

JustOPD: On-Policy Distillation from Autoregressive to Diffusion Language Models

Zikun Zhang* Jiayuan Sheng* David D. Yao* Wenpin Tang*

Abstract

On-policy distillation (OPD) is a post-training approach for large language models (LLMs) that enables a student model to learn a teacher model’s distribution from sequences sampled from the student’s own policy. Despite its promising capability, OPD for diffusion language models (DLMs) remains underexplored and existing studies primarily focuses on DLM-to-DLM distillation. The limited availability of strong teacher DLMs and the flexible generation order of DLMs impose additional technical challenges as well. However, it remains unclear how to effectively distill knowledge from a strong autoregressive language model (ARM) into a weaker DLM, given the mismatch between the left-to-right decoding of ARMs and the flexible decoding order of DLMs. Inspired by [Ni et al. \(2026\)](#), we propose JustOPD, an ARM-to-DLM on-policy distillation method that uses autoregressive rollouts during training to align the student’s generation process with the ARM teacher’s left-to-right decoding. This alignment enables effective distillation from ARM teachers while preserving the DLM’s flexible generation-order capability at inference time. Experiments using Qwen3-14B as the teacher and Dream-7B as the student demonstrate that JustOPD consistently improves mathematical reasoning performance over the base DLM across multiple divergence choices.

1 Introduction

On-policy distillation (OPD) ([Agarwal et al., 2024](#); [Lu and Lab, 2025](#)) has recently emerged as a key component of LLM post-training, where the student learns directly from the teacher’s distribution on responses sampled from the student’s own policy, while the teacher provides dense supervision signals for these student-generated responses. For autoregressive language models (ARMs), OPD is relatively straightforward, as the student and teacher can be chosen from the same model family (e.g., the Qwen3 series), leveraging the many strong and readily available open-source ARMs. Moreover, the fixed left-to-right, token-by-token generation process of ARMs naturally aligns the generation trajectories of the student and teacher, making ARM-to-ARM OPD fundamentally compatible.

In contrast, applying OPD to mask-based diffusion language models (DLMs) introduces additional challenges. In particular, the arbitrary-order generation process of DLMs requires careful consideration of the student’s generation trajectory, while strong and readily available DLM teachers remain scarce. [Ren et al. \(2026\)](#) first studied DLM-to-DLM distillation by querying a stronger teacher DLM with the partially denoised sequences generated by the student, which naturally accommodates the arbitrary generation order of DLMs. However, explicitly recording and replaying the generation trajectory of each student sample can be computationally expensive, particularly for long sequences. Furthermore, restricting the teacher to be a DLM limits the scalability and

*Department of Industrial Engineering and Operations Research, Columbia University, New York, NY 10027. Emails: zz3367@columbia.edu, js6646@columbia.edu, yao@columbia.edu, wt2319@columbia.edu.

generality of this approach, given the scarcity of strong, open-source DLMs. Overall, the flexibility of arbitrary-order generation introduces substantial complexity to DLM post-training, particularly in aligning the student’s generation trajectory with that of the teacher.

Given the limited availability of strong DLM teachers, an appealing alternative is to distill knowledge from a well-trained ARM into a weaker DLM. This direction is particularly promising because ARMs are substantially more mature and widely adopted in the LLM community, with a large number of high-performing models available for distillation. Moreover, ARM-to-DLM distillation offers an attractive way to combine the complementary strengths of the two paradigms: ARMs generally achieve stronger generation quality, while DLMs enable parallel token generation and can substantially reduce generation latency compared with ARMs under comparable hardware and engineering optimization. However, transferring knowledge from an ARM to a DLM is fundamentally challenging due to their incompatible generation paradigms. An ARM is trained with a causal mask and generates tokens strictly from left to right, conditioning each token on a complete prefix, while a DLM progressively denoises a sequence and can predict tokens in an arbitrary order. As a result, the partially denoised sequences at intermediate denoising steps encountered by a DLM do not generally correspond to valid prefixes under the ARM’s training distribution, making it unclear how an ARM teacher can provide meaningful supervision throughout the DLM’s denoising process. This mismatch motivates the development of effective ARM-to-DLM distillation methods.

To bridge this gap, we propose to enforce left-to-right rollout generation for the DLM during training, so that the intermediate student states remain valid prefixes that can be directly processed by an ARM teacher. This idea is inspired by JustGRPO (Ni et al., 2026), which observes that arbitrary-order generation during reinforcement learning (RL) rollouts may hinder the reasoning capability of DLMs: the model may generate easier future tokens with high confidence before resolving the more difficult tokens that depend on them. JustGRPO therefore adopts autoregressive-style rollouts during RL training and demonstrates substantial performance gains. Motivated by these observations, we similarly leverage autoregressive rollouts during OPD, which we term **JustOPD**, to bridge the generation-paradigm gap between ARMs and DLMs, while preserving the flexible arbitrary-order generation capability of DLMs at inference time. We conduct experiments by choosing Dream-7B (Ye et al., 2025) as the base student DLM and Qwen3-14B (Yang et al., 2025) as the teacher ARM, because Dream-7B uses a full-attention diffusion model but initialized from a pretrained ARM checkpoint (Qwen2.5-7B weights). Our experiment results on several mathematical reasoning benchmarks (GSM8K and MATH500) demonstrate the effectiveness of JustOPD.

1.1 Contributions

Our contributions are summarized as follows:

1. We propose JustOPD, an effective ARM-to-DLM on-policy distillation method that transfers knowledge from an autoregressive teacher to a diffusion student. JustOPD enforces autoregressive-style rollouts for the DLM student during training, aligning its intermediate generation trajectory with the ARM teacher and thereby enabling the teacher to provide meaningful token-level supervision. We instantiate JustOPD with a sampled-token, PPO-style objective that supports general f -divergences, and we evaluate the loss only at the 25% of positions where the teacher’s entropy is highest, removing 75% of the student forward passes.
2. We empirically validate JustOPD using Dream-7B as the student and Qwen3-14B as the teacher on GSM8K and MATH500. After noting that strong ARM teachers suffer a sharp drop in performance under the short generation lengths typical of DLMs, we fine-tune Qwen3-14B to

generate concise responses while preserving accuracy, making it an effective teacher. JustOPD consistently improves student performance across all tested settings, including different α -divergences and the Jensen–Shannon divergence. We evaluate these gains using the DLM’s native any-order decoder, demonstrating that knowledge distilled through autoregressive rollouts transfers effectively to arbitrary-order inference.

1.2 Related Works

On-Policy Distillation. OPD was originally proposed to fine-tune ARMs by pushing the token-level distribution of the student model to that of the teacher model. Agarwal et al. (2024) introduced Generalized Knowledge Distillation, which formulates on-policy knowledge distillation as a supervised learning problem by training the student on sequences sampled from its own policy while using the teacher to provide token-level supervision. Lu and Lab (2025) further studied an RL-style formulation of OPD based on KL divergence, establishing a connection between on-policy distillation and policy optimization and showing that OPD can serve as a general framework for improving a student model using dense supervision from a stronger teacher. Jin et al. (2026) showed that the mode-seeking reverse KL collapses generation diversity and yields unstable learning signals at tokens where the teacher distribution has high entropy, so they propose Entropy-Aware On-Policy Distillation, which adds a forward KL term and computed over the teacher’s top-k tokens, only at such high-entropy positions. Zhao et al. (2026b) proposed On-Policy Self-Distillation (OPSD), in which a single ARM acts as both teacher and student: the teacher is conditioned on privileged information, the student sees only the question, and training minimizes the per-token divergence between the two along the student’s own rollouts. Recent studies investigate the mechanisms and data requirements of OPD (Li et al., 2026; Ding and Zhang, 2026; Yu et al., 2026). Complementary work shows that even a single training query can recover most of the gains of full-data OPD (Fu et al., 2026b). Song and Zheng (2026) reviews recent progress of OPD methods for ARMs.

OPD for DLMs. OPD for DLMs is limited in the existing literature. TOPD (Ren et al., 2026) distills a stronger DLM teacher on the committed decisions along the DLM student’s own denoising trace. HOPD (Huang et al., 2026) extends TOPD by matching teacher distributions at every masked position in each denoising step, while the decoder commits only to a confident subset of the DLM’s proposed tokens. Recent works also extend OPSD to DLMs by sourcing the teacher’s privileged information from the model’s own generation. d-OPSD (Luo et al., 2026) conditions the self-teacher on partially revealed tokens of the student’s final answer as a suffix and supervises at the denoising-step level. dOPSD (Dat et al., 2026) uses later, more-decoded states of the student’s own denoising trajectory as a peek-ahead teacher for an earlier intermediate state. Both report gains over SFT and RL-based post-training of DLMs. In addition, OPDLM (Su et al., 2026) instead uses OPD to convert an ARM into a DLM under self-distillation, supervising the student’s own denoising states with the frozen ARM teacher queried on causal prefixes of the student’s final output. ForkLeft (Liu et al., 2026) also resolves the ARM-to-DLM mismatch by comparing the causal teacher and the bidirectional student only on student-generated prefixes, but it generates training rollouts in an entropy-first order and applies supervision at uniformly sampled positions. JustOPD instead uses strictly left-to-right rollouts, so the student and teacher share the same autoregressive factorization and the standard OPD objective applies directly, and it uses teacher entropy not to order the rollout but to select which positions receive supervision.

Post-Training of DLMs. Due to fundamental modeling differences between the auto-regressive and diffusion generative paradigms for large language models, DLMs enable arbitrary-order decoding at the cost of intractable or computationally prohibitive trajectory likelihood, driving a line of work

(Zhao et al., 2025a,b; Yang et al., 2026a; Wang et al., 2026b,c; Ou et al., 2026; Ni et al., 2026; Zhang et al., 2026) focusing on improving the reasoning capabilities of DLMs using RL fine-tuning in recent years. In particular, Ni et al. (2026) indicated that arbitrary-order decoding can narrow rather than expand the reasoning potential of DLMs: confidence-based samplers defer high-entropy “forking” tokens until future context has resolved their uncertainty, which lowers Pass@ k relative to left-to-right decoding. They therefore propose JustGRPO, which treats the DLM as an autoregressive policy during RL training so that standard GRPO applies with exact likelihoods while keeping parallel decoding at inference. OPD is closely tied to RL: with the teacher–student log-probability gap as a dense per-token reward, it reduces to a policy-gradient objective that can be optimized in standard RL pipelines (Gu et al., 2024; Lu and Lab, 2025; Zhao et al., 2026b; Jin et al., 2026), and G-OPD (Yang et al., 2026b) further generalizes it into a KL-regularized RL objective whose reward can be extrapolated to surpass the teacher. Our work is closely related to the autoregressive rollout strategy introduced by Ni et al. (2026) and the entropy-aware generation strategy proposed by Jin et al. (2026), both of which address the challenges arising from arbitrary-order generation in diffusion language models during post-training.

2 Preliminaries

Notations. We use lowercase letters to denote scalars, and boldface lowercase letters to denote vectors. The i -th entry of a vector $\mathbf{x}$ is denoted by $\mathbf{x}^i$, or simply x^i when the context is clear, and $\mathbf{x}^{<i}$ denotes $(x^1, \dots, x^{i-1})$. $\mathbf{e}_j$ refers to a one-hot column vector with a 1 in the j -th position. We use $\text{Cat}(\cdot; \mathbf{p})$ for a categorical distribution over the vocabulary with probabilities given by $\mathbf{p}$.

