\theta^{conf}(o|q)$, leading to a mismatch between rollout and optimization that can degrade performance.

Method - JustGRPO

- Formulation.
  - Standard GRPO assumes a policy $\pi(o_k|o_{<k},q)$ accepting a partial sequence o<k with all tokens observed and predicts one token o_k at a time, where q is the query. Diffusion language models, however, are architected as sequence-level denoisers that accept a full sequence with mixed observed and masked tokens and predict the original values for all masked positions simultaneously.
  - By forgoing arbitrary-order generation, we are able to bridge the above gap and rigorously define an AR policy π_θ^{AR} for dLLMs. To obtain the probability of the next token o_k given history $o_{<k}$, we construct $\tilde{x}_k$ where the past is observed and the future is masked:

$$\tilde{x}_k = [\underbrace{o_1, \ldots, o_{k-1}}_{\text{Observed}}, \underbrace{\texttt{[MASK]}, \ldots, \texttt{[MASK]}}_{\text{Masked}}].$$

Method - JustGRPO

- Formulation.
  - Although the diffusion language model outputs predictions for all masked positions, the autoregressive policy concerns only the next token o_k. We define $\pi_\theta^{AR}(\cdot|o_{<k},q)$ as follows:

$$\pi_\theta^{\mathrm{AR}}(\cdot|o_{<k},q) \triangleq \mathrm{Softmax}(f_{\theta,k}(\tilde{x}_k,q))$$

  - where $f_{\theta,k}$ denotes the model logits at position k. By defining a surrogate autoregressive policy π_θ^{AR} on top of the dLLM backbone, we convert the intractable marginalization over permutations into an exactly computable likelihood:

$$\pi_\theta^{\mathrm{AR}}(o|q) = \prod_{k=1}^{|o|} \pi_\theta^{\mathrm{AR}}(o_k|o_{<k},q).$$

Method - JustGRPO

- Optimization.
  - The above formulation enables the direct application of standard GRPO to diffusion language models. For each query q, we sample a group of outputs $\{o_i\}_G^{i=1}$ using the old policy π_{AR}. The objective is:

$$\mathcal{J}(\theta) = \mathbb{E}_{q\sim P(Q),\{o_i\}_{i=1}^{G}\sim\pi_{\theta_{\text{old}}}^{\text{AR}}}\left[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_i|}\sum_{k=1}^{|o_i|}\left(\min\left(\rho_{i,k}\hat{A}_{i,k}, \text{clip}(\rho_{i,k}, 1-\varepsilon, 1+\varepsilon)\hat{A}_{i,k}\right) - \beta\mathbb{D}_{\text{KL}}\right)\right]$$

- Remarks.
  - One might worry whether training in AR mode fundamentally turns the diffusion model into a standard autoregressive model. This is not the case. The AR constraint is applied exclusively during RL training for more effective exploration and credit assignment. Crucially, this refines the model distribution without imposing structural constraints, e.g., causal masking, that could alter the underlying structure or compromise the parallel sampling native to dLLMs. Empirically, the model remains compatible with parallel samplers to accelerate decoding at inference. **JustGRPO thus benefits from AR-based reasoning exploration while preserving the parallel capability of dLLMs**.

Experiments

- Overall
  - We evaluate JustGRPO on standard **mathematical reasoning and coding benchmarks**. Our experimental design aims to examine two hypotheses: (i) that enforcing autoregressive (AR) order during RL training can effectively elicit reasoning capabilities without resorting to complex arbitrary-order approximations, and (ii) that this constraint applies only to the optimization objective, leaving the diffusion model's parallel decoding capabilities intact at inference.
  - We apply JustGRPO directly to the models without additional task-specific SFT, using full-parameter fine-tuning.

Experiments

- Performance on reasoning tasks.

Table 2: **Reproduction with configurations aligned to ours** (full fine-tuning, one token per decoding step, generation length 256). Reproduced baselines are marked with *. ESPO* results on HumanEval and MBPP are taken directly from the original paper, whose setting already matches ours (full fine-tuning, one token per decoding step).

Model	GSM8K	MATH-500	HumanEval	MBPP
d1*	83.8	39.2	–	–
ESPO*	84.7	40.3	42.1	44.6
SPG*	86.9	41.8	–	–
JustGRPO	**89.1**	**45.1**	**49.4**	**52.4**

Experiments

- JustGRPO preserves parallel decoding

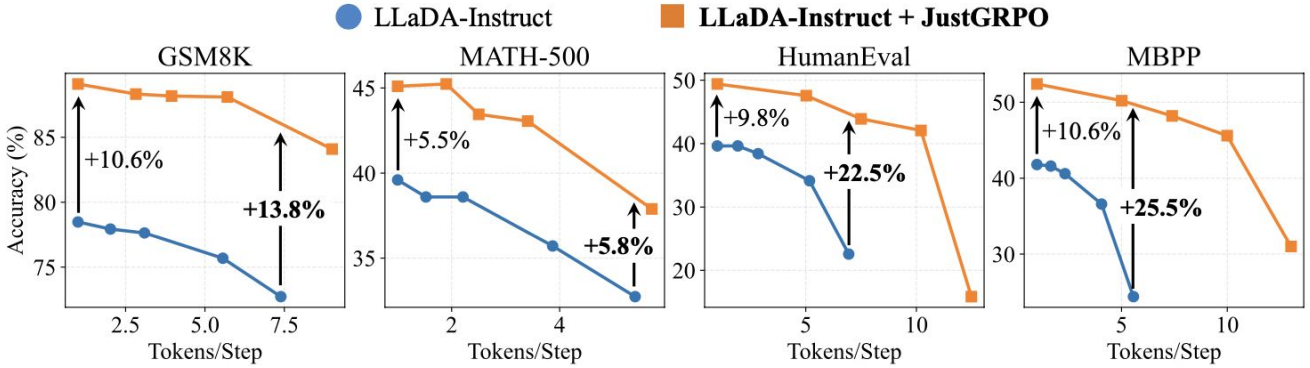


Figure 8: **JustGRPO preserves the parallel decoding capability of dLLMs.** Interestingly, when compared to original instruct model, accuracy gains are larger with more parallel tokens, likely due to more robust reasoning capabilities after JustGRPO training. We adopt training-free EB-sampler (Ben-Hamu et al., 2025) for parallel decoding. Generation length is set to 256.

Experiments

- Training efficiency

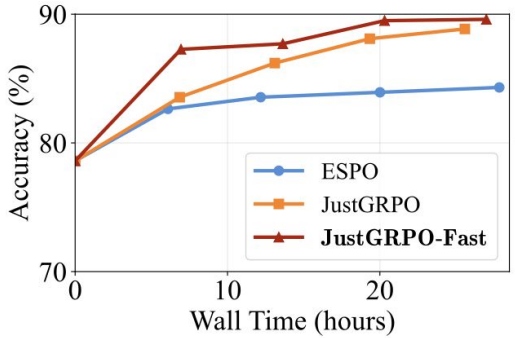


Figure 9: **Training efficiency on GSM8K.** Despite its larger per-iteration overhead, JustGRPO achieves a better accuracy/wall-time trade-off than the approximation-based baseline (ESPO), which saturates early. JustGRPO-Fast, which computes the probability ratio $\rho_{i,k}$ only at the top-25% highest-entropy positions, accelerates training further. Wall time is measured on 16×H100 GPUs.

Takeways

- LLMs vs dLLMs?
- Reasoning in 3D tasks?