Rethinking Knowledge Distillation for Diffusion Language Models

Yuxuan Sun[1] **Yuanjian Xu**[2] **Jianing Hao**[2] **Yu Li**[3]
Sowmen Das[4] **Zhong Li**[5] **Sangwoong Yoon**[6]* **Miguel R. D. Rodrigues**[1]*

[1]University College London [2]HKUST [3]Peking University
[4]MediaTek Research UK [5]MSRA [6]UNIST

ys507@cantab.ac.uk

Abstract

Knowledge distillation has recently proven highly effective for improving autoregressive models, but its applicability to DLMs remains poorly understood. We present the first systematic study of KD for DLMs and uncover two non-intuitive failures of classical forward-KL (FKL) distillation: (i) student perplexity does not improve monotonically with more teacher-generated data, and (ii) on the same teacher-generated corpus, FKL distillation underperforms simply training the student with the original DLM denoising objective. To address them, we propose a unified, data-free distillation framework for masked DLMs. First, the framework runs an off-policy distillation phase that absorbs general knowledge from a static set of teacher-generated data. Then, the framework employs on-policy distillation which consists of three techniques: a trajectory loss along the student's own reverse process, confidence reweighting that down-weights tokens where the teacher is unsure, and a trust-region anchor that stabilizes the moving target. On MDLM, the resulting student attains a GPT-2 perplexity of 24.75, surpassing the teacher's 27.04; on Dream-7B distillation to a 93M student model, the framework reduces Qwen-2.5 perplexity from the FKL-distilled baseline of 25.23 to 19.83 while preserving performance on zero-shot general capability benchmarks.

1 Introduction

Discrete diffusion language models (DLMs) have recently emerged as a compelling alternative to autoregressive large language models, supporting parallel generation while remaining competitive in modeling quality [Sahoo et al., 2024, Lou et al., 2024, Nie et al., 2025, Ye et al., 2025]. As frontier DLMs grow larger, deploying them in latency- or memory-constrained settings, such as on-device assistants, edge systems, and robotic systems with limited accelerators, increasingly motivates compressing them into much smaller students. Knowledge distillation (KD) [Hinton et al., 2015] is the canonical tool for this compression. KD has also been applied in modern AR LLM post-training, where distilling from a strong teacher often yields larger gains in reasoning ability than reinforcement learning [Yue et al., 2025, Yang et al., 2025, DeepSeek-AI, 2026]. This raises a natural question: can knowledge distillation be made similarly effective for closing the teacher–student gap in the DLM paradigm?

We study this question in the data-free setting, where the student is trained solely on samples synthesized by the teacher rather than on tokens from a real dataset. Two considerations motivate this choice. First, the training corpora of frontier DLMs are typically inaccessible [Song et al., 2025, Labs et al., 2025], making it important to distill without access to real corpora. Second, a static real-text corpus can give an incomplete or misaligned picture of what the teacher actually generates,

*Co-senior authors.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

since the teacher often generalizes beyond its original training set or has been shifted away from it by post-hoc finetuning. However, teacher-sampled trajectories can faithfully expose the teacher's generative dynamics, which is what the student should match [Tong et al., 2025].

We present the first systematic study of KD for diffusion language models. Applying classical Forward Kullback–Leibler (FKL) distillation, the objective used in the original KD paper [Hinton et al., 2015], to MDLM [Sahoo et al., 2024] and Dream [Ye et al., 2025] reveals two phenomena that contradict standard KD intuition:

- **Non-monotone scaling.** Student perplexity does not improve monotonically with more teacher-generated data. For MDLM, 10M tokens are best, while 30M and 100M are worse, as shown in Figure 1. Dream exhibits the same non-monotone pattern, shown in Table 1.
- **Objective mismatch.** Trained on the same teacher-generated corpus, FKL distillation underperforms simply training the student with the original DLM denoising objective. The same gap is observed for both MDLM and Dream.

We trace both failures to the same root cause: FKL is applied only at states drawn from the forward noising process of teacher samples, whereas during inference the student generates samples through its own reverse process. As more teacher-generated tokens are added, the student fits the forward-noised teacher states ever more tightly, while its sampling-time states drift away from them; the off-policy loss provides no corrective signal in this drifted region. This is precisely the "no implicit prior" issue raised by Sahoo et al. [2025]: in masked DLMs the noise-to-data map is not deterministic, so the states the student visits at sampling time cannot be derived simply by forward-noising teacher samples. The phenomenon also mirrors recent explanations of hallucination in AR LLMs [Kalai et al., 2025], where objectives that uniformly reward coverage of the teacher's full distribution also implicitly reward guessing in regions that the student never visits at inference time.

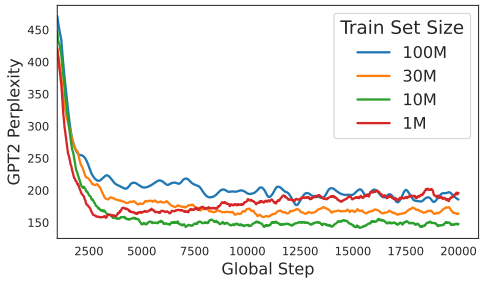

Figure 1: **Scaling failure of forward-KL distillation in MDLM.** GPT-2 perplexity of an MDLM student does not improve monotonically with the size of the teacher-generated corpus: 10M tokens are optimal, while larger sets (30M, 100M) yield worse perplexity. Dream exhibits the same pattern (Table 1).

Building on this diagnosis, we propose a unified, data-free distillation framework for masked DLMs that explicitly aligns training-time states with the student's reverse-process occupancy. The framework has two phases. An *off-policy* phase absorbs general teacher knowledge from a static, teacher-generated corpus. Once it stops improving on a held-out evaluator, training switches to an *on-policy* phase whose loss combines three ingredients: (i) a trajectory loss that supervises the student at multiple intermediate states sampled from its own reverse process, breaking the single-step, single-state bottleneck of FKL; (ii) confidence reweighting that scales each token's loss by the teacher's max-probability, focusing learning on states where the teacher is reliable; and (iii) a trust-region anchor on a periodically refreshed copy of the student that stabilizes optimization against the moving rollout distribution. For the smaller MDLM teacher, where FKL is independently too weak, we further replace FKL with the α-β divergence of Wang et al. [2025] in both phases; for the much stronger Dream-7B teacher, FKL suffices.

Contributions.

1. The first systematic study of KD for DLMs across two representative scale regimes, identifying two failure phenomena of FKL distillation for both a 110M (MDLM) and a 7B (Dream) teacher model (Section 3).
2. A unified, data-free distillation framework combining off- and on-policy phases, a trajectory loss, confidence reweighting, and a trust-region anchor, with explicit choices of divergence (α-β divergence for MDLM, FKL for Dream) motivated by teacher capacity (Section 4).
3. Empirical results: on MDLM, the distilled 10.5M student reaches GPT-2 perplexity 24.75, below the 27.04 of the 110M teacher; on Dream-7B distillation into a 93M student, the

framework reduces Qwen-2.5 perplexity from 25.23 (FKL) to 19.83 while preserving zero-shot benchmark accuracy. Ablations isolate the contribution of each component (Section 5).

2 Background and Related Work

Discrete diffusion language models. A masked discrete DLM defines a forward process that progressively replaces tokens of a clean sequence x_0 with a special $[\texttt{MASK}]$ token according to a schedule indexed by $t \in [0, T]$, and learns a parameterized denoiser $f_\theta(x_t, t)$ that predicts the clean tokens at masked positions. Recent work has advanced DLMs in several directions. Lou et al. [2024] introduced a discrete diffusion model trained with a score entropy objective derived from Bregman divergence. Sahoo et al. [2024] proposed MDLM, a masked diffusion model optimized with a Rao-Blackwellized objective, establishing a new state of the art in DLM performance. LLaDA [Nie et al., 2025] and Dream [Ye et al., 2025] have scaled masked DLMs to 8B and 7B parameters respectively, demonstrating performance competitive with autoregressive models of similar scale.

Knowledge distillation. Classical KD [Hinton et al., 2015] matches a student's output distribution to that of a fixed teacher under a forward KL divergence. Wang et al. [2025] introduce α-β divergence, which interpolates between FKL and reverse KL via two scalars controlling hardness concentration and confidence concentration; we adopt ABKL on MDLM where FKL underperforms. In the AR model setting, several approaches have refined distillation objectives. Gu et al. [2024] introduced MiniLLM, which minimizes reverse KL divergence between teacher and student distributions. Ko et al. [2024] proposed DISTILLM, employing a skew KL divergence and improving sample efficiency through a replay buffer mechanism; Lu and Lab [2025] highlight on-policy distillation, where the student is trained on data drawn from itself rather than from a static corpus. Agarwal et al. [2024] explore curriculum mixing of off- and on-policy losses for AR models; we revisit this in our ablations.

Distillation for DLMs. Research on distilling diffusion language models remains limited and has primarily focused on reducing the number of inference sampling steps. For instance, SDTT [Deschenaux and Gulcehre, 2025] aligns student and teacher log-probabilities to reduce sampling steps from 1024 to 64. DI4C [Hayakawa et al., 2025] models dimensional correlations to distill to one or a few steps. Our setting is orthogonal: we keep the sampling schedule and instead distill a much smaller student. This complements step distillation and could in principle be combined with it.

3 Two Failures of Forward-KL Distillation for DLMs

We use pretrained state-of-the-art DLMs as teachers and distill them into smaller students: an MDLM teacher (110M) and a Dream teacher (7B). For each teacher, we generate a static corpus by unconditional sampling and train a small student to minimize the forward KL between teacher and student output distributions on these samples. The MDLM student has 10.5M parameters and follows teacher's DiT architecture [Peebles and Xie, 2023]; the Dream student has 93M parameters and follows the Dream architecture (Table 8). To rule out distributional shift as a confounder, we use nested data subsets. The MDLM corpora satisfy $\mathcal{D}_{1\text{M}} \subset \mathcal{D}_{10\text{M}} \subset \mathcal{D}_{30\text{M}} \subset \mathcal{D}_{100\text{M}}$, and the Dream corpora satisfy $\mathcal{D}_{30\text{M}} \subset \mathcal{D}_{100\text{M}} \subset \mathcal{D}_{300\text{M}} \subset \mathcal{D}_{500\text{M}} \subset \mathcal{D}_{1000\text{M}}$, so any non-monotone scaling cannot be attributed to distributional shift across scales.