Diffusion Language Models. We consider masked diffusion models (Shi et al., 2024; Sahoo et al., 2024; Lou et al., 2024; Campbell et al., 2022). Let $\mathcal{S} = (\mathcal{V} \cup \{\mathbf{m}\})^L$ be the state space, where $\mathcal{V}$ is a discrete vocabulary space, $\mathbf{m}$ is the special “mask” token, and L is the sequence length.

The forward process of a DLM gradually corrupts the clean data distribution p_{data} over $\mathcal{S}$ into pure mask noise according to a continuous-time transition kernel. At noise level $\tau \in [0, 1]$, $\mathbf{x}_\tau$ is a partially-noised sequence, which is obtained by independently masking each dimension of $\mathbf{x}_0 \sim p_{\text{data}}$ with probability $1 - \alpha_\tau$. Here, α_τ is the noise scheduled to satisfy $\alpha_0 \approx 1$ and $\alpha_1 \approx 0$ and a common choice is $\alpha_\tau = 1 - \tau$ (Ye et al., 2025; Nie et al., 2026).

The reverse process gradually recovers clean tokens starting from the masks. Specifically, a denoiser $\mathbf{p}_\theta(\cdot)$ is trained using a cross-entropy loss (Shi et al., 2024; Sahoo et al., 2024; Ye et al., 2025; Nie et al., 2026) to predict the distributions of all masked tokens simultaneously. To be more specific, given the current partially denoised sequence $\mathbf{x}_\tau$, for any $i \in [L]$ with $x_\tau^i = \mathbf{m}$, $\mathbf{p}_\theta^{(i)}(\mathbf{x}_\tau)$, the i -th output of $\mathbf{p}_\theta(\mathbf{x}_\tau)$, predicts the conditional distribution $p_{0|\tau}(x_0^i = \cdot \mid \mathbf{x}_\tau)$. At inference time, the continuous-time reverse process is discretized over a set of time steps indexed by $t \in \mathbb{Z}_{\geq 0}$, and various strategies can be used to decide which masked tokens to unmask at each denoising step, such as selecting the k tokens with the highest confidence (Chang et al., 2022), the largest margin between the top two probabilities (Kim et al., 2025), or the lowest predictive entropy (Ye et al., 2025; Ben-Hamu et al., 2026). More details are deferred to Appendix A.

On-Policy Distillation for ARMs. In ARMs, given a prompt $\mathbf{q}$, the model is parametrized to predict the next-token distribution $p_\theta(\cdot \mid \mathbf{q}, \mathbf{x}^{<t})$ conditioned on the prefix $(\mathbf{q}, \mathbf{x}^{<t})$, so the probability of a response $\mathbf{x}$ factorizes as $p_\theta(\mathbf{x} \mid \mathbf{q}) = \prod_{t=1}^{|\mathbf{x}|} p_\theta(x^t \mid \mathbf{q}, \mathbf{x}^{<t})$. Let $f \in C^1(0, +\infty)$ be a convex function with $f(1) = 0$, and define the f -divergence $D_f(\mathbf{x} \parallel \mathbf{y}) := \sum_i y^i f(x^i/y^i)$. Given a teacher ARM p_{te} and a student ARM p_θ , OPD minimizes the token-level f -divergence between the teacher

and the student along the student’s own trajectories:

$$\min_{\theta} \mathbb{E}_{\mathbf{q} \sim \mathcal{D}, \mathbf{x} \sim p_{\theta}(\cdot | \mathbf{q})} \frac{1}{|\mathbf{x}|} \sum_{t=1}^{|\mathbf{x}|} D_f(p_{\mathbf{te}}(\cdot | \mathbf{q}, \mathbf{x}^{<t}) \| p_{\theta}(\cdot | \mathbf{q}, \mathbf{x}^{<t})). \quad (1)$$

By evaluating the teacher and student distributions on prefixes generated by the student itself, OPD directly trains the student on the states it is likely to encounter during inference, enabling it to iteratively improve through the teacher’s guidance on its own outputs. Typical choices of f include $f(u) = -\log u$ (reverse KL), $f(u) = u \log u$ (forward KL), $f(u) = \frac{u^{\alpha} - \alpha u - (1 - \alpha)}{\alpha(\alpha - 1)}$ with $\alpha \in (0, 1)$ (α -divergence, which recovers centralized reverse KL as $\alpha \rightarrow 0^+$ and centralized forward KL as $\alpha \rightarrow 1^-$, and $f(u) = \frac{1}{2}(u \log u - (1 + u) \log \frac{1+u}{2})$ (Jensen-Shannon divergence).

3 JustOPD

Motivation. Our goal is to distill the knowledge from a stronger ARM to a weaker DLM. [Ni et al. \(2026\)](#) argue that arbitrary-order generation during RL rollouts may limit the reasoning capability of DLMs, as it may generate easier future tokens with high confidence before determining the more difficult logical dependencies leading to them, which are inherently high-entropy. Hence, during training, we consider a token-level autoregressive policy for the student DLM, where a sequence $\mathbf{x}$ is generated via left-to-right decoding. Specifically, given a prompt $\mathbf{q}$, for $t = 1, 2, \dots$, the student policy follows

$$\pi_{\theta}(\cdot | \mathbf{q}, \mathbf{x}^{<t}) := \text{Cat}(\cdot; \mathbf{p}_{\theta}^{(t)}(\mathbf{q}, \mathbf{x}^{<t}, \mathbf{m}, \dots, \mathbf{m})), \quad (2)$$

where the denoiser $\mathbf{p}_{\theta}^{(t)}(\cdot)$ takes as input the prefix $(\mathbf{q}, \mathbf{x}^{<t})$ concatenated with a masked future suffix and predicts the clean token at the t -th position. The resulting sequence-level policy has the token-level factorization:

$$\pi_{\theta}(\mathbf{x} | \mathbf{q}) := \prod_{t=1}^{|\mathbf{x}|} \pi_{\theta}(x^t | \mathbf{q}, \mathbf{x}^{<t}).$$

Let $\pi_{\mathbf{te}}$ be an ARM teacher model with the natural trajectory likelihood factorization:

$$\pi_{\mathbf{te}}(\mathbf{x} | \mathbf{q}) := \prod_{t=1}^{|\mathbf{x}|} \pi_{\mathbf{te}}(x^t | \mathbf{q}, \mathbf{x}^{<t}).$$

As a result, the generation order of the ARM teacher and DLM student coincides, which enables us to use the OPD objective (1) developed for ARMs by simply replacing p_{θ} by π_{θ} and $p_{\mathbf{te}}$ by $\pi_{\mathbf{te}}$.

Full/TopK-Vocabulary Distillation. In practice, for training stability, the f -divergence in (1) is clipped ([Zhao et al., 2026b](#)) with a constant $C > 0$:

$$D_f^{\text{clip}}(\pi_{\mathbf{te}}(\cdot | \mathbf{q}, \mathbf{x}^{<t}) \| \pi_{\theta}(\cdot | \mathbf{q}, \mathbf{x}^{<t})) := \min \left(D_f(\pi_{\mathbf{te}}(\cdot | \mathbf{q}, \mathbf{x}^{<t}) \| \pi_{\theta}(\cdot | \mathbf{q}, \mathbf{x}^{<t})), C \right),$$

which prevents excessively large divergence values from inducing high-variance or unstable gradient updates. Alternatively, the divergence can be computed only over the teacher’s top- K tokens ([Jin et al., 2026](#)):

$$D_f^{\text{TopK}}(\pi_{\mathbf{te}}(\cdot | \mathbf{q}, \mathbf{x}^{<t}) \| \pi_{\theta}(\cdot | \mathbf{q}, \mathbf{x}^{<t})) := \sum_{v \in \text{TopK}(\pi_{\mathbf{te}}(\cdot | \mathbf{q}, \mathbf{x}^{<t}))} \pi_{\theta}(v | \mathbf{q}, \mathbf{x}^{<t}) f \left(\frac{\pi_{\mathbf{te}}(v | \mathbf{q}, \mathbf{x}^{<t})}{\pi_{\theta}(v | \mathbf{q}, \mathbf{x}^{<t})} \right).$$

The on-policy training objective is

$$\min_{\theta} \mathbb{E}_{\mathbf{q} \sim \mathcal{D}, \mathbf{x} \sim \pi_{\theta_{\text{old}}}(\cdot | \mathbf{q})} \frac{1}{|\mathbf{x}|} \sum_{t=1}^{|\mathbf{x}|} D_f^{\dagger}(\pi_{\mathbf{te}}(\cdot | \mathbf{q}, \mathbf{x}^{<t}) \parallel \pi_{\theta}(\cdot | \mathbf{q}, \mathbf{x}^{<t})), \quad (3)$$

where $\dagger \in \{\text{clip}, \text{TopK}\}$. The pseudocode for full/top- K vocabulary distillation is presented in Algorithm 2 in Appendix B. However, computing the f -divergence at the vocabulary level remains computationally expensive, as it requires evaluating the teacher and student probabilities over a large number of vocabulary tokens at every sequence position, which is even more computationally prohibitive for DLMS. We therefore focus on a more computationally efficient sampled-token distillation approach below in this work.

Sampled-Token Distillation. In objective (1), let $u_{\theta,t} := \pi_{\mathbf{te}}(x^t | \mathbf{q}, \mathbf{x}^{<t}) / \pi_{\theta}(x^t | \mathbf{q}, \mathbf{x}^{<t})$ for a sampled student-generated sequence $\mathbf{x}$. Since $x^t \sim \pi_{\theta}(\cdot | \mathbf{q}, \mathbf{x}^{<t})$, we have $\mathbb{E}_{x^t}[f(u_{\theta,t})] = D_f(\pi_{\mathbf{te}}(\cdot | \mathbf{q}, \mathbf{x}^{<t}) \parallel \pi_{\theta}(\cdot | \mathbf{q}, \mathbf{x}^{<t}))$, so the f -divergence admits a single-sample Monte Carlo (MC) estimate (Gu et al., 2024; Lu and Lab, 2025; Zhao et al., 2026b,a; Jin et al., 2026):

$$\min_{\theta} \mathbb{E}_{\mathbf{q} \sim \mathcal{D}, \mathbf{x} \sim \pi_{\theta}(\cdot | \mathbf{q})} \frac{1}{|\mathbf{x}|} \sum_{t=1}^{|\mathbf{x}|} f(u_{\theta,t}). \quad (4)$$

The objective (4) can be optimized with policy-gradient methods in RL. In particular, the Proximal Policy Optimization (PPO)-type (Schulman et al., 2017) surrogate objective is

$$\max_{\theta} \mathbb{E}_{\mathbf{q} \sim \mathcal{D}, \mathbf{x} \sim \pi_{\theta_{\text{old}}}(\cdot | \mathbf{q})} \frac{1}{|\mathbf{x}|} \sum_{t=1}^{|\mathbf{x}|} \min(\rho_t^{\theta} \cdot A_t, \text{clip}(\rho_t^{\theta}, 1 - \epsilon, 1 + \epsilon) \cdot A_t), \quad (5)$$

where $\rho_t^{\theta} := \pi_{\theta}(x^t | \mathbf{q}, \mathbf{x}^{<t}) / \pi_{\theta_{\text{old}}}(x^t | \mathbf{q}, \mathbf{x}^{<t})$ corrects for sampling under $\pi_{\theta_{\text{old}}}$, and $A_t := -f(u_{\theta_{\text{old}},t}) + u_{\theta_{\text{old}},t} f'(u_{\theta_{\text{old}},t})$ is the per-token corrected advantage (Zhao et al., 2026a) of the sampled token x^t (also see Proposition 2).

