# Accelerating Diffusion Language Models via Trajectory Distillation

Tunyu Zhang, Dimitris Metaxas (PI)

CBIM Lab, CS Department, Rutgers University

Feb 27, 2026

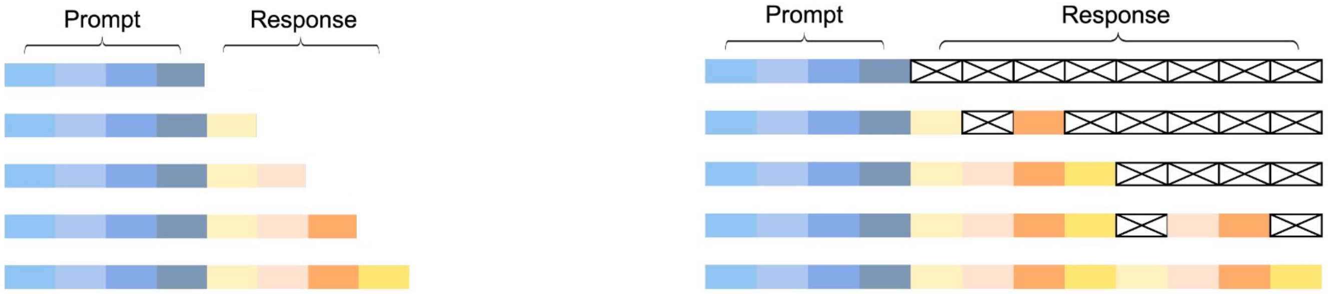
# Content

- Background: Diffusion Language Model (DLM)
  - Masked Diffusion Language Model (MDLM)
  - The Challenges of Few-Step Decoding
- **T3D**: **T**rajectory **D**istillation via **D**irect **D**iscriminative Optimization
  - Trajectory Self-Distillation
  - DDO Objective: Reverse KL
  - Experiments
  - Theory Justification
- Future Directions

# Background: Masked Diffusion Language Model

**Auto-Regressive Model (ARM)**: Generates text sequentially, token by token (next-token prediction).

**Masked Diffusion Language Model (MDLM)**: Generates text by iteratively refining masked tokens in parallel. (mask predictor).




(a) Auto-Regressive Generation

(b) Diffusion Process Generation

Figure: Compare the generation process of AR model (a) with DLM (b).[1]

[1] Nie, Shen, et al. "Large Language Diffusion Models." The Thirty-ninth Annual Conference on Neural Information Processing Systems.

# Background: Formulation of MDLM

**Masked Diffusion Language Model (MDLM)** [1] is a simple and efficient variant of DLM.

**The forward process:** Given a clean input $\mathbf{x}$, for all $t$, we have $q(\mathbf{z}_t|\mathbf{x}) = \text{Cat}(\mathbf{z}_t; \alpha_t \mathbf{x} + (1 - \alpha_t)\mathbf{m})$ $\alpha_t \in [0,1], \alpha_0 \approx 1, \alpha_1 \approx 0$

*We corrupt each token in a probability $\alpha_t$ at different time step.*

**The Reverse process:** By the properties of MDLM, we can get a simple form of posterior $q(\mathbf{z}_s \mid \mathbf{z}_t, \mathbf{x})$, where $s < t$

$$q(\mathbf{z}_s|\mathbf{z}_t,\mathbf{x}) = \begin{cases} \text{Cat}(\mathbf{z}_s;\mathbf{z}_t) & \mathbf{z}_t \neq \mathbf{m}, \\ \text{Cat}\left(\mathbf{z}_s;\frac{(1-\alpha_s)\mathbf{m}+(\alpha_s-\alpha_t)\mathbf{x}}{1-\alpha_t}\right) & \mathbf{z}_t = \mathbf{m}. \end{cases}$$

*We can model the reverse process by parameterization of* $\mathbf{x}_\theta$

$$p_\theta(\mathbf{z}_s|\mathbf{z}_t) = q(\mathbf{z}_s|\mathbf{z}_t,\mathbf{x}=\mathbf{x}_\theta(\mathbf{z}_t,t)) = \begin{cases} \text{Cat}(\mathbf{z}_s;\mathbf{z}_t), & \mathbf{z}_t \neq \mathbf{m}, \\ \text{Cat}\left(\mathbf{z}_s;\frac{(1-\alpha_s)\mathbf{m}+(\alpha_s-\alpha_t)\mathbf{x}_\theta(\mathbf{z}_t,t)}{1-\alpha_t}\right). & \mathbf{z}_t = \mathbf{m}, \end{cases}$$

[1] Sahoo, Subham, et al. "Simple and effective masked diffusion language models." *Advances in Neural Information Processing Systems* 37 (2024): 130136-130184.