Finding 1: Non-monotone scaling For MDLM, FKL-distilled students attain GPT-2 perplexities at $\{1\text{M}, 10\text{M}, 30\text{M}, 100\text{M}\}$ shown in Figure 1: 10M is optimal, and additional data harms perplexity. The same non-monotonicity is observed when distilling Dream into a 93M student (Table 1): the 300M-token corpus is optimal at 25.23, while the 500M and 1000M corpora regress.

Finding 2: Objective mismatch Holding the corpus fixed, FKL distillation underperforms simply training the student with the teacher's own denoising objective on the same data. For MDLM at the optimal 10M corpus, FKL distillation yields 122.37 versus 90.20 for the original MDLM objective (Table 2). The same pattern holds for Dream at the optimal 300M corpus: FKL distillation reaches 25.23 while original Dream denoising objective on the same teacher-generated tokens reaches 20.71.

This diagnosis is not merely empirical: we later show in Proposition 4.1 that a student can attain zero FKL at every forward-visited state yet have total-variation distance arbitrarily close to 1 from the data, because its reverse process visits states the forward process never reaches. This motivates the method in Section 4: we replace off-policy supervision on forward-noised teacher samples with

Table 1: Dream-7B FKL distillation to a 93M student exhibits non-monotone corpus scaling. Lower is better.

Corpus size	30M	100M	300M	500M	1000M
Qwen-2.5 PPL ↓	46.58	33.54	**25.23**	44.48	37.68

Table 2: Objective mismatch for MDLM at the 10M-token corpus. Lower is better.

Method	GPT-2 PPL ↓
FKL distillation on real OWT	138.84
FKL distillation on teacher samples	122.37
MDLM objective on teacher samples	**90.20**

on-policy supervision along the student's own trajectory, weighting the teacher signal by confidence and stabilizing updates with a trust region.

4 Method

4.1 Two-phase pipeline

Let $q_t(\cdot \mid x_0)$ denote the masked-diffusion forward kernel that, given a clean sequence x_0 and time $t \in [0, T]$, produces a partially masked sequence x_t. Let $p_T(\cdot \mid x_t, t)$ and $p_\theta(\cdot \mid x_t, t)$ denote the teacher's and student's per-position output distributions at time t given x_t, and let $D(\cdot \,\|\, \cdot)$ be a divergence; in our experiments D is FKL for Dream and α-β divergence for MDLM.

We use a two-phase pipeline. The first phase, *off-policy distillation*, draws a clean sequence x_0 from a static teacher-generated corpus $\mathcal{D}_{\text{off}}$, samples a time t and a noised state $x_t \sim q_t(\cdot \mid x_0)$, and matches the teacher and student distributions at x_t:

$$\mathcal{L}_{\text{off}}(\theta) \;=\; \mathbb{E}_{x_0 \sim \mathcal{D}_{\text{off}},\; t \sim \mathcal{U}(0,T),\; x_t \sim q_t(\cdot|x_0)} \big[\, D(p_T(\cdot \mid x_t, t) \,\|\, p_\theta(\cdot \mid x_t, t)) \big]\,. \tag{1}$$

We use a static corpus rather than on-the-fly teacher generation because (i) teacher generation is the expensive step, and decoupling it from training improves throughput, and (ii) a static corpus extracts more signal per generated sample than discarding it after a single optimizer step.

We empirically observe that prolonged off-policy training overfits: the student's negative log-likelihood on OWT (the teacher's training data) keeps decreasing while its perplexity on generic text begins to rise. Figure 2 shows this behavior for MDLM with α-β divergence: after the 5k-step checkpoint, validation NLL continues to decrease while GPT-2 generative perplexity rises. This is the symptom of the off-policy/on-policy mismatch diagnosed in Section 3. Once this divergence appears, we switch to the *on-policy* phase, initialized from the off-policy checkpoint.

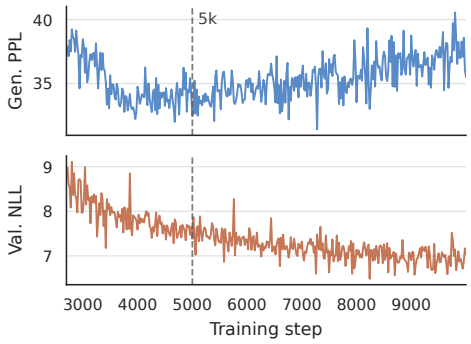

Figure 2: **Off-policy α-β divergence overfitting dynamics in MDLM.** Validation NLL on OWT keeps decreasing, but GPT-2 generative perplexity rises after the 5k-step checkpoint.

The on-policy loss combines confidence-weighted trajectory matching with a trust-region anchor:

$$\boxed{\mathcal{L}_{\text{on}}(\theta) \;=\; \mathcal{L}^{\text{CW}}_{\text{traj}}(\theta) \;+\; \lambda\, \mathbb{E}_{x_t \sim \rho^t_\theta} \big[\, \mathrm{KL}(p_{\theta_{\text{old}}}(\cdot \mid x_t, t) \,\|\, p_\theta(\cdot \mid x_t, t)) \big]} \tag{2}$$

where $\mathcal{L}^{\text{CW}}_{\text{traj}}$ is the confidence-weighted trajectory loss defined in Sections 4.2 and 4.3, and the second term is trust-region anchor of Section 4.4. The full pipeline is shown in Figure 3 and Algorithm 1.

4.2 Trajectory loss along the student's reverse process

On-policy training in our setting works as follows: the student rolls out a partial reverse trajectory with the same number of diffusion steps used at inference; from this trajectory, we sample K intermediate states $x_{t_1}, \ldots, x_{t_K}$ by drawing the timesteps $t_1, \ldots, t_K$ uniformly at random; both teacher and

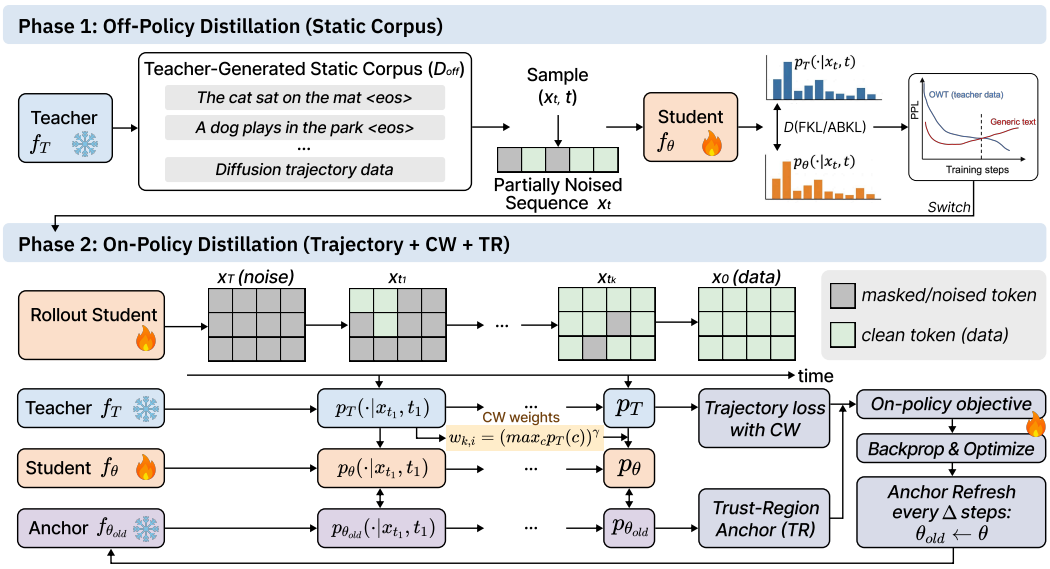

Figure 3: **Two-phase distillation pipeline.** Phase 1 (off-policy): the student is trained on a static teacher-generated corpus with divergence D. When held-out perplexity stops improving, training switches to Phase 2 (on-policy): the student rolls out partial reverse trajectories, intermediate states x_{t_k} are scored by both teacher and student under a trajectory loss, the loss is reweighted by teacher confidence (CW), and a trust-region KL anchor (TR) against a periodically refreshed copy θ_{old} stabilizes optimization.

student are queried on each x_{t_k}. The trajectory loss is

$$\mathcal{L}_{\text{traj}}(\theta) \;=\; \mathbb{E}_{x_0\sim\rho_\theta,\ \{t_k\}\sim\mathcal{U}\{1..T\}^{\uparrow}}\left[\frac{1}{K}\sum_{k=1}^{K} D(p_T(\cdot \mid x_{t_k}, t_k)\,\|\,p_\theta(\cdot \mid x_{t_k}, t_k))\right]. \tag{3}$$

The trajectory states $\{x_{t_k}\}$ are treated as stop-gradient: they are produced inside a `torch.no_grad()` block during rollout, and only the divergence is differentiated.

Why trajectory matters. A single-timestep loss provides only local supervision and gives no signal at the intermediate states that the student actually visits during sampling. Without trajectory supervision, the student can develop self-reinforcing errors: a mistake at one denoising step corrupts the input to the next, and a single random timestep is unlikely to surface that mistake. Supervising multiple intermediate states constrains the student to remain consistent with the teacher across the reverse process, reducing error accumulation.

Bias–variance trade-off in K. Empirically, $K = 4$ performs best: $K = 1$ gives a noisy estimate of the trajectory loss, whereas $K \geq 8$ tends to over-constrain the update (Table 6). The reason is that increasing K improves trajectory coverage but also introduces many correlated training signals from the same reverse trajectory. Writing the gradient variance as $\mathrm{Var}[\nabla\mathcal{L}_{\text{traj}}] = \frac{1}{K^2}\sum_{i,j}\mathrm{Cov}\big[\nabla\ell_{t_i}, \nabla\ell_{t_j}\big]$. There are only K diagonal variance terms but $K(K-1)$ off-diagonal covariance terms. Since all K intermediate timesteps are sampled from the same student trajectory, their gradients are positively correlated; therefore, the off-diagonal terms do not vanish. Consequently, the variance saturates near the average pairwise covariance rather than decreasing at the ideal $1/K$ Monte Carlo rate.