Centralized f -Divergence. One may be concerned that the single-sample Monte Carlo estimation in (4) introduces errors compared with the full-vocabulary calculation. Here, we argue that for any f -divergence, one can choose its unique centralized version, which improves the signal-to-noise ratio of the estimation while retaining the corresponding advantage up to a constant, so the algorithm remains the same. For any convex $f \in C^1(0, +\infty)$ with $f(1) = 0$, let

$$\tilde{f}(u) := f(u) - f(1) - f'(1)(u - 1), \quad \forall u \in (0, +\infty).$$

$\tilde{f}$ is still convex and it holds that $\tilde{f}(1) = \tilde{f}'(1) = 0$, so $\tilde{f}$ is a *centralized* version of f . The following Proposition 1 shows that centralized f -divergence improves the local signal-to-noise ratio of Monte Carlo estimation, with the proof deferred to Appendix C.1.

Proposition 1. Fix a prefix $(\mathbf{q}, \mathbf{x}^{<t})$. Let $P_t(\cdot) = \pi_{\mathbf{te}}(\cdot | \mathbf{q}, \mathbf{x}^{<t})$ and $Q_{\theta,t}(\cdot) = \pi_{\theta}(\cdot | \mathbf{q}, \mathbf{x}^{<t})$ be normalized, strictly positive distributions on the same finite support. Define

$$u_{\theta,t}(x) := \frac{P_t(x)}{Q_{\theta,t}(x)} \quad \text{and} \quad D_{f,t}(\theta) := \mathbb{E}_{x' \sim Q_{\theta,t}}[f(u_{\theta,t}(x'))].$$

Assume that there exists $\gamma \in (0, 1)$ such that $|u_{\theta}(x) - 1| \leq \gamma$ for all x in the support of $Q_{\theta,t}$, and that f is four times continuously differentiable on a neighborhood of $[1 - \gamma, 1 + \gamma]$ with $f''(1) > 0$. Let $\sigma_{f,t}^2(\theta) := \text{Var}_{x' \sim Q_{\theta,t}}[f(u_{\theta,t}(x'))]$ be the variance of the single-sample estimate. Then we have the local noise-to-signal ratios $\sigma_{f,t}(\theta) / D_{f,t}(\theta) = \Theta(1/\gamma)$ and $\sigma_{\tilde{f},t}(\theta) / D_{\tilde{f},t}(\theta) = O(1)$.

Divergence	$f(u)$	$\tilde{f}(u)$	$A_f(u)$
Forward KL	$u \log u$	$u \log u - u + 1$	u
Reverse KL	$-\log u$	$-\log u + u - 1$	$\log u - 1$
α -divergence, $\alpha \in (0, 1)$	$\frac{u^\alpha - \alpha u - (1 - \alpha)}{\alpha(\alpha - 1)}$	unchanged	$\frac{u^\alpha - 1}{\alpha}$
Jensen-Shannon	$\frac{1}{2}(u \log u - (1 + u) \log \frac{1+u}{2})$	unchanged	$\frac{1}{2} \log \frac{1+u}{2}$

Table 1: Typical f -divergences and their centralized versions $\tilde{f}$ and per-token advantages A_f .

The following proposition characterizes the effect of centralizing the f -divergence on the corresponding policy gradient and corrected advantage function; the proof is provided in Appendix C.2.

Proposition 2. *Let the advantage function (Zhao et al., 2026a) be $A_f(u) := -f(u) + uf'(u)$ for all $u \in (0, +\infty)$. We have:*

(i) *The gradient of the negative divergence satisfies*

$$\nabla_\theta (-D_{f,t}(\theta)) = \mathbb{E}_{x \sim Q_{\theta,t}} [A_f(u_{\theta,t}(x)) \nabla_\theta \log Q_{\theta,t}(x)];$$

(ii) *f and $\tilde{f}$ induce the same divergence, and their corrected advantages differ by a constant:*

$$D_{\tilde{f},t}(\theta) = D_{f,t}(\theta) \quad \text{and} \quad A_{\tilde{f}}(u) = A_f(u) - f'(1), \quad \forall u \in (0, +\infty).$$

Typical centralized f -divergences along with the corrected advantages are summarized in Table 1. By Proposition 2 (ii), f and its centralized version $\tilde{f}$ share the same OPD objective and equivalent corrected advantages up to a constant baseline $f'(1)$. As a result, OPD implicitly employs a centralized f -divergence, which improves the signal-to-noise ratio of the single-sample Monte Carlo estimator in (4).

Practical Considerations. Unlike ARMs, DLMs must evaluate each position independently rather than in a single causal forward pass, so computing exact likelihoods $\{\rho_t^\theta\}_{t=1}^{|\mathbf{x}|}$ in (5) incurs extra per-iteration overhead. Ni et al. (2026) suggest that reasoning is steered by a sparse set of high-entropy forking tokens, and Jin et al. (2026) indicate that high-entropy tokens in the teacher model are not merely noise but encode important knowledge in the form of multiple plausible reasoning paths and meaningful uncertainty. Thus, we evaluate ρ_t^θ only at the top-25% highest-entropy positions in the teacher model, eliminating 75% of these forward evaluations. Our final ARM-to-DLM OPD objective is

$$\max_{\theta} \mathcal{L}(\theta) := \mathbb{E}_{\mathbf{q} \sim \mathcal{D}, \mathbf{x} \sim \pi_{\theta_{\text{old}}}(\cdot | \mathbf{q})} \frac{1}{|\mathcal{H}^{\text{te}}|} \sum_{t \in \mathcal{H}^{\text{te}}} \min \left(\rho_t^\theta \cdot A_t, \text{clip}(\rho_t^\theta, 1 - \epsilon, 1 + \epsilon) \cdot A_t \right), \quad (6)$$

where $\mathcal{H}^{\text{te}} := \{t \in [|\mathbf{x}|] : H_t^{\text{te}} \in \text{Top}_{25\%}(\{H_s^{\text{te}}\}_{s=1}^{|\mathbf{x}|})\}$ with $H_t^{\text{te}} := -\sum_{x \in \mathcal{V}} \pi_{\text{te}}(x | \mathbf{q}, \mathbf{x}_{<t}) \log \pi_{\text{te}}(x | \mathbf{q}, \mathbf{x}_{<t})$ being the teacher’s token-level entropy at position t . The JustOPD algorithm is presented in Algorithm 1.

Remark 1. Following the argument of Ni et al. (2026), the autoregressive-style sampling in JustOPD does not turn the DLM into an ARM. We use autoregressive sampling only during training, to match the sampling mode of the teacher ARM. At test time, the model retains its any-order generation capability, and its architecture, including bidirectional attention, is unchanged. $\square$

Algorithm 1 JustOPD for DLMs: Sampled-Token Distillation

Input: ARM teacher policy π_{te} , DLM student policy π_θ , PPO clip ϵ , learning rate η , prompt dataset $\mathcal{D}$

```
1: for each training iteration do
2:   Set  $\pi_{\theta_{\text{old}}} \leftarrow \pi_\theta$ 
3:   Sample a prompt batch  $\mathcal{B} \subset \mathcal{D}$ 
4:   Rollout buffer  $\mathcal{R} \leftarrow \emptyset$ 
5:   for each prompt  $\mathbf{q} \in \mathcal{B}$  do
6:     Sample  $\mathbf{x} = (x^1, \dots, x^{|\mathbf{x}|}) \sim \pi_{\theta_{\text{old}}}(\cdot \mid \mathbf{q})$ 
7:     for  $t = 1$  to  $|\mathbf{x}|$  do
8:       Query teacher:  $(\pi_{\text{te}}(x^t \mid \mathbf{q}, \mathbf{x}^{<t}), H_t^{\text{te}})$ 
9:     end for
10:     $\mathcal{H}^{\text{te}} \leftarrow \{t \in [|\mathbf{x}|] : H_t^{\text{te}} \in \text{Top}_{25\%}(\{H_s^{\text{te}}\}_{s=1}^{|\mathbf{x}|})\}$ 
11:    Store trajectory-level data:  $(\mathbf{q}, \mathbf{x}, \mathcal{H}^{\text{te}}, \{\pi_{\text{te}}(x^t \mid \mathbf{q}, \mathbf{x}^{<t})\}_{t \in \mathcal{H}^{\text{te}}}) \rightarrow \mathcal{R}$ 
12:  end for
13:  for each mini-batch gradient step do
14:    Sample a mini-batch of prompts  $\mathcal{B}_{\text{mini}} \subset \mathcal{R}$ 
15:    Compute the loss using (6):
```

$$\mathcal{L}(\theta) = \frac{1}{|\mathcal{B}_{\text{mini}}|} \sum_{(\mathbf{q}, \mathbf{x}) \in \mathcal{B}_{\text{mini}}} \frac{1}{|\mathcal{H}^{\text{te}}|} \sum_{t \in \mathcal{H}^{\text{te}}} \min(\rho_t^\theta \cdot A_t, \text{clip}(\rho_t^\theta, 1 - \epsilon, 1 + \epsilon) \cdot A_t)$$

```
16:    Update parameters:  $\theta \leftarrow \theta + \eta \nabla_\theta \mathcal{L}(\theta)$ 
17:  end for
18: end for
```

Remark 2 (Clipping Teacher-to-Student Probability Ratio). The variance of the single-sample MC estimator is governed by how fast A_t grows in u_t , i.e., on tokens to which the teacher assigns much higher probability than the student. For reverse KL and Jensen-Shannon divergence, A_t grows only logarithmically in u_t , so the single-sample estimator is well behaved. For the α -divergence, A_t grows as u_t^α , interpolating between logarithmic growth as $\alpha \rightarrow 0^+$ and linear growth as $\alpha \rightarrow 1^-$ (see Figure 1). For forward KL, $A_t = u_t$ grows linear in u_t and its second moment $\mathbb{E}_{x^t \sim \pi_\theta}[u_t^2] = \sum_v \pi_{\text{te}}(v \mid \mathbf{q}, \mathbf{x}^{<t})^2 / \pi_\theta(v \mid \mathbf{q}, \mathbf{x}^{<t})$ can be very large whenever the student underweights tokens favored by the teacher, which leads to high-variance gradients and unstable training. For such teacher-weighted divergences, it is therefore preferable to compute the divergence at the vocabulary level, either over the full vocabulary with clipping or over the teacher’s Top- K tokens. Since we focus on sampled-token distillation in the experiments, we clip $\log u_t$ to $[-5, 5]$ to stabilize training for the forward KL and the α -divergence. $\square$

Black-Box OPD for DLMs. When the teacher model is a black box that can only be queried without access to its logits, we consider full-vocabulary distillation with the forward KL divergence ($f(u) = u \log u$). In this case, the on-policy training objective in (3) can be written without explicitly

