ICML 2026

Discrete Tilt Matching

Likelihood-free reward fine-tuning for masked diffusion LLMs

TL;DR Recast dLLM fine-tuning as **state-level matching of local unmasking posteriors** under progressive reward tilting — no intractable sequence likelihoods. The objective is a weighted cross-entropy with an **explicit tilted-posterior minimizer**, a **variance-reducing control variate**, and a **KL guarantee** on the fine-tuned output.

Yuyuan Chen* · Shiyi Wang* · Peter Potaptchik · Jaeyeon Kim · Michael S. Albergo

Harvard University · University of Oxford · IAIFI · Kempner Institute * equal contribution

Paper & code
arXiv:2604.18739

THE PROBLEM

why likelihood-based RL fails

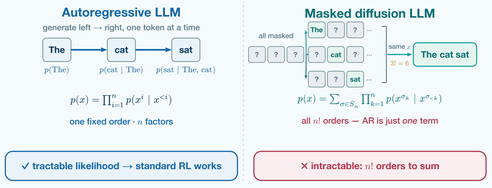
1 Diffusion LLMs unmask gradually

A masked diffusion LLM turns a **fully-masked** prompt into a **clean** sequence by revealing tokens over time $t : 0 \to 1$, in **any order** along a schedule $\alpha(t)$ — unlike an autoregressive model's fixed left-to-right factorization.

Generation = gradual, any-order unmasking. Fine-tuning must respect this state-by-state structure.

2 Why RL breaks for diffusion LLMs

RL fine-tuning (GRPO-style) needs the **sequence likelihood** $p(x)$ — but a dLLM can reach the same sequence through **exponentially many unmasking orders**, so $p(x)$ sums over all of them and is **intractable**. Prior methods lean on biased surrogates or high-variance estimators.

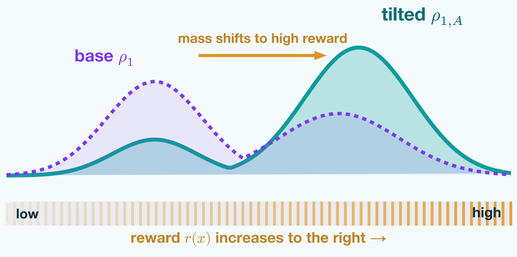


Fix: fine-tune with state-level quantities, not sequence likelihood

3 Our reframe: tilt the target

Rather than policy optimization, push the pretrained target ρ_1 toward high reward with an **exponential tilt** of strength A:

$$\rho_{1,A}(x) \propto \rho_1(x)\, e^{A r(x)}$$

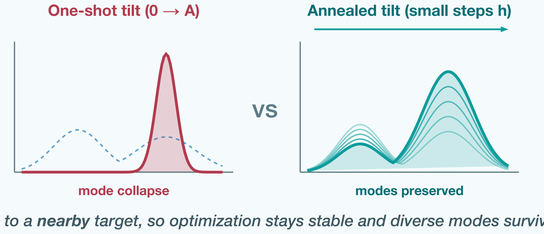


Reward r reweights probability mass toward preferred sequences; A sets how hard.

4 Anneal — don't jump

Tilting to the full target in **one shot** has explosive variance and **collapses modes**. Instead we **anneal** $a : 0 \to A$ in small steps h, matching a sequence of nearby targets:

$$\rho_{1,a}(x) \propto \rho_1(x) e^{a r(x)}, \quad a : 0 \to A \quad \text{in steps} \quad h$$



*Each step moves to a **nearby** target, so optimization stays stable and diverse modes survive.*

THE METHOD

tilt · reweight · match

5 The learning target: the next posterior

A dLLM's **only** learnable piece is the local **unmasking posterior** at a masked position i — and a probability is just an **expectation of an indicator**:

$$\pi_a(v \mid x_t, i) = \Pr_a[x_1^i = v \mid x_t] = \mathbb{E}_a[\mathbb{1}\{x_1^i = v\} \mid x_t]$$

Fine-tuning anneals $a \to a + h$, so each step's target is the **next** posterior π_{a+h}. If we knew it, training would be **ordinary dLLM cross-entropy** — at every masked position, pull the model's posterior π_θ onto that target.

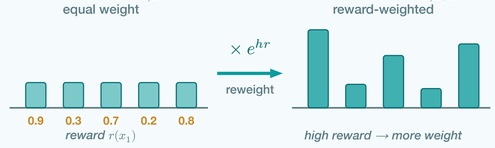
The catch. π_{a+h} is an expectation under the **next** tilt $\rho_{1,a+h}$ — which we can neither **sample** nor **evaluate**. The rest of this column removes that catch.