4.3 Confidence reweighting

On-policy rollouts can visit states x_t that lie far from the teacher's training-time states. At such states, the teacher distribution p_T is uncertain, so enforcing the divergence uniformly can inject high-variance and potentially biased supervision. We therefore use token-level confidence reweighting:

$$\mathcal{L}^{\text{CW}}_{\text{traj}}(\theta) \;=\; \mathbb{E}\left[\frac{1}{K}\sum_{k=1}^{K}\sum_{i} w_{k,i}\, D(p_{T,k,i}\,\|\,p_{\theta,k,i})\right], \quad w_{k,i} \;=\; \big(\max_c p_{T,k,i}(c)\big)^{\gamma}, \tag{4}$$

Algorithm 1 Two-phase data-free distillation for masked DLMs.

Require: teacher f_T, student f_θ, static teacher corpus $\mathcal{D}_{\text{off}}$, divergence D, hyperparameters $K, \gamma, \lambda, \Delta$, switch criterion $\mathcal{C}$.

Off-policy phase. Train θ on $\mathcal{D}_{\text{off}}$ by minimizing $\mathcal{L}_{\text{off}}(\theta)$ in Equation (1) until $\mathcal{C}$ indicates that held-out evaluator perplexity no longer improves.
Initialize $\theta_{\text{old}} \leftarrow \theta$ and refresh counter $c \leftarrow 0$.
while on-policy training has not converged **do**
    Roll out the current student reverse process from noise; sample and sort K intermediate states $\{(x_{t_k}, t_k)\}_{k=1}^{K}$. ▷ stop-gradient on states
    Query the teacher and student to obtain $p_T(\cdot \mid x_{t_k}, t_k)$ and $p_\theta(\cdot \mid x_{t_k}, t_k)$ for all k.
    Set $w_{k,i} \leftarrow (\max_c p_{T,k,i}(c))^\gamma$ and compute $\mathcal{L}_{\text{traj}}^{\text{CW}} \leftarrow \frac{1}{K}\sum_k \sum_i w_{k,i}\, D(p_{T,k,i} \| p_{\theta,k,i})$.
    Query the anchor model $p_{\theta_{\text{old}}}(\cdot \mid x_{t_k}, t_k)$ and compute $\mathcal{L}_{\text{TR}} \leftarrow \frac{1}{K}\sum_k \text{KL}(p_{\theta_{\text{old}},k} \| p_{\theta,k})$.
    Update θ using $\nabla_\theta(\mathcal{L}_{\text{traj}}^{\text{CW}} + \lambda \mathcal{L}_{\text{TR}})$.
    $c \leftarrow c + 1$; if $c \bmod \Delta = 0$, refresh $\theta_{\text{old}} \leftarrow \theta$.
end while

where i indexes token positions and $\gamma \geq 0$ controls the strength of reweighting. We find $\gamma = 1$ works best in our experiments. This weighting down-weights supervision at states where the teacher is uncertain. In particular, when the teacher distribution is close to uniform over the vocabulary, $\max_c p_T(c)$ is close to $1/C$, so $w_{k,i}$ becomes small. Confidence reweighting therefore reduces the influence of unreliable teacher targets while preserving stronger supervision at states where the teacher assigns high probability to a small set of tokens.

4.4 Trust-region anchor

Because on-policy training has a moving data distribution, the student can drift away from a working solution, especially when CW down-weights large fractions of a batch. We add a forward-KL anchor to a periodically refreshed copy θ_{old} of the student. The full on-policy objective is Equation (2), repeated here:

$$\mathcal{L}_{\text{on}}(\theta) \;=\; \mathcal{L}_{\text{traj}}^{\text{CW}}(\theta) \;+\; \lambda\, \mathbb{E}_{x_t \sim \rho_\theta^t}\left[\text{KL}(p_{\theta_{\text{old}}}(\cdot \mid x_t, t) \,\|\, p_\theta(\cdot \mid x_t, t))\right]. \tag{5}$$

The trust-region states are the same on-policy rollout states $x_t \sim \rho_\theta$ used by the trajectory loss; only the anchor distribution inside the KL is computed under the frozen θ_{old}. We use a constant λ and refresh $\theta_{\text{old}} \leftarrow \theta$ every Δ optimizer steps. We only tested FKL for the anchor: the goal here is a soft regularizer that keeps the student close to a recently known-good policy. Schedules for λ, alternative anchor divergences, and adaptive Δ are interesting directions we leave to future work.

4.5 Theoretical analysis

We next make the preceding intuition precise. The analysis shows how off-policy distillation incurs a mismatch between training and sampling states, and how on-policy supervision, confidence reweighting, and the trust-region anchor mitigate this mismatch. Full proofs are deferred to Section A.

Proposition 4.1 (Off-policy failure for masked DLMs)**.** *For any $\epsilon > 0$, there exists a data distribution over sequences of length n, a masked diffusion process with $T = 2$ steps, and a student that achieves zero off-policy divergence at every forward-visited state, yet whose generation quality satisfies* $\text{TV}(p_\theta^{\text{gen}}, p_{\text{data}}) > 1 - \epsilon$.

The proof in Section A.2 uses a two-token vocabulary where forward noising preserves token correlations, but the student's reverse process breaks these correlations by sampling positions independently, creating states outside forward support. This formalizes the non-monotone scaling of Section 3.

Unified bound for on-policy distillation. Our main result shows that on-policy trajectory loss with confidence reweighting yields a controlled bound on the generation quality gap; the trust-region anchor separately stabilizes the moving rollout distribution during optimization.

Theorem 4.2 (Calibrated divergence theorem)**.** *Let $\ell : \mathcal{V}^L \to [0, B]$ be any bounded evaluation function. Let D be a divergence satisfying the calibration condition*

$$\text{TV}(P, Q) \;\leq\; \kappa_D \sqrt{D(P \| Q)} \tag{6}$$

for all teacher/student one-step conditionals encountered under the student rollout distribution. Define the confidence weight $w_t(x_t) = \big(\max_c p_T(c \mid x_t, t)\big)^\gamma$*, and assume the* teacher-calibration condition*: there exists* $\varepsilon_0 \geq 0$ *such that*

$$\varepsilon_{\text{lowconf}} \;\coloneqq\; \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}_{x_t \sim q_\theta}\big[(1 - w_t(x_t))\, D_t(x_t)\big] \;\leq\; \varepsilon_0, \tag{7}$$

where $D_t(x_t) \coloneqq D(p_T(\cdot \mid x_t, t) \| p_\theta(\cdot \mid x_t, t))$*. Then for any student* θ*:*

$$\underbrace{\big|\mathbb{E}_{q_\theta}[\ell(x_0)] - \mathbb{E}_p[\ell(x_0)]\big|}_{\textit{generation gap}} \;\leq\; BT\kappa_D \sqrt{\underbrace{\mathcal{L}_{\text{traj}}^{\text{CW}}}_{\textit{(I) trajectory}} + \underbrace{\varepsilon_0}_{\textit{(II) low-confidence residual}}}, \tag{8}$$

where $\mathcal{L}_{\text{traj}}^{\text{CW}}$ *is the time-averaged CW trajectory loss under divergence* D*.*

Term (I) is minimized directly by the trajectory loss under the student's own states. Term (II) is the low-confidence residual: the divergence accumulated on positions the teacher itself flags as uncertain ($w_t \to 0$). Assumption (7) is a *teacher-strength condition*: a sufficiently capable teacher whose confidence is calibrated with its reliability rarely places non-trivial mass on uncertain positions in regions the student visits, so $(1-w_t)\, D_t$ is uniformly small and (7) holds with a small ε_0. Empirically this is the regime in which our bound is tight. The trust-region penalty $\mathcal{L}_{\text{TR}}$ does not appear in this bound; its role is to stabilize optimization by controlling the non-stationarity of the rollout distribution between refreshes (analogous to TRPO [Schulman et al., 2015]; see Section A.5).

For Dream, FKL satisfies Equation (6) with $\kappa_D = 1/\sqrt{2}$ by Pinsker's inequality. For MDLM, we use α-β divergence with $(\alpha, \beta) = (0.2, 0.5)$, hence $\alpha + \beta = 0.7 \leq 1$, so Lemma A.3 shows that the theorem applies to the α-β divergence trajectory loss with the same $\kappa_D = 1/\sqrt{2}$.

Our on-policy trajectory loss is an instance of the DAgger framework [Ross et al., 2011] adapted to diffusion: the teacher is the "expert" queried at the student's visited states. Off-policy distillation corresponds to behavioral cloning and therefore suffers $\mathcal{O}(T^2\epsilon)$ compounding error. In contrast, on-policy training reduces this to $\mathcal{O}(T\epsilon)$, improving the horizon dependence from quadratic to linear. The proof is given in Section A.3.

5 Experiments

5.1 Setup

We evaluate on two teachers: MDLM-110M [Sahoo et al., 2024] and Dream-7B [Ye et al., 2025]. Both students are smaller versions of their teachers' architectures: a 10.5M-parameter DiT [Peebles and Xie, 2023] for MDLM and a 93M-parameter Dream model for Dream-7B. Full architectures and optimizer hyperparameters are in Section B.

For both teachers we run a fully data-free pipeline: we generate samples unconditionally from the teacher, store them as a static corpus $\mathcal{D}_{\text{off}}$, and train the student on this corpus and on its own on-policy rollouts. MDLM samples have length 1024 (GPT-2 tokenizer); Dream samples have length 256 (Qwen-2.5 tokenizer). Off-policy uses batch size 512 (MDLM) and 1024 (Dream), with learning rates 3×10^{-4} and 1×10^{-4} respectively. On-policy is initialized from the converged off-policy checkpoint (5k steps for MDLM, 25k for Dream) and uses smaller batches and a much lower learning rate (batch sizes 16 and 32 at 3×10^{-8}, following Lu and Lab [2025]).

For MDLM we use α-β divergence with $(\alpha, \beta) = (0.2, 0.5)$ throughout; for Dream we use FKL. We adopt α-β divergence for MDLM because FKL is itself insufficient there: Table 2 shows FKL distillation reaching only 122.37 perplexity, whereas α-β divergence with $(\alpha, \beta) = (0.2, 0.5)$ reaches 32.48 at the same token budget, as shown in Table 5. For Dream, we use FKL because its 7B teacher is substantially stronger than the MDLM teacher. In this setting, the choice of divergence has a smaller effect on performance. We use the same (α, β) in both phases for MDLM, having found in experiments that mixing divergences across phases is worse. We use $K = 4$ trajectory samples in both. The anchor is refreshed every $\Delta = 100$ steps (MDLM) and $\Delta = 150$ steps (Dream).