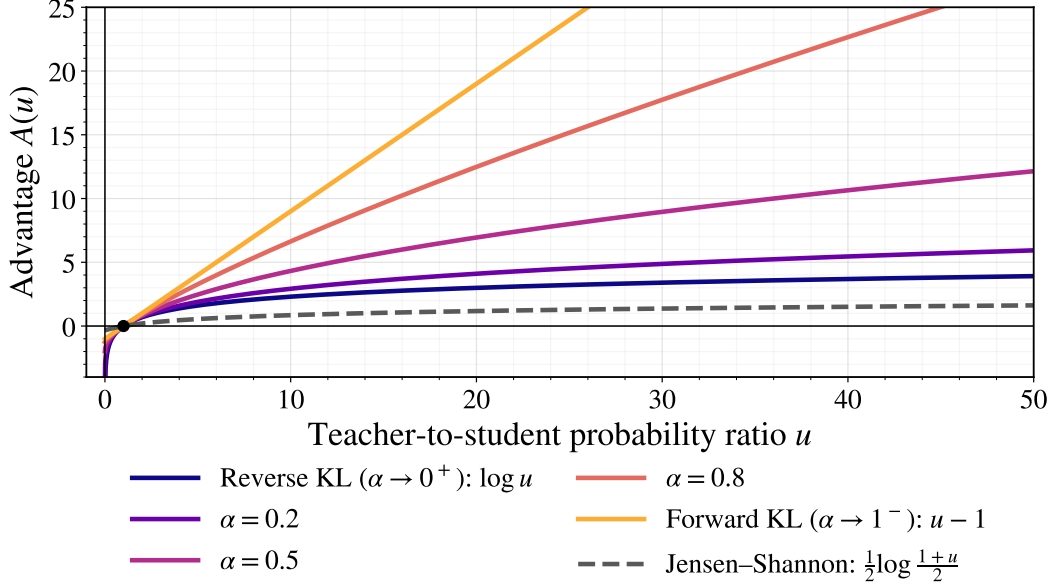


Figure 1: The α -divergence advantage interpolates between the logarithmic growth of reverse KL and the linear growth of forward KL. The Jensen-Shannon advantage also grows logarithmically.

estimating the KL divergence as follows:

$$\begin{aligned}
& \min_{\theta} \mathbb{E}_{\mathbf{q} \sim \mathcal{D}, \mathbf{x} \sim \pi_{\theta_{\text{old}}}(\cdot | \mathbf{q})} \frac{1}{|\mathbf{x}|} \sum_{t=1}^{|\mathbf{x}|} D_{\text{KL}}(\pi_{\text{te}}(\cdot | \mathbf{q}, \mathbf{x}^{<t}) \| \pi_{\theta}(\cdot | \mathbf{q}, \mathbf{x}^{<t})) \\
&= \min_{\theta} \mathbb{E}_{\mathbf{q} \sim \mathcal{D}, \mathbf{x} \sim \pi_{\theta_{\text{old}}}(\cdot | \mathbf{q})} \frac{1}{|\mathbf{x}|} \sum_{t=1}^{|\mathbf{x}|} \mathbb{E}_{y^t \sim \pi_{\text{te}}(\cdot | \mathbf{q}, \mathbf{x}^{<t})} \log(\pi_{\text{te}}(y^t | \mathbf{q}, \mathbf{x}^{<t}) / \pi_{\theta}(y^t | \mathbf{q}, \mathbf{x}^{<t})) \\
&= \min_{\theta} \mathbb{E}_{\mathbf{q} \sim \mathcal{D}, \mathbf{x} \sim \pi_{\theta_{\text{old}}}(\cdot | \mathbf{q})} \frac{1}{|\mathbf{x}|} \sum_{t=1}^{|\mathbf{x}|} - \mathbb{E}_{y^t \sim \pi_{\text{te}}(\cdot | \mathbf{q}, \mathbf{x}^{<t})} \log \pi_{\theta}(y^t | \mathbf{q}, \mathbf{x}^{<t}) + C', \tag{7}
\end{aligned}$$

where C' is a constant that is not dependent on θ . The objective (7) is similar to supervised fine-tuning (SFT) objective (Sun et al., 2026), which gives us a black box OPD algorithm for DLMs by sampling tokens from the teacher (Algorithm 3 in Appendix B). The inner expectation in (7) is approximated via the sampled tokens from teacher by Monte Carlo estimation. In addition, since we do not have access to the logits of teacher model, we cannot compute the teacher entropy as in Algorithm 1. To reduce the number of forward passes required for loss computation, we consider trajectory subsampling by randomly sampling a subset of time steps $\mathcal{T} \subset [|\mathbf{x}|]$ along each trajectory $\mathbf{x}$ and computing the loss only at the sampled time steps (Zhang et al., 2026):

$$\min_{\theta} \mathbb{E}_{\mathbf{q} \sim \mathcal{D}, \mathbf{x} \sim \pi_{\theta_{\text{old}}}(\cdot | \mathbf{q})} \frac{1}{|\mathcal{T}|} \sum_{t \in \mathcal{T}} - \mathbb{E}_{y^t \sim \pi_{\text{te}}(\cdot | \mathbf{q}, \mathbf{x}^{<t})} \log \pi_{\theta}(y^t | \mathbf{q}, \mathbf{x}^{<t}). \tag{8}$$

4 Experiments

4.1 Setup

The choice of teacher and student should be carefully chosen because ARMs and DLMs usually use different tokenizers. Dream-7B (Ye et al., 2025) is a full-attention diffusion language model

initialized from a pretrained ARM checkpoint (Qwen2.5-7B). It inherits the Qwen tokenizer, so it shares its vocabulary with the Qwen model family. This allows token-level distillation from Qwen models without complex vocabulary alignment. We therefore use Dream-7B as the student and Qwen3-14B (Yang et al., 2025) as the teacher in all experiments.

Teacher-student gap under restricted generation lengths. OPD is expected to be most effective when there is a clear performance gap between teacher and student. DLMS are typically trained and decoded with much shorter sequence lengths than ARMs. As a result, the teacher is queried on relatively short sequences generated by the DLM (e.g., $L \leq 1024$), rather than the much longer sequences (e.g., $L = 8192$) that ARMs normally use. We therefore first evaluate both models on two mathematical reasoning benchmarks, GSM8K (Cobbe et al., 2021) and MATH500 (Lightman et al., 2024), with generation lengths restricted to $L \in \{256, 512, 1024\}$. We report accuracy in Table 2. Throughout this work, Dream-7B is evaluated with $L/2$ denoising steps, temperature 0.1, top- p sampling with $p = 0.9$, and the entropy-based unmasking strategy, which unmask the 2 lowest-entropy positions at each denoising step. Evaluation therefore uses the DLM’s native arbitrary-order decoding, even though training rollouts are generated left to right.

Table 2 shows that the teacher–student gap depends strongly on L , and the two models move in opposite directions. Qwen3-14B degrades sharply at short lengths. At $L = 256$ it reaches only 51.71% on GSM8K and 14.40% on MATH500, far below Dream-7B (77.03% and 31.80%). The likely cause is that Qwen3-14B is trained to produce much longer responses: its native context length is 32K tokens, compared with 2K for Dream-7B. Many of its responses are therefore truncated before reaching a final answer. We also see that Dream-7B degrades at long lengths of $L = 1024$ on both tasks. Based on these observations, we use the original Qwen3-14B as the teacher at $L = 512$ for GSM8K and at $L = 1024$ for MATH500.

	GSM8K		MATH500	
Accuracy	Dream-7B	Qwen3-14B	Dream-7B	Qwen3-14B
$L = 256$	77.03	51.71	31.80	14.40
$L = 512$	78.77	93.93	48.20	48.20
$L = 1024$	56.25	94.84	36.80	70.80

Table 2: Performance of Dream-7B and Qwen3-14B with different generation lengths L .

Making the teacher effective at short lengths. To study distillation in the short-sequence regime, where DLMS are most efficient, we fine-tune a separate Qwen3-14B model for each task. Each model is trained to produce shorter answers (e.g., fewer than 256 tokens) while largely preserving accuracy. We denote the resulting models by Qwen3-14B-FT; fine-tuning details are provided in Appendix D.1. As shown in Table 3, fine-tuning raises Qwen3-14B’s accuracy at $L = 256$ from 51.71% to 93.48% on GSM8K and from 14.40% to 53.20% on MATH500. Both now exceed Dream-7B by a wide margin, which makes Qwen3-14B-FT a suitable teacher at $L = 256$.

4.2 Results

We fine-tune Dream-7B with JustOPD for 100 training steps, evaluate every 10 steps, and report the best checkpoint. We compare α -divergences with $\alpha \in \{0, 0.2, 0.5, 0.8, 1\}$ at $L = 256$, where $\alpha = 0$ recovers the reverse KL and $\alpha = 1$ the forward KL, as well as the Jensen–Shannon divergence (JSD). Further training details are given in Appendix D.2. Table 4 reports results with Qwen3-14B-FT

	GSM8K		MATH500	
Accuracy	Dream-7B	Qwen3-14B-FT	Dream-7B	Qwen3-14B-FT
$L = 256$	77.03	93.48	31.80	53.20

Table 3: Performance of Dream-7B and fine-tuned Qwen3-14B with a short sequence length of $L = 256$.

as the teacher at $L = 256$. Table 5 reports results with the original Qwen3-14B as the teacher at $L = 512$ (GSM8K) and $L = 1024$ (MATH500), where we choose $\alpha \in \{0, 0.2, 0.5, 0.8, 1\}$ due to the time cost at longer sequences.

Acc. ($L = 256$)	Dream-7B	$\alpha = 0$	$\alpha = 0.2$	$\alpha = 0.5$	$\alpha = 0.8$	$\alpha = 1$	JSD
GSM8K	77.03	<u>80.97</u>	81.80	80.36	80.67	79.68	80.52
MATH500	31.80	38.80	36.60	38.40	39.60	38.40	<u>39.40</u>

Table 4: Performance of Dream-7B before and after OPD with Qwen3-14B-FT on GSM8K and MATH500 on the short sequence length of $L = 256$ across different α -divergences and Jensen-Shannon divergence (JSD). $\alpha = 0$ is the reverse KL and $\alpha = 1$ is the forward KL.

Acc.	Dream-7B	$\alpha = 0$	$\alpha = 0.5$	$\alpha = 1$	JSD
GSM8K ($L = 512$)	78.77	84.45	<u>83.24</u>	79.23	81.96
MATH500 ($L = 1024$)	36.80	43.80	<u>46.20</u>	42.80	48.00

Table 5: Performance of Dream-7B before and after OPD on GSM8K and MATH500 with Qwen3-14B across different α -divergences and JSD. The generation length L is specified for each benchmark.

JustOPD consistently improves the student. In every setting, JustOPD improves over the base Dream-7B, across all divergences, both benchmarks, and all three generation lengths. These gains are measured with the DLM’s native entropy-based, arbitrary-order decoder, even though the student was trained only on left-to-right rollouts. Knowledge distilled through autoregressive rollouts therefore transfers to any-order inference, which supports Remark 1: JustOPD aligns training with the ARM teacher without constraining the DLM’s generation mode at test time.

Distillation partially offsets the student’s length-dependent weaknesses. At $L = 256$, the JustOPD-trained student reaches 38.80% on MATH500 with reverse KL and up to 39.60% with $\alpha = 0.8$. Both exceed the base Dream-7B at $L = 1024$ (36.80%) while generating a quarter as many tokens. At $L = 1024$, the best distilled model (48.00% with JSD) matches the base model’s accuracy at $L = 512$ (48.20%). This suggests that a large part of the gain at long lengths comes from mitigating the degradation Dream-7B shows when asked to generate longer responses.