6 Esscher: rewrite it as a reward-weighted average

One transform fixes both. The next tilt only **reweights** the current one, $\rho_{1,a+h} \propto \rho_{1,a} e^{hr}$, so **any** expectation under the next tilt is a reward-weighted expectation under the current one:

$$\mathbb{E}_{a+h}[\cdot] = \frac{\mathbb{E}_a[e^{hr}\,\cdot]}{\mathbb{E}_a[e^{hr}]}$$

(Esscher)



*No resampling: reward-weight the rollouts you **already have** under $\rho_{1,a}$ and they stand in for draws from the next tilt $\rho_{1,a+h}$.*

Applied to the posterior — itself an expectation of an indicator — the **unknown target** becomes a ratio of two averages we can read off those rollouts:

$$\pi_{a+h}(v \mid x_t, i) = \frac{\mathbb{E}_a[e^{hr}\,\mathbb{1}\{x_1^i = v\} \mid x_t]}{\mathbb{E}_a[e^{hr} \mid x_t]}$$

7 The objective the target hands us

Clear the denominator and this identity becomes **exactly the first-order optimality condition** of a weighted cross-entropy. So we don't design a loss — the target **hands us one**: roll out from tilt a, weight by e^{hr}, and regress the model onto each rollout's **realized token** T_c:

$$\mathcal{L}^{\text{c-DTM}}_{a\to a+h}(\theta) = \int_0^1 \lambda(t)\, \mathbb{E}_a\left[\underbrace{\frac{e^{hr(x_1)}}{\text{reward weight}}}\sum_{i:x_1^i=m} \text{CE}\left(\underbrace{T_c}_{\text{realized token}} \,\|\, \underbrace{\pi_\theta(\cdot \mid x_t, i)}_{\text{model}}\right)\right] \mathrm{d}t$$

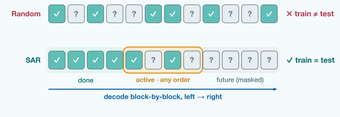
With $c = 0$ the target T_c is simply **the realized token** $\mathbb{1}\{v = x_1^i\}$; a **control variate** c — unbiased by construction — leaves the minimizer fixed but **cuts gradient variance**:

$$T_c(v \mid x_t, i, x_1) = \frac{c\,\pi_a(v \mid x_t, i) + \left(e^{hr(x_1)} - c\right)\mathbb{1}\{v = x_1^i\}}{e^{hr(x_1)}}$$

✓ **Unique minimizer ≡ exact π_{a+h} — no bias** ✓ **Loss bounds the KL on the fine-tuned law**

8 SAR-aligned training

SOTA dLLMs decode **semi-autoregressively** — block-by-block left → right, but **any order within a block**. DTM masks only the **active block**, so training states match those seen at decode time — preserving the KL guarantee.



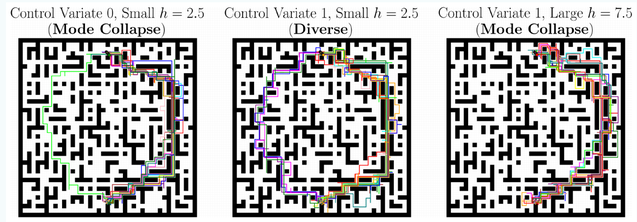
♻ **Replay buffer** — one rollout yields many training states x_t, amortizing costly online generation.

RESULTS

planning · math · efficiency

9 Mode collapse, made visible

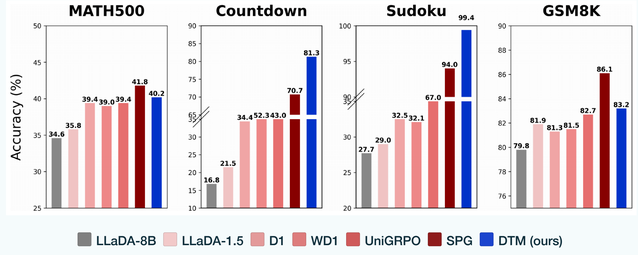
On a synthetic **maze-planning** task (reward = stay away from center), we overlay many generated paths:



*Control variate + **small** h keeps solutions **diverse** (center). Dropping the control variate (left) or taking **large** h (right) → visible **mode collapse** — exactly the intuition of Panel 4.*

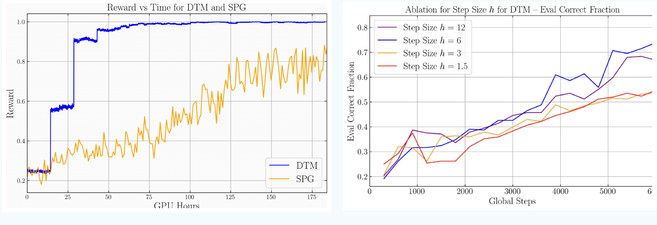
10 Benchmark gains at scale

Fine-tuning **LLaDA-8B-Instruct**, **DTM** is competitive with or beats prior RL methods for dLLMs:



Sudoku 27.7 → 99.2 · Countdown best **math competitive**

11 Efficient — and robust to the step size



Faster convergence (left, Fig 4). On Sudoku, same 8×H100 budget, **DTM** reaches higher reward **faster** than SPG — state-level posteriors sidestep the costly, high-variance sequence-likelihood estimates that slow likelihood-based RL.

Robust to the step size (right, Fig 3). Sweeping $h \in \{1.5, 3, 6, 12\}$ on Countdown, **DTM** improves under **every** setting; a moderate $h = 6$ is the sweet spot — fast, stable gains.

12 Takeaways

- **Likelihood-free, state-level** fine-tuning for dLLMs with an explicit minimizer and a KL guarantee.
- Especially strong on **structured planning** (Sudoku, Countdown); competitive on math, improving further with more decode budget.
- Honest limit: gains on **long-horizon** math reasoning are smaller — local posteriors help most where reasoning is local.

Broader view. Likelihood-based RL may not be the right frame for diffusion LLMs — **DTM** opens post-training paradigms aligned with their structure.