We follow prior work [Sahoo et al., 2024, Deschenaux and Gulcehre, 2025, Hayakawa et al., 2025] and report perplexity under a stronger model with a matched tokenizer. For MDLM, we use GPT-2 Large [Radford et al., 2019]. For Dream, we use Qwen-2.5-7B [Yang et al., 2024], whose tokenizer matches Dream because Dream is initialized from Qwen-2.5.

Table 3: Perplexity of the distilled MDLM (10.5M, GPT-2 PPL) and Dream (93M, Qwen-2.5 PPL) students under successive ablations. Lower is better. Best value in each column is in **bold**.

Method	MDLM (GPT-2 PPL) ↓	Dream (Qwen-2.5 PPL) ↓
Off-policy distillation only	32.48	25.23
+ on-policy, trajectory $K=4$	25.38	24.96
+ on-policy, traj. + CW	27.44	20.88
+ on-policy, traj. + CW + TR	**24.75**	**19.83**
Teacher reference	27.04	12.07

5.2 Does teacher-generated data hurt?

Since FKL distillation underperforms the original denoising objective on the same teacher data (Section 3), one hypothesis is that the teacher-generated data is itself worse than real text. We test this directly on MDLM at the optimal 10M-token budget by training students with FKL on (i) teacher-generated tokens versus (ii) real OWT tokens, and additionally compare to the original MDLM denoising objective on the teacher tokens. Results are shown in Table 2: teacher-generated tokens outperform real OWT under FKL distillation. This is consistent with the FreeFlow observation [Tong et al., 2025] that real corpora constrain intermediate diffusion trajectories to the empirical data manifold, while a student should instead follow the teacher's generative dynamics. The remaining gap is therefore not a data problem but a state-distribution problem: FKL is evaluated only on forward-noised teacher samples, whereas at inference time the student must denoise states produced by its own reverse process. Closing this mismatch is precisely what our on-policy framework does.

5.3 Main results

Table 3 reports the GPT-2 / Qwen-2.5 perplexity of the distilled student under each ablation of our framework. Off-policy alone is already a strong starting point; switching to on-policy with our trajectory loss at $K=4$ improves further; adding confidence reweighting and the trust-region anchor gives the best result. On MDLM the final student reaches 24.75 perplexity, below the 27.04 of the 110M MDLM teacher. Similar improvements over the teacher have also been observed in continuous diffusion models: SiD [Zhou et al., 2024] reports a distilled student with better FID and Inception Score than its teacher. We hypothesize that the on-policy phase of our pipeline enables the MDLM gain by supervising the student at multiple states along its own reverse trajectory, aligning training more closely with inference than the teacher's pretraining on forward-noised data. On Dream, the final 93M student reaches 19.83, comfortably below the 25.23 obtained by FKL distillation alone.

Generation quality (MAUVE). Table 4 reports MAUVE scores [Pillutla et al., 2021] for Dream, measuring distributional similarity between student-generated and reference text. All on-policy variants substantially improve over the off-policy baseline of 0.60. Interestingly, the unconstrained trajectory variant with $K = 4$ achieves the highest MAUVE score of 0.76, while adding confidence weighting and the trust-region anchor slightly reduces MAUVE. This suggests a precision–diversity trade-off: CW and TR improve token-level accuracy by down-weighting uncertain teacher targets and stabilizing updates, but they also modestly reduce distributional coverage.

Table 4: MAUVE scores for Dream (93M) students. Higher is better.

Method	MAUVE ↑
Off-policy distillation only	0.6034
+ on-policy, trajectory $K=4$	**0.7643**
+ on-policy, traj. + CW	0.6787
+ on-policy, traj. + CW + TR	0.6677

5.4 Ablations

Choice of α and β for Off-Policy α-β divergence on MDLM. We sweep (α, β) for training MDLM and compare the perplexity once all configurations have converged. Although Wang et al. [2025] report that small α and large β are best for AR distillation, we observe a different regime here: small $\alpha=0.2$ paired with moderate $\beta=0.5$ is best (Table 5). We adopt this configuration as the off-policy initialization and reuse it during the on-policy phase.

Number of Trajectory Timesteps K for On-Policy MDLM. Table 6 sweeps $K \in \{1, 4, 6, 8\}$ and compares against reverse-KL on-policy distillation. $K=4$ is best; larger values over-constrain the

Table 5: Off-policy MDLM perplexity for different (α, β) in α-β divergence.

α	β	GPT-2 PPL ↓
0.2	0.2	65.75
0.2	0.5	**32.48**
0.2	0.7	59.98
0.2	0.9	150.91
0.5	0.5	88.75
0.7	0.2	89.99
0.7	0.7	530.22

Table 6: Effect of the number of trajectory timesteps K on MDLM on-policy distillation. Lower is better.

Method	GPT-2 PPL ↓
$K = 1$	27.64
$K = 4$	**25.38**
$K = 6$	25.80
$K = 8$	26.78
Reverse-KL on-policy	27.72
Teacher reference	27.04

student and degrade perplexity. Notably, trajectory-consistent α-β divergence at $K = 4$ surpasses the teacher's own perplexity, supporting the view that intermediate-state supervision provides a richer signal than final-output matching.

Curriculum on the on-policy weight does not transfer from AR. Agarwal et al. [2024] use a mixture of real data and student-generated samples for on-policy AR LLM distillation. We tried the same idea in our setting by gradually mixing on-policy data into off-policy training with a stepwise / linear / cosine schedule on the on-policy weight h. This results in model collapse for all three schedules. For Dream, Figure 4 shows the loss curves under the linear schedule. The off-policy loss initially decreases and then increases, while the on-policy loss exhibits abrupt rises and drops despite an overall downward trend. These trends suggest competing objectives and unstable optimization. We hypothesize that the conflict arises from their different state distributions: off-policy distillation matches the teacher on forward-noised teacher samples, whereas on-policy distillation matches it on states visited by the student's reverse process. Joint optimization may therefore destabilize the student and contribute to collapse. This forward–reverse mismatch does not exist in AR models. We therefore use a hard switch from off-policy to on-policy in the final framework. This shows that AR and DLM training dynamics differ in non-trivial ways, and that AR-derived recipes do not always carry over.

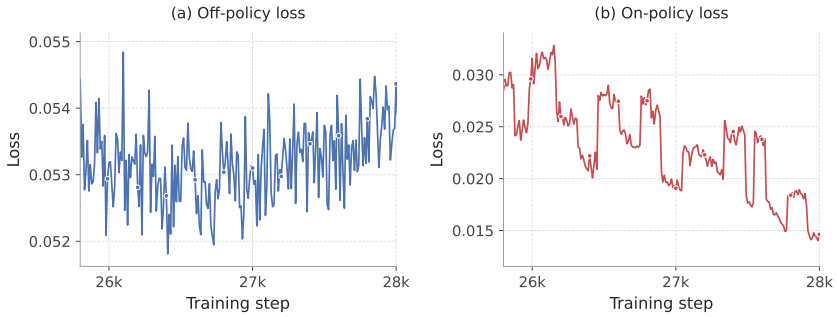

Figure 4: **Loss dynamics under linear curriculum mixing for Dream.** (a) The off-policy loss decreases initially and then rises. (b) The on-policy loss shows abrupt rises and drops despite an overall downward trend.

5.5 Zero-shot benchmarks

Table 7 reports zero-shot accuracy on HellaSwag (HS), PIQA, ARC-Easy (ARC-E), and Winogrande (WG) for the Dream student, starting from the optimal 300M-token off-policy checkpoint. On-policy trajectory training preserves the benchmark performance of the off-policy model and slightly improves several metrics, including PIQA and ARC-Easy. Adding confidence reweighting keeps HS, PIQA, and WG broadly comparable, although ARC-Easy decreases modestly. Overall, the zero-shot results remain close across ablations, while Table 3 shows substantial perplexity improvements. Thus, within the capacity of a smaller student, our framework transfers the teacher's generation behavior more effectively than forward-KL, improves generation quality, and preserves downstream performance.

Table 7: Zero-shot accuracy of the Dream-93M student on-policy ablations starting from the 300M off-policy checkpoint.

Setting	HS	PIQA	ARC-E	WG
off-policy ckpt	0.2742	0.5582	0.3481	0.4949
traj. $K=4$	**0.2751**	**0.5609**	**0.3514**	0.4941
traj. $K=4$ + CW	0.2723	0.5566	0.3308	**0.4957**
traj. $K=4$ + CW + TR	0.2719	0.5566	0.3316	0.4949

5.6 Qualitative samples

We provide qualitative MDLM samples in Section D, comparing the FKL-distilled student and our student. The FKL student exhibits severe degeneration, including repetition loops, malformed phrases, and weak discourse structure. Our student substantially mitigates these issues, producing longer stretches with a more stable topic and fewer catastrophic loops under the same sampling budget.

6 Conclusion

We presented the first systematic study of knowledge distillation for diffusion language models, identified two failure modes of classical forward-KL distillation, and proposed a unified data-free distillation framework that combines off- and on-policy phases, a trajectory loss along the student's own reverse process, confidence reweighting, and a trust-region anchor. The resulting students substantially improve over FKL baselines in both the small (MDLM-110M) and large (Dream-7B) regimes, with the MDLM student surpassing its teacher in perplexity and the Dream student preserving zero-shot accuracy at the smaller student scale. We hope this serves as a foundation for further work on scalable training and sampling for diffusion-based language models.

Acknowledgments

M. Rodrigues is supported by the Engineering and Physical Sciences Research Council (EPSRC) through the AI Hub in Generative Models [grant number EP/Y028805/1]. S. Yoon was supported in part by the InnoCORE program of the Ministry of Science and ICT (No. 1.260017.01), the Institute of Information & Communications Technology Planning & Evaluation (IITP) grants funded by the Korea government (MSIT) (RS-2025-25442824, AI Star Fellowship Program (Ulsan National Institute of Science and Technology)) and (No. RS-2020-II201336, Artificial Intelligence Graduate School Program (UNIST)), and the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. RS-2026-25506577).

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 *The twelfth international conference on learning representations*, 2024.

DeepSeek-AI. Deepseek-v4: Towards highly efficient million-token context intelligence, 2026.