Effect of the divergence. No single divergence is best across all settings, but there are some consistent patterns. Mode-seeking divergences work well on GSM8K. In Table 5, reverse KL gives by far the largest improvement (+6.68), followed by $\alpha = 0.5$ (+4.47) and JSD (+3.19). The forward KL barely improves on the base model (+0.46). In Table 4, the spread on GSM8K is much narrower (79.68% to 81.80%), but the forward KL is again the weakest, which matches Remark 2. With sampled-token estimation, the forward-KL advantage $A_t = u_t$ is linear in u_t , so the gradient has high

variance whenever the student underweights tokens the teacher favors. Clipping $\log u_t$ stabilizes training but may also discard useful signal. Divergences that interpolate between the two KL directions work well on MATH500. At $L = 1024$, JSD (48.00%) and $\alpha = 0.5$ (46.20%) outperform both the forward KL (44.60%) and the reverse KL (43.80%); at $L = 256$, the best results are from $\alpha = 0.8$ (39.60%) and JSD (39.40%). One possible explanation is that MATH500 solutions involve more branching reasoning, making the teacher’s next-token distributions more likely to assign substantial probability to multiple plausible continuations. In this case, an objective that does not overly favor the most likely continuation may better preserve the teacher’s distribution over plausible alternatives. JSD is a robust choice, which improves every setting and gives the best MATH500 result at $L = 1024$. Its advantage grows only logarithmically in u_t , which keeps the single-sample estimator well behaved.

Overall, these results show that ARM-to-DLM on-policy distillation with autoregressive rollouts is effective. A stronger ARM teacher can provide useful dense supervision to a DLM student, provided the teacher’s generation length is matched to the regime in which the student operates.

5 Conclusions and Future Works

We studied on-policy distillation from autoregressive to diffusion language models and proposed JustOPD. JustOPD resolves the mismatch between the two generation paradigms by using left-to-right rollouts for the DLM student during training only. This lets an ARM teacher supervise every intermediate student state, while the student keeps its arbitrary-order decoding at inference. Experiments with Dream-7B as the student and Qwen3-14B as the teacher show consistent gains on GSM8K and MATH500 across f -divergences and generation lengths.

Several promising directions remain for future work. First, preliminary results on AIME 2024 show that JustOPD enables Dream-7B to achieve a nonzero Pass@16, whereas the base model solves no problems (Table 8). Future work could evaluate JustOPD on more challenging reasoning benchmarks, with longer generation lengths and stronger teachers. Second, JustOPD could be extended to block-attention DLMs, such as Efficient-DLM (Fu et al., 2026a) and SDAR (Cheng et al., 2026), whose semi-autoregressive structure may be particularly well suited to distillation from ARM teachers. Third, future work would evaluate the black-box OPD for DLMs, which would enable distillation from teachers whose internal logits are inaccessible. Finally, the current token-level distillation approach assumes that the teacher and student share a tokenizer, limiting the range of compatible model pairs. Incorporating cross-tokenizer distillation techniques (Patiño et al., 2025) could remove this constraint and broaden JustOPD’s applicability.

Acknowledgment

Tang is supported by NSF CAREER Award DMS-2538791, NSF MFAI Award DMS-2602038 and the Tang Family Assistant Professorship. Sheng and Zhang are supported by NSF CAREER Award DMS-2538791. This work is part of a Columbia-CityU/HK collaborative project that is supported by InnoHK Initiative, The Government of the HKSAR and the AIFT Lab.

References

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-

- generated mistakes. In *International Conference on Learning Representations*, volume 2024, pages 21246–21263, 2024.
- Kazuoki Azuma. Weighted sums of certain dependent random variables. *Tohoku Mathematical Journal, Second Series*, 19(3):357–367, 1967.
- Heli Ben-Hamu, Itai Gat, Daniel Severo, Niklas S Nolte, and Brian Karrer. Accelerated sampling from masked diffusion models via entropy bounded unmasking. *Advances in Neural Information Processing Systems*, 38:55981–56007, 2026.
- Andrew Campbell, Joe Benton, Valentin De Bortoli, Thomas Rainforth, George Deligiannidis, and Arnaud Doucet. A continuous time framework for discrete denoising models. *Advances in Neural Information Processing Systems*, 35:28266–28279, 2022.
- 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)*, pages 11305–11315. IEEE, 2022.
- Shuang Cheng, Yihan Bian, Dawei Liu, Yuhua Jiang, Yihao Liu, Linfeng Zhang, Qian Yao, Zhongbo Tian, Wenhai Wang, Qipeng Guo, et al. Sdar: A synergistic diffusion-autoregression paradigm for scalable sequence generation. In *Findings of the Association for Computational Linguistics: ACL 2026*, pages 22058–22075, 2026.
- Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, and Reiichiro Nakano. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.
- Phuong Tuan Dat, Qi Li, and Xinchao Wang. dopsd: On-policy self-distillation for diffusion language models. *arXiv preprint arXiv:2607.04428*, 2026.
- Yi Ding and Ruqi Zhang. Does on-policy distillation really distill? from noisy teacher to self-improvement. *arXiv preprint arXiv:2608.31046*, 2026.
- David A Freedman. On tail probabilities for martingales. *The Annals of Probability*, pages 100–118, 1975.
- Yonggan Fu, Lexington Whalen, Zhifan Ye, Xin Dong, Shizhe Diao, Jingyu Liu, Chengyue Wu, Hao Zhang, Enze Xie, Song Han, Maksim Khadkevich, Jan Kautz, Yingyan Celine Lin, and Pavlo Molchanov. Efficient-DLM: From autoregressive to diffusion language models, and beyond in speed. In *Forty-third International Conference on Machine Learning*, 2026a. URL <https://openreview.net/forum?id=ynRn0qI2W8>.
- Zixuan Fu, Bingxiang He, Yuxin Zuo, Haohuan Huang, Jinqian Zhang, Ruhang Xiao, Cheng Qian, Qinyu Luo, Huan-ang Gao, Yudong Wang, Zhiyuan Liu, Ning Ding, and Chaojun Xiao. Rethinking on-policy distillation of large language models II: One training example. *arXiv preprint arXiv:2609.04172*, 2026b.
- Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. Minillm: Knowledge distillation of large language models. In *International Conference on Learning Representations*, volume 2024, pages 32694–32717, 2024.
- Peter Hall, Christopher C Heyde, ZW Birnbaum, and E Lukacs. *Martingale limit theory and its application*, volume 12. Elsevier, 1980.

- Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American statistical association*, 58(301):13–30, 1963.
- Jianguo Huang, Lipeng Wan, Yanchen Deng, and Bo An. Hesitation-aware on-policy distillation for diffusion language models. *arXiv preprint arXiv:2609.33301*, 2026.
- Woogyoul Jin, Taywon Min, Yongjin Yang, Dennis Wei, Yi Zhou, Swanand Ravindra Kadhe, Nathalie Baracaldo, and Kimin Lee. Entropy-aware on-policy distillation of language models. In *Forty-third International Conference on Machine Learning*, 2026. URL <https://openreview.net/forum?id=J5i09fa00f>.
- Jaeyeon Kim, Kulin Shah, Vasilis Kontonis, Sham M. Kakade, and Sitan Chen. Train for the worst, plan for the best: Understanding token ordering in masked diffusions. In *Forty-second International Conference on Machine Learning*, 2025. URL <https://openreview.net/forum?id=DjJmre5IkP>.
- Yaxuan Li, Yuxin Zuo, Bingxiang He, Jinqian Zhang, Chaojun Xiao, Cheng Qian, Tianyu Yu, Huan-ang Gao, Wenkai Yang, Zhiyuan Liu, and Ning Ding. Rethinking on-policy distillation of large language models: Phenomenology, mechanism, and recipe. *arXiv preprint arXiv:2604.13016*, 2026.
- Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let’s verify step by step. In *International Conference on Learning Representations*, volume 2024, pages 39578–39601, 2024.
- Junming Liu, Jicheng Wang, Yifeng He, Hao Chen, and Jianzhong Qi. Forkleft: Entropy-first rollouts for prefix-aligned autoregressive-to-diffusion distillation. *arXiv preprint arXiv:2609.32448*, 2026.
- Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion language modeling by estimating the ratios of the data distribution. In *International Conference on Machine Learning*, 2024.
- Kevin Lu and Thinking Machines Lab. On-policy distillation. *Thinking Machines Lab: Connectionism*, 2025. doi: 10.64434/tml.20251026. <https://thinkingmachines.ai/blog/on-policy-distillation>.
- Yifu Luo, Zeyu Chen, Haoyu Wang, Xinhao Hu, Yuxuan Zhang, Zhizhou Sha, and Shiwei Liu. Learning from the self-future: On-policy self-distillation for dllms. *arXiv preprint arXiv:2606.18195*, 2026.
- Zanlin Ni, Shenzhi Wang, Yang Yue, Tianyu Yu, Weilin Zhao, Yeguo Hua, Tianyi Chen, Jun Song, Cheng Yu, Bo Zheng, and Gao Huang. The flexibility trap: Rethinking the value of arbitrary order in diffusion language models. In *Forty-third International Conference on Machine Learning*, 2026. URL <https://openreview.net/forum?id=kpgURPRMGf>.
- 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.
- Jingyang Ou, Jiaqi Han, Minkai Xu, Shaoxuan Xu, Jianwen Xie, Stefano Ermon, Yi Wu, and Chongxuan Li. Principled RL for diffusion LLMs emerges from a sequence-level perspective. In *The Fourteenth International Conference on Learning Representations*, 2026. URL <https://openreview.net/forum?id=S5YeC9l1lL>.

- Carlos Miguel Patiño, Kashif Rasul, Quentin Gallouédec, Ben Burtenshaw, Sergio Paniego, Vaibhav Srivastav, Thibaud Frere, Ed Beeching, Lewis Tunstall, Leandro von Werra, and Thomas Wolf. Unlocking on-policy distillation for any model family. <https://huggingface.co/spaces/HuggingFaceH4/on-policy-distillation>, 2025.
- Haolin Ren, Ziyang Huang, Chenhao Yuan, Jun Zhao, and Kang Liu. Trace-based on-policy distillation for masked diffusion language models. *arXiv preprint arXiv:2607.16872*, 2026.
- 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.
- John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.
- Jiaxin Shi, Kehang Han, Zhe Wang, Arnaud Doucet, and Michalis Titsias. Simplified and generalized masked diffusion for discrete data. *Advances in neural information processing systems*, 37:103131–103167, 2024.
- Mingyang Song and Mao Zheng. A survey of on-policy distillation for large language models. *arXiv preprint arXiv:2604.00626*, 2026.
- Xingyu Su, Jacob Helwig, Shubham Parashar, Atharv Chagi, Lakshmi Jotsna, Degui Zhi, James Caverlee, Dileep Kalathil, and Shuiwang Ji. Data-efficient autoregressive-to-diffusion language models via on-policy distillation. *arXiv preprint arXiv:2606.06712*, 2026.
- Changhui Sun, Lanbo Liu, Hang Lei, Tong Ling, Jiahang Xie, Zhiyong Zheng, Yujia Wang, Hao Liu, Feng Xiao, and Lu et al Liu. Step-level on-policy distillation: Interpolating between on-policy distillation and supervised fine-tuning. *arXiv preprint arXiv:2608.16333*, 2026.
- Guanghan Wang, Yair Schiff, Subham Sahoo, and Volodymyr Kuleshov. Remasking discrete diffusion models with inference-time scaling. *Advances in Neural Information Processing Systems*, 38: 147282–147339, 2026a.
- Guanghan Wang, Gilad Turok, Yair Schiff, Marianne Arriola, and Volodymyr Kuleshov. d2: Improved techniques for training reasoning diffusion language models. In *The Forty-third International Conference on Machine Learning*, 2026b.
- Yinjie Wang, Ling Yang, Bowen Li, Ye Tian, Ke Shen, and Mengdi Wang. Revolutionizing reinforcement learning framework for diffusion large language models. In *International Conference on Learning Representations*, volume 2026, pages 101476–101501, 2026c.
- An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.
- Ling Yang, Ye Tian, Bowen Li, Xinchun Zhang, Ke Shen, Yunhai Tong, and Mengdi Wang. MMaDA: Multimodal large diffusion language models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026a. URL <https://openreview.net/forum?id=wczmXLuLGd>.
- Wenkai Yang, Weijie Liu, Ruobing Xie, Kai Yang, Saiyong Yang, and Yankai Lin. Learning beyond teacher: Generalized on-policy distillation with reward extrapolation. *arXiv preprint arXiv:2602.12125*, 2026b.