Justin Deschenaux and Caglar Gulcehre. Beyond autoregression: Fast llms via self-distillation through time. In *The Thirteenth International Conference on Learning Representations*, 2025.

GL Gilardoni. On pinsker's type inequalities and csiszár's f-divergence. *Part I: Second and Fourth-order inequalities (Preprint arXiv: cs/0603097v2)*, 2008.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. Minillm: Knowledge distillation of large language models. In *The Twelfth International Conference on Learning Representations*, 2024.

Satoshi Hayakawa, Yuhta Takida, Masaaki Imaizumi, Hiromi Wakaki, and Yuki Mitsufuji. Distillation of discrete diffusion through dimensional correlations. In *Forty-second International Conference on Machine Learning*, 2025.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Adam Tauman Kalai, Ofir Nachum, Santosh S Vempala, and Edwin Zhang. Why language models hallucinate. *arXiv preprint arXiv:2509.04664*, 2025.

Jongwoo Ko, Sungnyun Kim, Tianyi Chen, and Se-Young Yun. Distillm: Towards streamlined distillation for large language models. In *International Conference on Machine Learning*, pages 24872–24895. PMLR, 2024.

Inception Labs, Samar Khanna, Siddhant Kharbanda, Shufan Li, Harshit Varma, Eric Wang, Sawyer Birnbaum, Ziyang Luo, Yanis Miraoui, Akash Palrecha, et al. Mercury: Ultra-fast language models based on diffusion. *arXiv preprint arXiv:2506.17298*, 2025.

Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion modeling by estimating the ratios of the data distribution. In *Proceedings of the 41st International Conference on Machine Learning*, pages 32819–32848, 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.

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*, 2025.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4195–4205, 2023.

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

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

Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 627–635. JMLR Workshop and Conference Proceedings, 2011.

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

Subham Sekhar Sahoo, Justin Deschenaux, Aaron Gokaslan, Guanghan Wang, Justin Chiu, and Volodymyr Kuleshov. The diffusion duality. *arXiv preprint arXiv:2506.10892*, 2025.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International conference on machine learning*, pages 1889–1897. PMLR, 2015.

Yuxuan Song, Zheng Zhang, Cheng Luo, Pengyang Gao, Fan Xia, Hao Luo, Zheng Li, Yuehang Yang, Hongli Yu, Xingwei Qu, et al. Seed diffusion: A large-scale diffusion language model with high-speed inference. *arXiv preprint arXiv:2508.02193*, 2025.

Shangyuan Tong, Nanye Ma, Saining Xie, and Tommi Jaakkola. Flow map distillation without data. *arXiv preprint arXiv:2511.19428*, 2025.

Guanghui Wang, Zhiyong Yang, Zitai Wang, Shi Wang, Qianqian Xu, and Qingming Huang. Abkd: Pursuing a proper allocation of the probability mass in knowledge distillation via $alpha$-$beta$-divergence. In *Forty-second International Conference on Machine Learning*, 2025.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

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.

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.

Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Shiji Song, and Gao Huang. Does reinforcement learning really incentivize reasoning capacity in llms beyond the base model? *Advances in Neural Information Processing Systems*, 2025.

Mingyuan Zhou, Huangjie Zheng, Zhendong Wang, Mingzhang Yin, and Hai Huang. Score identity distillation: Exponentially fast distillation of pretrained diffusion models for one-step generation. In *Forty-first International Conference on Machine Learning*, 2024.

A Extended Theoretical Results

A.1 Telescoping error bound

Theorem A.1 (On-Policy Error Decomposition)**.** *Let p and q_θ denote the joint distributions over reverse trajectories induced by the teacher and student, respectively. Both chains start from the fully masked state $x_T \sim \mu$. Then*

$$\mathrm{TV}\big(q_\theta(x_0),\ p(x_0)\big)\ \le\ \sum_{t=1}^{T} \mathbb{E}_{x_t\sim q_\theta(\cdot,t)}\Big[\mathrm{TV}\big(p_\theta(\cdot\mid x_t,t),\ p_T(\cdot\mid x_t,t)\big)\Big], \tag{9}$$

where $q_\theta(\cdot,t)$ is the marginal under the student's reverse process.

Proof. Define the hybrid chain $q^{(s)}$ that uses the student kernel for steps $T,\dots,s+1$ and the teacher kernel for steps $s,\dots,1$. Then $q^{(0)}=q_\theta$ and $q^{(T)}=p$. By the triangle inequality on marginals at x_0:

$$\mathrm{TV}\big(q_\theta(x_0),\ p(x_0)\big)\ \le\ \sum_{s=1}^{T} \mathrm{TV}\big(q^{(s-1)}(x_0),\ q^{(s)}(x_0)\big). \tag{10}$$

The chains $q^{(s-1)}$ and $q^{(s)}$ differ only at step s and share the upstream marginal $q_\theta(\cdot,s)$. By the data processing inequality (post-processing by the shared teacher kernels for steps $s-1,\dots,1$ cannot increase TV):

$$\mathrm{TV}\big(q^{(s-1)}(x_0),\ q^{(s)}(x_0)\big)\ \le\ \mathbb{E}_{x_s\sim q_\theta(\cdot,s)}\Big[\mathrm{TV}\big(p_\theta(\cdot\mid x_s,s),\ p_T(\cdot\mid x_s,s)\big)\Big]. \tag{11}$$

Summing over s yields the result. □

Corollary A.2 (Trajectory Loss as Unbiased Estimator)**.** *Let $x_{0:T}\sim q_\theta$ be a student reverse trajectory, and let $t_1,\dots,t_K$ be drawn i.i.d. uniformly from $\{1,\dots,T\}$, independently of the trajectory (sorting the t_k does not affect the result). Define $D_t(x_t) \coloneqq D(p_T(\cdot\mid x_t,t)\,\|\,p_\theta(\cdot\mid x_t,t))$ and assume $\mathbb{E}_{x_t\sim q_\theta}[D_t]<\infty$ for every t. Then*

$$\mathbb{E}[\mathcal{L}_{\mathrm{traj}}(\theta)]\ =\ \frac{1}{T}\sum_{t=1}^{T}\mathbb{E}_{x_t\sim q_\theta(\cdot,t)}[D_t(x_t)]\ =:\ \overline{D}. \tag{12}$$

Consequently, when $D=\mathrm{KL}$ and p_θ has full support, combining Theorem A.1 with Pinsker's inequality and the Cauchy–Schwarz inequality yields

$$\mathrm{TV}\big(q_\theta(x_0),\,p(x_0)\big)\ \le\ \sum_{t=1}^{T}\mathbb{E}_{x_t\sim q_\theta}\left[\sqrt{\tfrac{1}{2}D_t}\right]\ \le\ \sqrt{T}\,\sqrt{\tfrac{1}{2}\textstyle\sum_t \mathbb{E}[D_t]}\ =\ T\sqrt{\tfrac{1}{2}\,\overline{D}}. \tag{13}$$

Hence minimizing $\mathcal{L}_{\mathrm{traj}}$ in expectation minimizes an upper bound on $\mathrm{TV}(q_\theta(x_0),p(x_0))$, with explicit constant $T\sqrt{\overline{D}/2}$.

Proof. For each k, by independence of t_k from the trajectory and uniformity of t_k on $\{1,\dots,T\}$,

$$\mathbb{E}[D_{t_k}(x_{t_k})]\ =\ \frac{1}{T}\sum_{t=1}^{T}\mathbb{E}_{x_t\sim q_\theta(\cdot,t)}[D_t(x_t)]\ =\ \overline{D}. \tag{14}$$

Linearity of expectation over $k=1,\dots,K$ then gives $\mathbb{E}[\mathcal{L}_{\mathrm{traj}}]=\overline{D}$, proving the first claim. For the TV bound, Theorem A.1 gives $\mathrm{TV}(q_\theta(x_0),p(x_0))\le\sum_t\mathbb{E}_{x_t\sim q_\theta}[\mathrm{TV}_t]$. Pinsker's inequality applied pointwise yields $\mathrm{TV}_t\le\sqrt{D_t/2}$, and Jensen's inequality on the concave $\sqrt{\cdot}$ gives $\mathbb{E}[\sqrt{D_t/2}]\le\sqrt{\mathbb{E}[D_t]/2}$. Finally, Cauchy–Schwarz on $\sum_t a_t\le\sqrt{T}\sqrt{\sum_t a_t^2}$ with $a_t=\sqrt{\mathbb{E}[D_t]/2}$ produces the factor T. □

A.2 Proof of Proposition 4.1 (Off-policy failure)

Proof. Fix vocabulary $\mathcal{V} = \{a, b\}$, sequence length n, and $T = 2$. Let $p_{\text{data}} = \frac{1}{2}\delta_{(a,\dots,a)} + \frac{1}{2}\delta_{(b,\dots,b)}$. The forward process masks each token independently with probability $\frac{1}{2}$ at $t = 1$ and fully masks at $t = 2$.

Zero off-policy loss. Recall the data distribution $p_{\text{data}} = \frac{1}{2}\delta_{(a,\dots,a)} + \frac{1}{2}\delta_{(b,\dots,b)}$, so any clean sample x_0 is *monochromatic*: either all-a or all-b. Off-policy training visits states x_t obtained by forward-noising such an x_0, i.e. $x_t \sim q(\cdot \mid x_0)$ with $x_0 \sim p_{\text{data}}$.

(1) What states are visited at $t = 1$. At $t = 1$ each position is independently masked with probability $\frac{1}{2}$, so a typical visited state x_1 has some subset $U \subseteq \{1, \dots, n\}$ of unmasked positions and the remaining positions equal to $[\texttt{MASK}]$. Crucially, because x_0 is monochromatic, *every* unmasked position in x_1 shows the *same* symbol $s \in \{a, b\}$. Mixed states such as $(a, b, \texttt{M}, a, \dots)$ are **never** visited by the forward process—they have probability zero under $q(x_1)$. This is the off-policy support that the student is trained on.

(2) The true per-position posterior on this support. Fix any visited x_1 with at least one unmasked position $j \in U$, all showing symbol s. For any masked position $i \notin U$, Bayes' rule gives

$$p(x_0^i = c \mid x_1) \;=\; \frac{\sum_{x_0} \mathbf{1}[x_0^i = c]\, p_{\text{data}}(x_0)\, q(x_1 \mid x_0)}{\sum_{x_0} p_{\text{data}}(x_0)\, q(x_1 \mid x_0)}.$$

Only $x_0 = (s, \dots, s)$ is consistent with x_1 (the other monochromatic candidate $(\bar{s}, \dots, \bar{s})$ is ruled out by the unmasked j), so the posterior collapses to the point mass $p(x_0^i = c \mid x_1) = \delta_{c=s}$. At $t = 2$ the entire sequence is masked, no unmasked position constrains x_0, and the posterior is the uniform mixture $\frac{1}{2}\delta_a + \frac{1}{2}\delta_b$ at every position.

(3) The student that achieves zero off-policy loss. Define a per-position student

$$p_\phi(x_0^i = c \mid x_t, t) \;=\; \begin{cases} \delta_{c=s(x_t)}, & t = 1 \text{ and } x_t \text{ has any unmasked position showing } s(x_t), \\ \frac{1}{2}\delta_a + \frac{1}{2}\delta_b, & t = 2, \\ \text{arbitrary}, & \text{on states outside the forward support.} \end{cases}$$

On every state in $\operatorname{supp} q$, p_ϕ matches the true posterior *exactly*, so the off-policy distillation loss $\mathbb{E}_{t,\, x_t \sim q}\big[D(p_T(\cdot \mid x_t, t) \,\|\, p_\phi(\cdot \mid x_t, t))\big]$ is identically zero, regardless of the divergence D. The behavior of p_ϕ on *off-support* states (e.g. mixed-symbol x_1) is unconstrained by the off-policy objective—and this is precisely the freedom that the next paragraph exploits to make generation fail.

Generation fails. We now show that the very same student p_ϕ that achieves zero off-policy loss generates valid monochromatic sequences with vanishing probability.

(1) The student's reverse process. Sampling starts from $x_2 = (\texttt{M}, \dots, \texttt{M})$, the fully masked state. At $t = 2$ the student's per-position posterior is $\frac{1}{2}\delta_a + \frac{1}{2}\delta_b$. We adopt the standard masked-DLM denoising step: at each reverse step a fresh subset of masked positions is unmasked by sampling *independently* from the student's per-position posterior at those positions. Per-position independence at a single denoising step is the assumption made by virtually all masked-DLM samplers [Sahoo et al., 2024, Nie et al., 2025, Ye et al., 2025] and is exactly the assumption Hayakawa et al. [2025] identify as the source of difficulty in few-step distillation.

In our $T = 2$ construction, after the first reverse step the chain transitions $x_2 \to x_1$, where each position is, with probability $\frac{1}{2}$, unmasked to an independent uniform draw from $\{a, b\}$ (and otherwise stays masked). The second step $x_1 \to x_0$ then fills the remaining masked positions according to $p_\phi(\cdot \mid x_1, 1)$.

(2) Probability of landing in the forward support after step 1. Let $U \subseteq \{1, \dots, n\}$ be the set of positions unmasked at step 1, and let $k = |U|$. Conditioned on U, each unmasked position is an i.i.d. uniform draw from $\{a, b\}$, so the probability that all k unmasked positions show the same symbol is

$$\Pr[x_1 \text{ monochromatic on } U \mid |U| = k] \;=\; \begin{cases} 1, & k \le 1, \\ 2^{1-k}, & k \ge 1. \end{cases}$$

Marginalizing over $|U| \sim \mathrm{Binomial}(n, \frac{1}{2})$,

$$\begin{aligned}\Pr[x_1 \text{ monochromatic on } U] \;&=\; \sum_{k=0}^{n} \binom{n}{k} \left(\tfrac{1}{2}\right)^n \cdot 2^{1-\max(k,1)} \\ &\le\; \sum_{k=0}^{n} \binom{n}{k} \left(\tfrac{1}{2}\right)^n \cdot 2 \cdot 2^{-k} \;=\; 2\left(\tfrac{3}{4}\right)^n,\end{aligned}$$

where the last equality uses the binomial identity $\sum_k \binom{n}{k}(1/2)^n 2^{-k} = (1/2 + 1/4)^n = (3/4)^n$. Equivalently: for the student to even *enter* the forward support at step 1, all $|U|$ independent fair-coin draws must agree, an event of probability exponentially small in n.

(3) Why this implies generation fails. Conditioned on x_1 being monochromatic on U (showing some $s \in \{a, b\}$), the student's step-1 posterior $p_\phi(\cdot \mid x_1, 1) = \delta_s$ correctly fills the remaining masked positions with s, producing a valid monochromatic x_0. Conditioned on x_1 being mixed-symbol (off the forward support), the student's behavior is unconstrained by the off-policy objective. Hence

$$\Pr_{x_0 \sim p_\phi^{\mathrm{gen}}}\!\left[x_0 \in \{(a, \ldots, a), (b, \ldots, b)\}\right] \;\le\; \Pr[x_1 \text{ monochromatic on } U] \;\le\; 2\left(\tfrac{3}{4}\right)^n.$$

(4) From mass on the support to TV. Since p_{data} places all of its mass on the two monochromatic sequences and p_ϕ^{gen} places at most $2(3/4)^n$ there,

$$\mathrm{TV}(p_\phi^{\mathrm{gen}}, p_{\mathrm{data}}) \;\ge\; p_{\mathrm{data}}\big(\{(a, \ldots, a), (b, \ldots, b)\}\big) - p_\phi^{\mathrm{gen}}\big(\{(a, \ldots, a), (b, \ldots, b)\}\big) \;\ge\; 1 - 2\left(\tfrac{3}{4}\right)^n.$$

For any $\epsilon > 0$, choosing $n \ge \log(2/\epsilon)/\log(4/3)$ makes $2(3/4)^n \le \epsilon$, so $\mathrm{TV}(p_\phi^{\mathrm{gen}}, p_{\mathrm{data}}) > 1 - \epsilon$, as claimed.

□

Lemma A.3 (Pinsker-type control for α-β divergence)**.** *Let P and Q be probability distributions on a finite alphabet. For $\alpha > 0$, $\beta > 0$, and $s := \alpha + \beta \le 1$, the α-β divergence satisfies*

$$\mathrm{TV}(P, Q) \;\le\; \sqrt{\frac{1}{2}\,\mathrm{D}_{\alpha,\beta}(P\|Q)}. \tag{15}$$

Proof. Let $\lambda = \alpha/s \in (0, 1)$. Write

$$G_s(P, Q) \;=\; \sum_c \left[\lambda P(c)^s + (1-\lambda)Q(c)^s - P(c)^{s\lambda}Q(c)^{s(1-\lambda)}\right], \tag{16}$$

so that

$$\mathrm{D}_{\alpha,\beta}(P\|Q) \;=\; \frac{G_s(P, Q)}{s^2\lambda(1-\lambda)}. \tag{17}$$

For each coordinate with positive mass, set $u = \log P(c)$ and $v = \log Q(c)$; zero probabilities are handled by continuity. The coordinate summand in G_s is the Jensen gap of $\varphi_s(z) = e^{sz}$ between u and v with weights λ and $1 - \lambda$. Since $u, v \le 0$ and $s \le 1$,

$$\left(\varphi_s - s^2\varphi_1\right)''(z) \;=\; s^2\left(e^{sz} - e^z\right) \;\ge\; 0 \qquad \text{for all } z \le 0. \tag{18}$$

Thus the Jensen gap of φ_s is at least s^2 times the Jensen gap of φ_1, and therefore

$$G_s(P, Q) \;\ge\; s^2 G_1(P, Q). \tag{19}$$

It follows that

$$\mathrm{D}_{\alpha,\beta}(P\|Q) \;\ge\; \frac{G_1(P, Q)}{\lambda(1-\lambda)} \;=\; \frac{1 - \sum_c P(c)^\lambda Q(c)^{1-\lambda}}{\lambda(1-\lambda)}. \tag{20}$$

The right-hand side is the normalized alpha divergence generated by

$$f_\lambda(r) \;=\; \frac{\lambda r + (1-\lambda) - r^\lambda}{\lambda(1-\lambda)}, \qquad f_\lambda''(1) = 1. \tag{21}$$

By the standard Pinsker inequality for this f-divergence [Gilardoni, 2008, Theorem 3],

$$\frac{1 - \sum_c P(c)^\lambda Q(c)^{1-\lambda}}{\lambda(1-\lambda)} \;\ge\; 2\,\mathrm{TV}(P, Q)^2. \tag{22}$$

Combining the two inequalities gives Equation (15). □

A.3 Proof of Theorem 4.2 (Unified distillation bound)

Proof. The argument has three steps.

Step 1: Telescoping. Since $\ell \in [0, B]$, the variational representation of TV gives $\big|\mathbb{E}_{q_\theta}[\ell] - \mathbb{E}_p[\ell]\big| \le B \cdot \mathrm{TV}\big(q_\theta(x_0), p(x_0)\big)$. Applying Theorem A.1:

$$\big|\mathbb{E}_{q_\theta}[\ell] - \mathbb{E}_p[\ell]\big| \;\le\; B \cdot \mathrm{TV}\big(q_\theta(x_0), p(x_0)\big) \;\le\; B \sum_{t=1}^{T} \mathbb{E}_{x_t \sim q_\theta}[\mathrm{TV}(p_T \| p_\theta \mid x_t)].$$

Step 2: TV calibration + Jensen + Cauchy–Schwarz. We convert the sum of TV's into a single $\sqrt{\overline{D}}$ expression in three sub-steps. Write $D_t(x_t) \coloneqq D(p_T(\cdot|x_t, t) \| p_\theta(\cdot|x_t, t))$ and $a_t \coloneqq \mathbb{E}_{x_t \sim q_\theta}[D_t]$.

(a) By the calibration condition in Equation (6), for each fixed x_t, $\mathrm{TV}(p_T \| p_\theta \mid x_t) \le \kappa_D \sqrt{D_t(x_t)}$.

(b) Jensen on the inner expectation. Since $\sqrt{\cdot}$ is concave, $\mathbb{E}_{x_t}[\kappa_D \sqrt{D_t}] \le \kappa_D \sqrt{\mathbb{E}_{x_t}[D_t]} = \kappa_D \sqrt{a_t}$.