- Jiacheng Ye, Zhihui Xie, Lin Zheng, Jiahui Gao, Zirui Wu, Xin Jiang, Zhenguo Li, and Lingpeng Kong. Dream 7b: Diffusion large language models. *arXiv preprint arXiv:2508.15487*, 2025.
- Zichao Yu, Qianshuo Ye, Xu Wang, and Difan Zou. Understanding on-policy distillation: A mechanistic interpretability perspective via sparse crosscoders. *arXiv preprint arXiv:2609.35210*, 2026.
- Yifan Zhang and Team Math-AI. American invitational mathematics examination (aime) 2024, 2024.
- Zikun Zhang, Jiayuan Sheng, David D Yao, and Wenpin Tang. A continuous-time reinforcement learning framework for fine-tuning discrete diffusion models. *arXiv preprint arXiv:2607.14522*, 2026.
- Hanyang Zhao, Dawen Liang, Wenpin Tang, David Yao, and Nathan Kallus. Diffpo: Training diffusion llms to reason fast and furious via reinforcement learning. *arXiv preprint arXiv:2510.02212*, 2025a.
- Hanyang Zhao, Haoxian Chen, Han Lin, Genta Indra Winata, David Yao, and Wenpin Tang. Opd+: Rethinking the advantage design for on-policy distillation. *arXiv preprint arXiv:2606.01039*, 2026a.
- Siyan Zhao, Devaansh Gupta, Qinqing Zheng, and Aditya Grover. d1: Scaling reasoning in diffusion large language models via reinforcement learning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025b. URL <https://openreview.net/forum?id=7ZVR1BFuEv>.
- Siyan Zhao, Zhihui Xie, Mengchen Liu, Jing Huang, Guan Pang, Feiyu Chen, and Aditya Grover. Self-distilled reasoner: On-policy self-distillation for large language models. *arXiv preprint arXiv:2601.18734*, 2026b.

A Inference Strategies of Masked Diffusion Models

For masked diffusion models, at a noise level $\tau \in [0, 1]$, the forward transition kernel factorizes across dimensions as $p_{\tau|0}(\mathbf{x}_\tau | \mathbf{x}_0) = \prod_{i=1}^L p_{\tau|0}(x_\tau^i | x_0^i)$, where

$$p_{\tau|0}(x_\tau^i | x_0^i) = \text{Cat}(x_\tau^i; \alpha_\tau \mathbf{e}_{x_0^i} + (1 - \alpha_\tau) \mathbf{e}_m),$$

where α_τ is the noise scheduled to satisfy $\alpha_0 \approx 1$ and $\alpha_1 \approx 0$. Given $\mathbf{x}_0 \sim p_{\text{data}}$, for $0 \leq \sigma \leq \tau \leq 1$, the reverse process gradually recovers the clean data from the pure mask, with the transition probability (Shi et al., 2024) given by

$$p_{\sigma|\tau,0}(x_\sigma^i | \mathbf{x}_\tau, \mathbf{x}_0) = \begin{cases} \text{Cat}(x_\sigma^i; \mathbf{e}_{x_\tau^i}), & x_\tau^i \neq m; \\ \text{Cat}(x_\sigma^i; \frac{1-\alpha_\sigma}{1-\alpha_\tau} \mathbf{e}_m + \frac{\alpha_\sigma - \alpha_\tau}{1-\alpha_\tau} \mathbf{e}_{x_0^i}), & x_\tau^i = m. \end{cases} \quad (9)$$

We see that conditional on $\mathbf{x}_0$, a masked token $x_\tau^i = m$ will jump to x_0^i at time σ with probability $\frac{\alpha_\sigma - \alpha_\tau}{1 - \alpha_\tau}$ and then remains the same until $\tau = 0$.

At inference time, starting from a fully masked sequence $\mathbf{x}_0$, a clean sequence $\mathbf{x}$ is unmasked via an iterative trajectory $(\mathbf{x}_t)_{t=0}^T$ with $\mathbf{x}_0 = (m, \dots, m) \in \{m\}^L$, where each intermediate $\mathbf{x}_t$ is obtained from $\mathbf{x}_{t-1}$ by unmasking a subset of its masked positions while keeping all previously unmasked tokens fixed¹, and $\mathbf{x}_T (= \mathbf{x})$ is the final clean output, where $T (\leq L) \in \mathbb{Z}_+$ is the number of denoising steps. According to (9), the vanilla random sampling (Shi et al., 2024; Nie et al., 2026) is: for $t = 1, \dots, T$,

- Sample a set of masked positions $\mathcal{S}_t \subset \mathcal{M}(\mathbf{x}_{t-1}) := \{i \in [L] : x_{t-1}^i = m\}$, where each $i \in \mathcal{M}(\mathbf{x}_{t-1})$ is included independently with probability

$$\mathbb{P}(i \in \mathcal{S}_t) = \frac{\alpha_t - \alpha_{t-1}}{1 - \alpha_{t-1}},$$

where $0 = \alpha_0 < \alpha_1 < \dots < \alpha_T = 1$ is the unmasking schedule (so that, in expectation, a fraction α_t of the tokens is unmasked after step t).

- For each $i \in \mathcal{S}_t$, sample $x_t^i \sim \text{Cat}(\cdot; \mathbf{p}_\theta^{(i)}(\mathbf{x}_{t-1}))$ independently across positions, and set $x_t^i = x_{t-1}^i$ for all $i \notin \mathcal{S}_t$.

In contrast, adaptive sampling (Kim et al., 2025) replaces the random selection of positions by a deterministic (or model-dependent) oracle $\mathcal{F}$, i.e., $\mathcal{S}_t = \mathcal{F}(\theta, \mathbf{x}_{t-1}) \subset \mathcal{M}(\mathbf{x}_{t-1})$, while the second step stays the same. A common choice of $\mathcal{F}$ is the top-probability (confidence) strategy (Chang et al., 2022): to unmask K_t positions at step t (i.e., $|\mathcal{S}_t| = K_t$), each masked position $i \in \mathcal{M}(\mathbf{x}_{t-1})$ is assigned the confidence score

$$c_i := \max_{j \in \mathcal{V}} p_\theta^{(i)}(x^i = j | \mathbf{x}_{t-1}) \in [0, 1],$$

and $\mathcal{F}(\theta, \mathbf{x}_{t-1}) = \text{TopK}_{K_t}(\{c_i\}_{i \in \mathcal{M}(\mathbf{x}_{t-1})})$ selects the K_t most confident masked positions. Another widely used choice is the lowest-entropy strategy (Ye et al., 2025): each masked position $i \in \mathcal{M}(\mathbf{x}_{t-1})$ is assigned the entropy of its predicted marginal,

$$H_i := - \sum_{j \in \mathcal{V}} p_\theta^{(i)}(x^i = j | \mathbf{x}_{t-1}) \log p_\theta^{(i)}(x^i = j | \mathbf{x}_{t-1}) \in [0, \log |\mathcal{V}|],$$

¹We do not consider remasking for correction (Wang et al., 2026a) here.

and $\mathcal{F}(\theta, \mathbf{x}_{t-1}) = \text{TopK}_{K_t}(\{-H_i\}_{i \in \mathcal{M}(\mathbf{x}_{t-1})})$ selects the K_t masked positions with the smallest entropy. Unlike the confidence score c_i , which only looks at the most likely token, H_i accounts for the whole predictive distribution. A position is therefore considered certain only when the probability mass is concentrated on few tokens, not merely when one token is ahead of the rest. The autoregressive-type sampling procedure in (2) corresponds to the intermediate state

$$\mathbf{x}_t = (\mathbf{q}, \mathbf{x}^{\leq t}, \mathbf{m}, \dots, \mathbf{m}), \quad t \in [T],$$

with the final clean sequence given by $\mathbf{x} = \mathbf{x}_T$.

B Algorithms

JustOPD algorithms for Full/TopK-Vocabulary Distillation and Black-Box Teacher Distillation are presented in Algorithms 2 and 3, respectively.

Algorithm 2 JustOPD for DLMS: Full/TopK-Vocabulary Distillation

Input: ARM teacher policy π_{te} , DLM student policy π_θ , f -divergence clip C or Topk threshold K , learning rate η , prompt dataset $\mathcal{D}$

```

1: Select  $\dagger \in \{\text{clip}, \text{TopK}\}$ 
2: for each training iteration do
3:   Set  $\pi_{\theta_{\text{old}}} \leftarrow \pi_\theta$ 
4:   Sample a prompt batch  $\mathcal{B} \subset \mathcal{D}$ 
5:   Rollout buffer  $\mathcal{R} \leftarrow \emptyset$ 
6:   for each prompt  $\mathbf{q} \in \mathcal{B}$  do
7:     Sample  $\mathbf{x} = (x^1, \dots, x^{|\mathbf{x}|}) \sim \pi_{\theta_{\text{old}}}(\cdot \mid \mathbf{q})$ 
8:     for  $t = 1$  to  $|\mathbf{x}|$  do
9:       if  $\dagger = \text{TopK}$  then
10:        Query teacher:  $Q_t = (\pi_{\text{te}}(\cdot \mid \mathbf{q}, \mathbf{x}^{\leq t}), \text{TopK}(\pi_{\text{te}}(\cdot \mid \mathbf{q}, \mathbf{x}^{\leq t}))), H_t^{\text{te}}$ 
11:       else
12:        Query teacher:  $Q_t = \pi_{\text{te}}(\cdot \mid \mathbf{q}, \mathbf{x}^{\leq t}), H_t^{\text{te}}$ 
13:       end if
14:     end for
15:      $\mathcal{H}^{\text{te}} \leftarrow \{t \in [|\mathbf{x}|] : H_t^{\text{te}} \in \text{Top}_{25\%}(\{H_s^{\text{te}}\}_{s=1}^{|\mathbf{x}|})\}$ 
16:     Store trajectory-level data:  $(\mathbf{q}, \mathbf{x}, \mathcal{H}^{\text{te}}, \{Q_t\}_{t \in \mathcal{H}^{\text{te}}}) \rightarrow \mathcal{R}$ 
17:   end for
18:   for each mini-batch gradient step do
19:     Sample a mini-batch of prompts  $\mathcal{B}_{\text{mini}} \subset \mathcal{R}$ 
20:     Compute the loss using (3):

```

$$\mathcal{L}(\theta) = \frac{1}{|\mathcal{B}_{\text{mini}}|} \sum_{(\mathbf{q}, \mathbf{x}) \in \mathcal{B}_{\text{mini}}} \frac{1}{|\mathcal{H}^{\text{te}}|} \sum_{t \in \mathcal{H}^{\text{te}}} D_f^\dagger(\pi_{\text{te}}(\cdot \mid \mathbf{q}, \mathbf{x}^{\leq t}) \parallel \pi_\theta(\cdot \mid \mathbf{q}, \mathbf{x}^{\leq t}))$$

```

21:     Update parameters:  $\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}(\theta)$ 
22:   end for
23: end for