(c) Cauchy–Schwarz on the sum over t. With $u_t = 1$ and $v_t = \sqrt{a_t}$:

$$\sum_{t=1}^{T} \sqrt{a_t} \;=\; \sum_t u_t v_t \;\le\; \sqrt{\textstyle\sum_t u_t^2} \sqrt{\textstyle\sum_t v_t^2} \;=\; \sqrt{T} \sqrt{\textstyle\sum_t a_t} \;=\; T\sqrt{\overline{D}},$$

where $\overline{D} \coloneqq \frac{1}{T} \sum_t a_t$ is the time-averaged divergence. Combining (a)–(c):

$$B \sum_t \mathbb{E}_{x_t \sim q_\theta}[\mathrm{TV}_t] \;\le\; B\kappa_D \sum_t \sqrt{a_t} \;\le\; BT\kappa_D \sqrt{\overline{D}}.$$

Step 3: CW splitting. The trajectory loss is evaluated under the current rollout ρ_θ (Algorithm 1, Line 1), so $\overline{D} = \frac{1}{T} \sum_t \mathbb{E}_{x_t \sim q_\theta}[D_t]$ is exactly the (unweighted) trajectory loss in expectation. Decompose pointwise via the confidence weight: $D_t = w_t\, D_t + (1 - w_t)\, D_t$. Taking expectations and averaging over t:

$$\overline{D} \;=\; \mathcal{L}_{\mathrm{traj}}^{\mathrm{CW}} \;+\; \varepsilon_{\mathrm{lowconf}},$$

where $\mathcal{L}_{\mathrm{traj}}^{\mathrm{CW}} \coloneqq \frac{1}{T} \sum_t \mathbb{E}_{x_t \sim q_\theta}[w_t\, D_t]$ and $\varepsilon_{\mathrm{lowconf}} \coloneqq \frac{1}{T} \sum_t \mathbb{E}_{x_t \sim q_\theta}[(1 - w_t)\, D_t]$. Invoking the teacher-calibration assumption (7), $\varepsilon_{\mathrm{lowconf}} \le \varepsilon_0$, so $\overline{D} \le \mathcal{L}_{\mathrm{traj}}^{\mathrm{CW}} + \varepsilon_0$. Combining with Step 2 yields

$$\big|\mathbb{E}_{q_\theta}[\ell] - \mathbb{E}_p[\ell]\big| \;\le\; BT\kappa_D \sqrt{\mathcal{L}_{\mathrm{traj}}^{\mathrm{CW}} + \varepsilon_0}. \tag{23}$$

Remark on the trust region. The trust-region penalty $\mathcal{L}_{\mathrm{TR}}$ does *not* appear in this bound: it is an optimization stabilizer that keeps θ close to a recently known good policy θ_{old} between refreshes, controlling the non-stationarity of ρ_θ during training rather than tightening the generation-quality bound. Its role is analogous to the trust region in TRPO [Schulman et al., 2015]: it bounds how much the rollout distribution can shift between consecutive optimizer steps, ensuring the per-step gradient estimate remains a faithful proxy for the true on-policy gradient (Section A.5). □

A.4 CW gradient variance reduction

Proposition A.4 (CW Variance Reduction)**.** *Let g_i be per-position gradient vectors with confidence weight $w_i = (\max_c p_T(c \mid x_t, t, i))^\gamma$. Partition positions into $\mathcal{S}_{\mathrm{on}} = \{i : \max_c p_T \ge \tau\}$ and $\mathcal{S}_{\mathrm{off}} = \mathcal{I} \setminus \mathcal{S}_{\mathrm{on}}$. Under conditional uncorrelatedness of per-position gradients, let α be the fraction of total gradient variance from off-manifold positions. Then:*

$$\frac{\mathrm{tr}\,\mathrm{Var}[G^{\mathrm{CW}}]}{\mathrm{tr}\,\mathrm{Var}[G]} \;\le\; 1 - \alpha(1 - \tau^{2\gamma}). \tag{24}$$

When off-manifold variance dominates ($\alpha \to 1$), the ratio approaches $\tau^{2\gamma}$, vanishing exponentially in γ.

Proof. Under conditional uncorrelatedness: $\mathrm{Var}[G^{\mathrm{CW}}] = n^{-2} \sum_i w_i^2 \mathrm{Var}[g_i]$. For $i \in \mathcal{S}_{\mathrm{on}}$: $w_i^2 \le 1$. For $i \in \mathcal{S}_{\mathrm{off}}$: $w_i^2 < \tau^{2\gamma}$. Dividing by $\mathrm{Var}[G] = n^{-2} \sum_i \mathrm{Var}[g_i]$ gives $\frac{(1-\alpha)+\tau^{2\gamma}\alpha}{1} = 1 - \alpha(1 - \tau^{2\gamma})$. □

A.5 Trust-region anchor bounds distribution shift

Proposition A.5. *If the trust-region constraint $\mathbb{E}_{t,x_t\sim\rho_{\theta'}}[\mathrm{KL}(p_{\theta_{\mathrm{old}}}\|p_{\theta'})] \le \varepsilon$ holds and the per-step loss is bounded by B_ℓ, then:*

$$\left|\mathcal{L}_{\mathrm{traj}}(\theta') - \mathcal{L}_{\mathrm{traj}}^{\rho_{\theta_{\mathrm{old}}}}(\theta')\right| \;\le\; B_\ell \cdot T\sqrt{\varepsilon/2}, \tag{25}$$

where $\mathcal{L}_{\mathrm{traj}}^{\rho_{\theta_{\mathrm{old}}}}$ is the loss under the old rollout distribution.

Proof. The argument adapts the TRPO performance-difference bound [Schulman et al., 2015, Theorem 1] to the finite-horizon masked-DLM setting without discount factor.

Step 1 (Expectation gap by TV). Write $\ell_t(x_t) \in [0, B_\ell]$ for the per-step loss at time t. The variational representation of TV gives, for each t,

$$\left|\mathbb{E}_{x_t\sim\rho_{\theta'}}[\ell_t] - \mathbb{E}_{x_t\sim\rho_{\theta_{\mathrm{old}}}}[\ell_t]\right| \;\le\; B_\ell \cdot \mathrm{TV}\big(\rho_{\theta'}(\cdot,t),\, \rho_{\theta_{\mathrm{old}}}(\cdot,t)\big).$$

Step 2 (State-marginal TV via second telescoping). The two rollouts share the initial state x_T and differ only in their per-step kernels $p_{\theta'}$ vs. $p_{\theta_{\mathrm{old}}}$. By the same telescoping argument as Theorem A.1,

$$\mathrm{TV}\big(\rho_{\theta'}(\cdot,t),\, \rho_{\theta_{\mathrm{old}}}(\cdot,t)\big) \;\le\; \sum_{s=t+1}^{T} \mathbb{E}_{x_s\sim\rho_{\theta'}(\cdot,s)}\big[\mathrm{TV}(p_{\theta_{\mathrm{old}}}\|p_{\theta'} \mid x_s)\big].$$

Step 3 (Pinsker on per-step TV). For each s, $\mathrm{TV}(p_{\theta_{\mathrm{old}}}\|p_{\theta'} \mid x_s) \le \sqrt{D(p_{\theta_{\mathrm{old}}}\|p_{\theta'} \mid x_s)/2}$. Combined with Jensen's inequality on the inner expectation and Cauchy–Schwarz on the sum over s:

$$\sum_{s=t+1}^{T} \mathbb{E}_{\rho_{\theta'}}\big[\sqrt{\mathrm{KL}_s^{\mathrm{TR}}/2}\big] \;\le\; \sum_{s=t+1}^{T} \sqrt{\mathbb{E}_{\rho_{\theta'}}[\mathrm{KL}_s^{\mathrm{TR}}]/2} \;\le\; \sqrt{T}\,\sqrt{\textstyle\sum_{s=1}^{T}\mathbb{E}_{\rho_{\theta'}}[\mathrm{KL}_s^{\mathrm{TR}}]/2} \;\le\; T\sqrt{\varepsilon/2},$$

using $\mathbb{E}_{\rho_{\theta'}}[\mathrm{KL}_s^{\mathrm{TR}}] \le \varepsilon$ uniformly in s from the constraint. So $\mathrm{TV}(\rho_{\theta'}(\cdot,t), \rho_{\theta_{\mathrm{old}}}(\cdot,t)) \le T\sqrt{\varepsilon/2}$ for each t.

Step 4 (Average over t). Combining Steps 1–3 and averaging over t:

$$\left|\mathcal{L}_{\mathrm{traj}}(\theta') - \mathcal{L}_{\mathrm{traj}}^{\rho_{\theta_{\mathrm{old}}}}(\theta')\right| \;\le\; B_\ell \cdot T\sqrt{\varepsilon/2}.$$

□

A.6 Optimal K: bias–variance tradeoff

Proposition A.6 (Gradient Variance Decomposition)**.** *Let σ^2 be the average per-timestep gradient variance and $\bar\rho$ the average pairwise gradient correlation across timesteps from the same trajectory. Then:*

$$\mathrm{tr}\big(\mathrm{Cov}[\nabla\mathcal{L}_K]\big) \;\approx\; \frac{(1-\bar\rho)\,\sigma^2}{K} + \bar\rho\,\sigma^2. \tag{26}$$

The first term is reducible by increasing K; the second is an irreducible floor from intra-trajectory correlation.

Corollary A.7 (Compute-optimal K under a fixed budget)**.** *Assume a total compute budget B_{comp}. Each independent trajectory costs C_{traj}, and querying/scoring each sampled timestep costs c. Thus the number of independent trajectories that can be used with K sampled timesteps is approximately*

$$N(K) = \frac{B_{\mathrm{comp}}}{C_{\mathrm{traj}} + cK}. \tag{27}$$

Under the variance model of Proposition A.6, the variance of the averaged gradient estimator scales as

$$\mathrm{Var}_{\mathrm{avg}}(K) \;\propto\; \frac{C_{\mathrm{traj}} + cK}{B_{\mathrm{comp}}}\left(\frac{(1-\bar\rho)\sigma^2}{K} + \bar\rho\sigma^2\right). \tag{28}$$

Treating K as continuous, the minimizer is

$$K^* = \sqrt{\frac{(1-\bar\rho)\,C_{\mathrm{traj}}}{\bar\rho\,c}}. \tag{29}$$

When $\bar{\rho}$ is large, correlated timesteps provide little additional variance reduction, so the compute-optimal K is small. This is consistent with our empirical finding that $K = 4$ is optimal (Table 6).