```

Algorithm 3 JustOPD for DLMs: Black-Box Teacher Distillation

Input: ARM teacher policy π_{te} , DLM student policy π_θ , learning rate η , prompt dataset $\mathcal{D}$

```
1: for each training iteration do
2:   Set  $\pi_{\theta_{\text{old}}} \leftarrow \pi_\theta$ 
3:   Sample a prompt batch  $\mathcal{B} \subset \mathcal{D}$ 
4:   Rollout buffer  $\mathcal{R} \leftarrow \emptyset$ 
5:   for each prompt  $\mathbf{q} \in \mathcal{B}$  do
6:     Sample  $\mathbf{x} = (x^1, \dots, x^{|\mathbf{x}|}) \sim \pi_{\theta_{\text{old}}}(\cdot \mid \mathbf{q})$ 
7:     for  $t = 1$  to  $|\mathbf{x}|$  do
8:       Sample tokens from the teacher:  $Q_t = \{y^t\} \sim \pi_{\text{te}}(\cdot \mid \mathbf{q}, \mathbf{x}^{<t})$ 
9:     end for
10:    Store trajectory-level data:  $(\mathbf{q}, \mathbf{x}, \{Q_t\}_{t \in \mathcal{H}^{\text{te}}}) \rightarrow \mathcal{R}$ 
11:  end for
12:  for each mini-batch gradient step do
13:    Sample a mini-batch  $\mathcal{B}_{\text{mini}} \subset \mathcal{R}$  and a time set  $\mathcal{T} \subset [|\mathbf{x}|]$  for each  $(\mathbf{q}, \mathbf{x}) \in \mathcal{B}_{\text{mini}}$ 
14:    Compute the loss using (8):

$$\mathcal{L}(\theta) = -\frac{1}{|\mathcal{B}_{\text{mini}}|} \sum_{(\mathbf{q}, \mathbf{x}) \in \mathcal{B}_{\text{mini}}} \frac{1}{|\mathcal{T}|} \sum_{t \in \mathcal{T}} \frac{1}{|Q_t|} \sum_{y^t \in Q_t} \log \pi_\theta(y^t \mid \mathbf{q}, \mathbf{x}^{<t})$$

15:    Update parameters:  $\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}(\theta)$ 
16:  end for
17: end for
```

C Proofs

C.1 Proof of Proposition 1

Proof of Proposition 1. We quantify the error incurred by the single-sample approximation in (4) and show that it is governed by the local behavior of f at $u = 1$.

Fix a prompt $\mathbf{q}$ and write the rollout as $\mathbf{x} = (x^1, \dots, x^T) \sim \pi_\theta(\cdot \mid \mathbf{q})$, where $T = |\mathbf{x}|$ is treated as deterministic, as is the case for DLMs decoded with a fixed generation length L .² Let $\mathcal{F}_t := \sigma(x^1, \dots, x^t)$ denote the natural filtration, with $\mathcal{F}_0$ trivial. For notational simplicity, for each position t , let the student and teacher next-token distributions be

$$Q_t(\cdot) := \pi_\theta(\cdot \mid \mathbf{q}, \mathbf{x}^{<t}) \quad \text{and} \quad P_t(\cdot) := \pi_{\text{te}}(\cdot \mid \mathbf{q}, \mathbf{x}^{<t})$$

are $\mathcal{F}_{t-1}$ -measurable, and $x^t \mid \mathcal{F}_{t-1} \sim Q_t$. We write

$$u_t(x) := \frac{P_t(x)}{Q_t(x)} = 1 + \Delta_t(x) \quad \text{and} \quad D_{f,t} := D_f(P_t \parallel Q_t) = \mathbb{E}_{x' \sim Q_t}[f(u_t(x'))].$$

Thus $D_{f,t}$ is the exact inner expectation in (4), while $f(u_t(x^t))$ is its single-sample estimate. The token-level error is

$$\xi_t := f(u_t(x^t)) - D_{f,t}$$

²If T is determined by an end-of-sequence stopping rule, the arguments below apply to the stopped martingale, with minor modifications to the $1/T$ normalization that we omit.

and trajectory-level approximation error is

$$e_T := \frac{1}{T} \sum_{t=1}^T \xi_t = \frac{M_T}{T}, \quad (10)$$

where we write $M_T := \sum_{t=1}^T \xi_t$. Because $D_{f,t}$ is $\mathcal{F}_{t-1}$ -measurable and $x^t \mid \mathcal{F}_{t-1} \sim Q_t$, we have $\mathbb{E}[\xi_t \mid \mathcal{F}_{t-1}] = 0$. Hence, $(\xi_t)_{t \geq 1}$ is a martingale difference sequence and $(M_t)_{t \geq 0}$ is a martingale with respect to $(\mathcal{F}_t)_{t \geq 0}$. This structure holds without any independence across positions: the tokens x^t are strongly dependent through their prefixes, but each error ξ_t is centered conditionally on the past. An immediate consequence is $\mathbb{E}[e_T] = 0$, so the single-sample approximation introduces no bias in the objective value.

Let $\sigma_t^2 := \mathbb{E}[\xi_t^2 \mid \mathcal{F}_{t-1}] = \text{Var}_{x' \sim Q_t} [f(u_t(x'))]$ denote the conditional variance of the single-sample estimate. Since martingale increments are orthogonal, i.e., $\mathbb{E}[\xi_s \xi_t] = \mathbb{E}[\xi_s \mathbb{E}[\xi_t \mid \mathcal{F}_{t-1}]] = 0$ for $s < t$, by the definition of e_T in (10), we have

$$\sqrt{\mathbb{E}[e_T^2]} = \frac{1}{T} \sqrt{\mathbb{E}\left[\sum_{t=1}^T \sigma_t^2\right]}. \quad (11)$$

In particular, if $\mathbb{E}[\sum_{t=1}^T \sigma_t^2] \sim \bar{\sigma}^2 T$, then $\sqrt{\mathbb{E}[e_T^2]} \sim \bar{\sigma}/\sqrt{T}$.

Consider a family of rollouts indexed by $T \rightarrow \infty$ (a martingale difference sequence). Suppose that $\frac{1}{T} \sum_{t=1}^T \sigma_t^2 \xrightarrow{P} \bar{\sigma}^2$ and that the conditional Lindeberg condition holds, i.e., for every $\eta > 0$,

$$\frac{1}{T} \sum_{t=1}^T \mathbb{E}[\xi_t^2 \mathbf{1}\{|\xi_t| > \eta\sqrt{T}\} \mid \mathcal{F}_{t-1}] \xrightarrow{P} 0,$$

where $\mathbf{1}\{\cdot\}$ is the indicator function. Then the martingale central limit theorem (Hall et al., 1980) yields $\sqrt{T} e_T \xrightarrow{d} \mathcal{N}(0, \bar{\sigma}^2)$, i.e., $e_T \approx \mathcal{N}(0, \bar{\sigma}^2/T)$ for long trajectories. In addition, if $|\xi_t| \leq B$ for all t , the Azuma-Hoeffding inequality (Hoeffding, 1963; Azuma, 1967) yields

$$\mathbb{P}(|e_T| \geq \eta) \leq 2 \exp\left(-\frac{T\eta^2}{2B^2}\right), \quad \forall \eta > 0.$$

A sharper variance-sensitive bound follows from Freedman's inequality (Freedman, 1975): for all $\eta, v > 0$, it holds that

$$\mathbb{P}\left(|e_T| \geq \eta, \frac{1}{T} \sum_{t=1}^T \sigma_t^2 \leq v\right) \leq 2 \exp\left(-\frac{T\eta^2}{2(v + B\eta/3)}\right).$$

Hence, a single trajectory of length T approximates the average exact token-level f -divergence, with the probability of an error of size η decaying exponentially in T . These statements, however, are only meaningful relative to the magnitude of the quantity being estimated: in the regime of interest, where the student is already close to the teacher, both $D_{f,t}$ and σ_t are small, and the relevant question is how they scale relative to each other. This is determined by the behavior of f near $u = 1$, which we analyze next.

We analyze each position t conditionally on its prefix and aggregate afterwards. First observe that

$$\mathbb{E}_{Q_t}[\Delta_t] = \sum_x P_t(x) - \sum_x Q_t(x) = 0.$$

Assume that $|\Delta_t(x)| \leq \gamma < 1$ for all x in the support of Q_t , and that f is four times continuously differentiable on a neighborhood of $[1 - \gamma, 1 + \gamma]$ with $f''(1) > 0$. Since $|\Delta_t| \leq \gamma$, we have $\mathbb{E}_{Q_t}|\Delta_t|^k \leq \gamma^{k-2}\chi_t^2$ for all $k \geq 2$, where $\chi_t^2 := \chi^2(P_t \| Q_t) = \mathbb{E}_{Q_t}[\Delta_t^2]$. Expanding $f(1 + \Delta) = f'(1)\Delta + \frac{1}{2}f''(1)\Delta^2 + \frac{1}{6}f'''(1)\Delta^3 + O(\Delta^4)$ and taking expectations under Q_t , the linear term vanishes and

$$D_{f,t} = \frac{1}{2}f''(1)\chi_t^2 + \frac{1}{6}f'''(1)\mathbb{E}_{Q_t}[\Delta_t^3] + O(\gamma^2\chi_t^2) = \frac{1}{2}f''(1)\chi_t^2(1 + O(\gamma)). \quad (12)$$

Thus, locally, every f -divergence is a multiple of the χ^2 -divergence, and in particular, $D_{f,t} = O(\gamma^2)$.

The single-sample error is:

$$\xi_t = f'(1)\Delta_t(x^t) + \frac{1}{2}f''(1)(\Delta_t(x^t)^2 - \chi_t^2) + O(\gamma^3).$$

Squaring, taking conditional expectations, and bounding every remaining term by a constant multiple of $\mathbb{E}_{Q_t}|\Delta_t|^k \leq \gamma^{k-2}\chi_t^2$ with $k \geq 4$, we obtain

$$\sigma_t^2 = f'(1)^2\chi_t^2 + f'(1)f''(1)\mathbb{E}_{Q_t}[\Delta_t^3] + O(\gamma^2\chi_t^2). \quad (13)$$

Consequently, whenever $f'(1) \neq 0$, the noise is of first order, $\sigma_t = |f'(1)|\chi_t(1 + O(\gamma))$, whereas the signal (12) is of second order. Therefore, the local noise-to-signal ratio is

$$\frac{\sigma_t}{D_{f,t}} = \frac{2|f'(1)|}{f''(1)\chi_t}(1 + O(\gamma)),$$

which diverges as the student approaches the teacher. In other words, the better the student already matches the teacher, the less informative each single-sample estimate becomes relative to the quantity it estimates. The source of this deficiency is transparent: the leading term $f'(1)\Delta_t(x^t)$ has zero conditional mean, so it contributes nothing to $D_{f,t}$ while dominating the variance.