A.7 Relationship to single-timestep training (LLaDA)

In standard masked diffusion training (e.g., LLaDA [Nie et al., 2025]), the state distribution $q(x_t \mid x_0)$ is *fixed* (forward noising of real data) and does not depend on θ. Single-timestep sampling yields an unbiased gradient. In on-policy *distillation*, states come from $q_\theta(x_t)$, which shifts with θ. A single-timestep loss corrects one random step while errors at the other $T - 1$ steps compound—the feedback loop described in Theorem A.1. Multi-step trajectory loss controls K terms of the bound simultaneously, breaking the compounding loop.

B Implementation Details

B.1 Student architectures

Table 8 lists the student-model hyperparameters. Each student preserves the corresponding teacher's architecture family and tokenizer while scaling down depth and width to obtain a smaller model. Teachers retain their original architectures .

Table 8: Hyperparameters of the MDLM and Dream students.

Hyperparameter	**MDLM Student**	**Dream Student**
Hidden size d_{model}	96	384
Layers	4	12
Attention heads	4	4
Sequence length	1024	256
Vocabulary size	50,257 (GPT-2)	151,643 (Qwen)

B.2 Optimization

Table 9 lists optimizer-level hyperparameters used for the final student distillation runs. For off-policy training, we follow the corresponding public MDLM and Dream training-code defaults for optimizer and implementation settings where applicable. The phase-specific learning rates and batch sizes are reported separately in Table 10; Dream's off-policy learning rate and batch size are tuned for our student distillation setting.

Table 9: Optimizer hyperparameters for the final student distillation runs.

Parameter	**MDLM**	**Dream**
Optimizer	AdamW	AdamW
β_1	0.9	0.9
β_2	0.999	0.95
ϵ	1×10^{-8}	1×10^{-8}
Weight decay	0	0.01
Gradient clipping	1.0	1.0

B.3 Distillation hyperparameters

Table 10 summarizes phase-level hyperparameters for the final off-policy and on-policy distillation experiments. For on-policy training, we follow the small-batch, low-learning-rate recipe of Lu and Lab [2025].

B.4 Compute and wall-clock.

Table 11 summarizes GPU-hours by phase, reporting MDLM on A100 hardware and Dream on H100 hardware.

Table 10: Run-level hyperparameters for the final MDLM and Dream distillation experiments.

Hyperparameter	**MDLM**	**Dream**
Off-policy batch size	512	1024
Off-policy learning rate	3×10^{-4}	1×10^{-4}
On-policy batch size	16	32
On-policy learning rate	3×10^{-8}	3×10^{-8}
Trajectory states K	4	4
Student sampling steps	1024	256
Confidence exponent γ	1.0	1.0
Trust-region coefficient λ	3×10^{-3}	1×10^{-2}
Trust-region refresh interval	100 steps	150 steps

Table 11: Compute cost per phase. MDLM is reported in A100 GPU-hours and Dream in H100 GPU-hours.

Phase	MDLM (A100)	Dream (H100)
Off-policy	41	1,433
On-policy	61	394
Total	102	1,827

C Data Generation and Preprocessing

We use a fully data-free pipeline: every training token comes from teacher generation. For MDLM we sample from the pretrained teacher starting from pure noise with the standard 1024-step schedule [Sahoo et al., 2024]. For Dream we use unconditional sampling with the same number of steps used at inference.

C.1 Nested data subsets.

To rigorously evaluate the scaling behavior of Section 3, the MDLM corpora are constructed as strictly nested subsets,

$$\mathcal{D}_{1\mathrm{M}} \subset \mathcal{D}_{10\mathrm{M}} \subset \mathcal{D}_{30\mathrm{M}} \subset \mathcal{D}_{100\mathrm{M}},$$

so that performance differences across sizes cannot be attributed to distributional shift across regenerations.

D Qualitative Samples

We provide representative generation samples of MDLM to qualitatively compare the baselines against our proposed method.

D.1 Forward-KL Failure Cases

The following sample illustrates the failure mode of standard Forward-KL distillation (trained on 10M teacher tokens). The model exhibits severe degeneration, characterized by local repetition loops and semantic incoherence.

> *"The same thing to us, for us to even more, especially all things out of down. More interesting things that are a FIFA player who loves someone thinks they think you can do it, otherwise we do not build up, obviously. You have to go time, they talk players and qualities we change, play fit around a team, it doesn't go be a part. That will continue to provide the others in relationship so little, and get an opportunity at wow. They don't try to be cautious. If it have someone from him, they need to call you get him into a ball where high to the things they've done with it. You four-year huddle throughout the next day, football and particularly skilled here. These guys got their good game every night and now, and not have a really fair week, but we know we've did the best guys and go around again. At the moment. K̊eep out a simple air camp and climbing because players are going out and everything you have to*

concentrate on, who kind of the confidence is fine.Älthough you stand this better to go to the board but maybe would return the game. Everybody knows the players weren't position or be playing with a good group team roster player. T̈he question why to not be the team and my demeanor there's three-game reasons where I think the only team at everyone will be and that is well as a team as an important place, everything has the best attitude on continuity inside the guys are being offered for the players, I've put the expectations and win in the right way yet. They're the reason about it too many things to do the other lot of them and going to show out for what it is great, it or they are still being made there but we plan out to us to stay as with a coaching meeting him and only if the players have been able to get over his career but it's only going to be paid. So we're losing the other team, coming forward out of him, "

D.2 Our Method

The following sample is taken from generation of our method. The model largely avoids the short local repetition loops seen in Forward-KL and maintains a loose film/music theme over a longer span.

" feel that's not true that we put on a conversation in the first way we're worried about being all my feeling. But away from what's key thing and other things are supposed to create that space. If true, you just felt smooth it." Which, he's not looking at every bigger rewards. That's a contemporary movies of all what was—and then he leaves me. Or that is reasonable to do my artists to bring him in which I know where these experienced new kids. That's thought of the DJ song: Hollywood "It's always hard for the so good think feel great. Still, I'm still having a respect. And that comes when shutting down just getting to me. This seems different piece and sounding bigger or more sensitive things between things. And, a lack of culture as a lot more of the whole chance to use it connected with those right now, the role in the hands of a sequence, which not stupid and instead of art, you can teach artists and some are so deep as a band on recent top project day. It's excited to be interesting thing with understanding of them constantly and as artistic and develop art and hopefully get one point. People have a real fundamental sense that they tend to learn these advantages, and thoughts where they have the choice for kids? "

E Broader Impacts

This work studies parameter distillation for diffusion language models and does not introduce a new dataset, user-facing application, or deployed system. Its most direct potential positive impact is efficiency. By improving the quality of much smaller DLM students, the method can reduce memory requirements and serving cost relative to using a large teacher model directly. This may make non-autoregressive language generation more practical on constrained hardware, including edge or on-device settings where parallel generation is attractive but large teachers are impractical. The data-free setting may also be useful when the teacher's original training corpus is unavailable or cannot be redistributed.

The same efficiency gains can also have negative impacts. Better small generative models lower the cost of producing large volumes of text, which can facilitate spam, misleading content, impersonation, and other forms of automated misuse. Because our method is explicitly designed to transfer behavior from a teacher model, the student may inherit undesirable properties of the teacher, including toxicity, social biases, unsafe instruction-following behavior, and memorized private information. The data-free setting does not remove these risks: teacher-generated samples and teacher logits can still reflect biases or memorized content from the teacher's original training data. Confidence reweighting may further emphasize high-confidence teacher behavior; if the teacher is confidently wrong or biased, the student can inherit that behavior.

Mitigating these risks requires evaluating distilled students with the same safety, privacy, and fairness checks used for their teachers, rather than assuming that compression or data-free training makes the model safer. Possible mitigations include filtering teacher-generated corpora, measuring toxicity and demographic bias, checking for memorization or privacy leakage, using watermarking or provenance tools for generated text, and applying access controls when the resulting model is capable enough for misuse. Since our experiments focus on modeling quality rather than safety alignment, we view these evaluations as necessary before deploying distilled DLMs in open-ended user-facing settings.

F Limitations

Our study focuses on data-free parameter distillation for masked diffusion language models. While we evaluate two representative regimes, MDLM-110M to a 10.5M student and Dream-7B to a 93M student, both students preserve the architecture family and tokenizer of their corresponding teachers. We therefore do not claim that the same recipe directly transfers to other discrete diffusion parameterizations, architecture-mismatched students, tokenizer-mismatched distillation, or continuous diffusion models.

The method also introduces additional computation and hyperparameters beyond standard off-policy distillation. On-policy training requires student rollouts and teacher evaluations at multiple trajectory states, making the teacher-query cost non-negligible, especially for large teachers such as Dream-7B. The final recipe depends on choices such as the number of trajectory states K, confidence exponent γ, trust-region coefficient λ, refresh interval Δ, and divergence family. We ablate several of these choices, but do not exhaustively explore adaptive schedules, alternative trust-region divergences, or automatic phase-switching criteria.

We focus on masked DLMs because they currently represent the more scalable and practically relevant DLM paradigm. At comparable model scales, masked DLMs generally outperform uniform DLMs, as illustrated by MDLM and Duo, while recent efforts to scale DLMs to frontier sizes, such as the 100B-parameter LLaDA 2, have adopted masked diffusion. Moreover, uniform DLMs such as Duo typically perform diffusion in continuous representation spaces, making existing distillation methods for continuous diffusion models more directly applicable to them. In contrast, masked DLMs operate over discrete masked-token states, introducing a distinct and comparatively underexplored distillation problem. We therefore chose masked DLMs as the primary scope of this study, while acknowledging that extending our analysis and framework to other DLM variants remains important future work.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims made in the abstract and introduction accurately reflect the paper's contributions and scope.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We include a through limitation discussion in Section F

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: For each theoretical result, the paper provides the full set of assumptions and a complete (and correct) proof.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The paper fully disclose all the information needed to reproduce the main experimental results of the paper, and we will release all our code to assist the reproducibility of our experimental results.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We will release our code, models and data.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We specify all the training and test details in the paper.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: DLM distillation with large teachers consumes substantial computational resources, making it impractical to conduct multiple full training runs to obtain error bars.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)

- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provided sufficient information on the computer resources in the main text and appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conducted in the paper conformed, in every respect, with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss potential societal impacts in the appendix.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [Yes]

    Justification: We describe safeguards for responsible release of models in appendix.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We properly credited the creators or original owners of assets (e.g., code, data, models), used in the paper and conformed the license and terms.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We document all assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.