The centralized generator $\tilde{f}(u) = f(u) - f(1) - f'(1)(u - 1)$ removes precisely the first-order noise. Because $\mathbb{E}_{Q_t}[u_t] = 1$, it leaves the signal unchanged, $D_{\tilde{f},t} = D_{f,t}$, while modifying the estimator to

$$\tilde{f}(u_t(x^t)) = f(u_t(x^t)) - f'(1)\Delta_t(x^t).$$

Since $\tilde{f}(1) = \tilde{f}'(1) = 0$ and $\tilde{f}'' = f''$, the first two terms of (13) vanish for $\tilde{f}$, and carrying the expansion one order further yields

$$\tilde{\sigma}_t^2 = \frac{1}{4}f''(1)^2 \text{Var}_{Q_t}(\Delta_t^2) + O(\gamma^3\chi_t^2) \leq \frac{1}{4}f''(1)^2 \gamma^2 \chi_t^2 (1 + O(\gamma)),$$

so we have

$$\frac{\tilde{\sigma}_t}{D_{\tilde{f},t}} = \frac{\sqrt{\text{Var}_{Q_t}(\Delta_t^2)}}{\mathbb{E}_{Q_t}[\Delta_t^2]}(1 + O(\gamma)) \leq \frac{\gamma}{\chi_t}(1 + O(\gamma)).$$

The leading factor is the coefficient of variation of Δ_t^2 under Q_t , a scale-free quantity that does not degenerate as the teacher and student approach each other. In the non-degenerate regime $\chi_t \sim \gamma$, in which the teacher and student are $\Theta(\gamma)$ apart, the local orders of magnitude are summarized below.

The condition $|\Delta_t| \leq \gamma$ may be too strong when the teacher assigns much larger probability than the student to a few tokens of small Q_t -mass. However, the analysis above uses γ only through moment bounds of the form $\mathbb{E}_{Q_t}|\Delta_t|^k \lesssim \gamma^{k-2}\chi_t^2$, so it suffices to control χ_t together with the higher moments of Δ_t . Specifically, suppose that for some $\delta \in (0, 1]$,

	f	$\tilde{f}$
Signal $D_{f,t}$	$\Theta(\gamma^2)$	$\Theta(\gamma^2)$
Single-sample variance σ_t^2	$\Theta(\gamma^2)$	$O(\gamma^4)$
Single-sample standard deviation σ_t	$\Theta(\gamma)$	$O(\gamma^2)$
Noise-to-signal ratio $\sigma_t/D_{f,t}$	$\Theta(1/\gamma)$	$O(1)$

Table 6: Comparison of signal and noise scaling for f and $\tilde{f}$.

- (i) the second-order Taylor remainder of f satisfies $|f(1 + \Delta) - f'(1)\Delta - \frac{1}{2}f''(1)\Delta^2| \leq K|\Delta|^{2+\delta}$ on the range of Δ_t ;
- (ii) the moments of Δ_t are comparable to its second moment, i.e., $\mathbb{E}_{Q_t}|\Delta_t|^k \leq C_k(\chi_t^2)^{k/2}$ for $k \in [2, 4 + 2\delta]$.

Then the same Taylor argument yields $D_{f,t} = \frac{1}{2}f''(1)\chi_t^2(1 + O(\chi_t^\delta))$, $\sigma_t^2 = f'(1)^2\chi_t^2(1 + O(\chi_t))$ and $\tilde{\sigma}_t^2 = O(\chi_t^4)$, so all of the conclusions above continue to hold with γ replaced by χ_t . Condition (i) holds for the forward KL. For the reverse KL, it holds whenever u_t is bounded away from zero, i.e., whenever the teacher does not assign vanishing probability to tokens that the student samples with non-negligible probability. $\square$

C.2 Proof of Proposition 2

Proof of Proposition 2. (i) Since P_t does not depend on θ , we have $\nabla_\theta u_{\theta,t} = -u_{\theta,t}\nabla_\theta \log Q_{\theta,t}$. Differentiating both the sampling probability and the ratio, which is valid term by term because the support is finite, and using $\nabla_\theta Q_{\theta,t} = Q_{\theta,t}\nabla_\theta \log Q_{\theta,t}$, gives

$$\begin{aligned}\nabla_\theta D_{f,t}(\theta) &= \sum_x [\nabla_\theta Q_{\theta,t}(x) f(u_{\theta,t}(x)) + Q_{\theta,t}(x) f'(u_{\theta,t}(x)) \nabla_\theta u_{\theta,t}(x)] \\ &= \mathbb{E}_{Q_{\theta,t}} [(f(u_{\theta,t}) - u_{\theta,t} f'(u_{\theta,t})) \nabla_\theta \log Q_{\theta,t}(x)] = -\mathbb{E}_{Q_{\theta,t}} [A_f(u_{\theta,t}) \nabla_\theta \log Q_{\theta,t}(x)].\end{aligned}$$

- (ii) Since $\mathbb{E}_{Q_{\theta,t}}[u_{\theta,t}] = \sum_x P_t(x) = 1$ and $f(1) = 0$,

$$D_{\tilde{f},t}(\theta) = \mathbb{E}_{Q_{\theta,t}}[\tilde{f}(u_{\theta,t})] = \mathbb{E}_{Q_{\theta,t}}[f(u_{\theta,t})] - f(1) - f'(1)(\mathbb{E}_{Q_{\theta,t}}[u_{\theta,t}] - 1) = \mathbb{E}_{Q_{\theta,t}}[f(u_{\theta,t})] = D_{f,t}(\theta).$$

Moreover, $\tilde{f}'(u) = f'(u) - f'(1)$, so

$$A_{\tilde{f}}(u) = -f(u) + f(1) + f'(1)(u - 1) + u(f'(u) - f'(1)) = A_f(u) + f(1) - f'(1) = A_f(u) - f'(1).$$

$\square$

D Experiment Details

D.1 Details on Fine-Tuning Qwen3-14B

To obtain teachers that remain effective under a short generation budget (e.g. $L = 256$), we fine-tune Qwen3-14B in advance for GSM8K and MATH using a length-aware, GRPO-style objective (see Algorithm 4). We use the model’s chat template with thinking disabled and append an instruction asking for concise step-by-step reasoning, without restating the problem. During training, responses are sampled in the auto-regressive way with temperature $T = 1.0$, top- p sampling with $p = 0.95$.

Algorithm 4 Length-Aware Fine-Tuning of Qwen3-14B

Input: Policy π_ϕ , dataset $\mathcal{D}$, correctness evaluator c , group size K , length cap L

```
1: for each training iteration do
2:   Sample a prompt batch  $\mathcal{B} \subset \mathcal{D}$ 
3:   Rollout buffer  $\mathcal{R} \leftarrow \emptyset$ 
4:   for each prompt  $\mathbf{q} \in \mathcal{B}$  do
5:     Construct  $\tilde{\mathbf{q}}$  with the concise-answer instruction
6:     Sample  $K$  responses  $\{\mathbf{x}_i\}_{i=1}^K$  from  $\pi_\phi(\cdot \mid \tilde{\mathbf{q}})$ , each capped at  $L$  tokens
7:     for  $i = 1$  to  $K$  do
8:       Set  $\ell_i = |\mathbf{x}_i|$  and  $c_i = c(\mathbf{q}, \mathbf{x}_i)$ 
9:       Compute reward  $r_i$  using (14).
10:    end for
11:    for  $i = 1$  to  $K$  do
12:      Compute stop-gradient group-normalized advantage  $A_i$ 
13:      Store  $(\tilde{\mathbf{q}}, \mathbf{x}_i, A_i) \rightarrow \mathcal{R}$ 
14:    end for
15:  end for
16:  Compute the loss:
```

$$\mathcal{L}(\phi) = -\frac{1}{|\mathcal{R}|L} \sum_{(\tilde{\mathbf{q}}, \mathbf{x}_i, A_i) \in \mathcal{R}} A_i \cdot \sum_{t=1}^{|\mathbf{x}_i|} \log \pi_\phi(x_i^t \mid \tilde{\mathbf{q}}, \mathbf{x}_i^{<t}).$$

```
17:   Update parameters:  $\phi \leftarrow \phi - \eta \nabla_\phi \mathcal{L}(\phi)$ 
18: end for
```

We reward correct answers with a small brevity bonus that is not applied to incorrect ones. Specifically, let c_i denote the correctness reward returned by the benchmark-specific evaluator for response i , with $c_i > 0$ indicating a correct answer. Let ℓ_i denote its generated length, including the first end-of-sequence token when present. We assign

$$r_i = c_i + \lambda_{\text{len}} \mathbf{1}_{\{c_i > 0\}} \left(1 - \frac{\ell_i}{L_{\text{max}}}\right). \quad (14)$$

Shorter correct responses receive larger rewards, while incorrect responses receive no brevity bonus. In Algorithm 4, we set the brevity coefficient $\lambda = 0.01$ and the learning rate $\eta = 5 \times 10^{-5}$. Group-normalized advantages A_i are divided by the within-group sample standard deviation plus the numerical stabilizer $\epsilon = 10^{-4}$. The advantages are held fixed during gradient computation. Adapter checkpoints are saved every 100 steps. The resulting task-specific teachers are denoted by Qwen3-14B-FT.

D.2 Training Details

We use Dream-v0-Instruct-7B (Ye et al., 2025) as the DLM student and Qwen3-14B (Yang et al., 2025) as the ARM teacher. All experiments are conducted on 7 NVIDIA L40S GPUs. During distillation the teacher stays frozen and is queried in non-thinking mode. Both models receive the same problem through their own chat templates. The student is trained with LoRA (rank $r = 128$, $\alpha = 64$, dropout 0.05). At each step, every GPU draws one prompt and generates 8 on-policy rollouts from the student, giving 7 prompts and 56 rollouts per optimizer step. Rollouts

have sampling temperature 1.0, and are generated in left-to-right order, unmasking one token per forward pass and stopping once every sequence has emitted an end-of-sequence token. The PPO clipping parameter is $\epsilon = 0.2$. For the reverse KL training case, we add a forward-KL term with coefficient 0.1, computed on the teacher’s top-16 tokens and applied only at positions where the teacher entropy exceeds $\tau = 0.8$, following the work of [Jin et al. \(2026\)](#). We optimize with AdamW ($\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$, no weight decay) at a constant learning rate of 5×10^{-5} .

D.3 Additional Experiment Results

We present preliminary results of JustOPD on AIME 2024 ([Zhang and Math-AI, 2024](#)) in Table 8, using reverse KL and training on all historical AIME problems before evaluating on AIME 2024. We observe that AIME 2024 is sufficiently challenging for Dream-7B that it fails to produce a single correct answer, even with a generation length of 1024 tokens (see Table 7). Future work will focus on improving the teacher model’s performance on AIME under shorter generation-length constraints, enabling more effective knowledge transfer to DLM students within their limited context budgets.

Gen. Length	GSM8K		MATH500		AIME 24			
	Acc.		Acc.		Pass@16		Avg@16	
	Stu.	Tr.	Stu.	Tr.	Stu.	Tr.	Stu.	Tr.
$L = 256$	77.03	51.71	31.80	14.40	0.00	0.00	0.00	0.00
$L = 512$	78.77	93.93	48.20	48.20	0.00	3.33	0.00	0.83
$L = 1024$	56.25	94.84	36.80	70.80	0.00	13.33	0.00	4.58

Table 7: Performance of Dream-7B and Qwen3-14B with different generation lengths L . Stu. and Tr. denote the base Dream-7B (student) and Qwen3-14B (teacher) models, respectively.

Model	GSM8K	MATH500	AIME 24	
	Acc. ($L = 512$)	Acc. ($L = 1024$)	Pass@16 ($L = 1024$)	Avg@16 ($L = 1024$)
Dream-7B	78.77	36.80	0.00	0.00
+JustOPD	84.45	43.80	3.33	0.21

Table 8: Performance of Dream-7B before and after OPD on GSM8K, MATH500, and AIME 24. The generation length L is specified for each benchmark.