# Background: Training of MDLM

It can yield the following NELBO:

$$\mathcal{L}_{\text{NELBO}}^{\infty} = \mathbb{E}_q \int_{t=0}^{t=1} \frac{\alpha_t'}{1-\alpha_t} \log\langle \mathbf{x}_\theta(\mathbf{z}_t, t), \mathbf{x}\rangle \mathrm{d}t$$

We apply masked diffusion to language modeling over sequence $\mathbf{x}^{1:L}$ with $L$ tokens.
Here, we have to introduce a strong assumption:

(**Mean Field Assumption**)[1] *The denoising process factorizes independently across tokens*

$$p_\theta(\mathbf{z}_s^{1:L} \mid \mathbf{z}_t^{1:L}) = \prod_{\ell=1}^{L} p_\theta(\mathbf{z}_s^{\ell} \mid \mathbf{z}_t^{1:L})$$

With this assumption, the NELBO can be written as:

$$\mathcal{L}_{\text{NELBO}}^{\infty} = \mathbb{E}_q \int_{t=0}^{t=1} \frac{\alpha_t'}{1-\alpha_t} \sum_{\ell} \log\langle \mathbf{x}_\theta^{\ell}(\mathbf{z}_t^{1:L}, t), \mathbf{x}^{\ell}\rangle \mathrm{d}t$$

[1] Sahoo, Subham, et al. "Simple and effective masked diffusion language models." *Advances in Neural Information Processing Systems* 37 (2024): 130136-130184.

# Background: Overview of MDLM

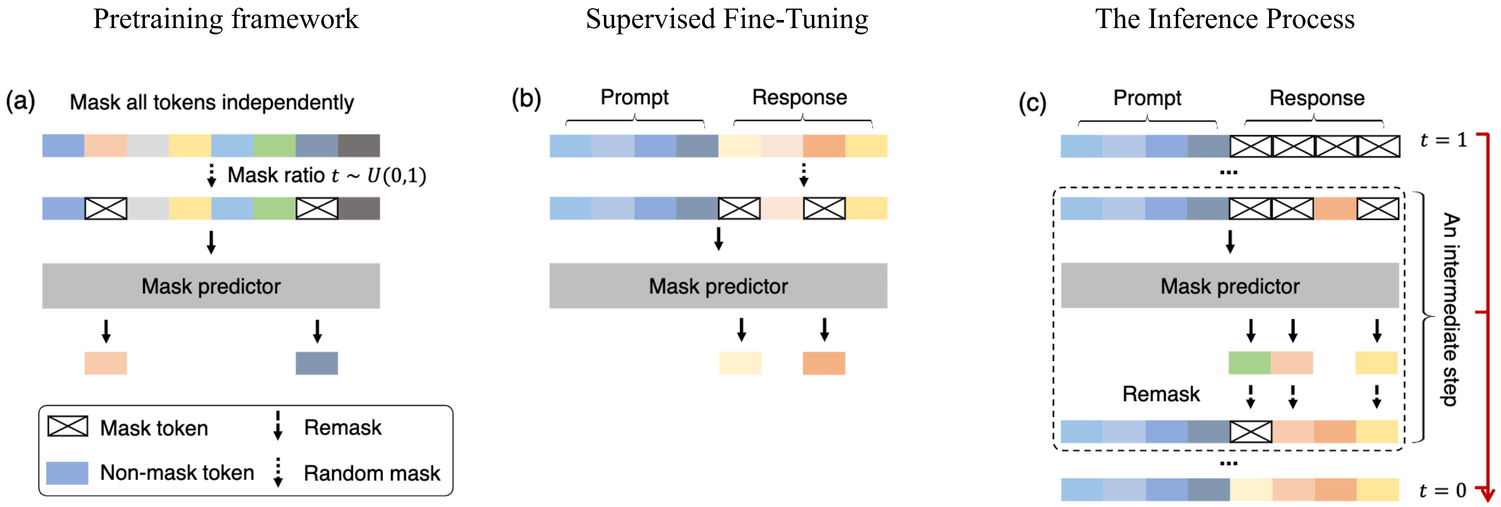



Figure: Overview of Masked Diffusion Language Model (LLaDA).[1]

[1] Nie, Shen, et al. "Large Language Diffusion Models." The Thirty-ninth Annual Conference on Neural Information Processing Systems.

# Background: Diffusion For text Generation

## A NEW TEXT GENERATION PARADIGM

### Diffusion Language Models

- Start from noise or masks and iteratively refine
- Predict multiple tokens in parallel
- Use full bidirectional context at every step

### Why This is Interesting?

- Refinement without long-context overhead
- Less cascading error than autoregressive decoding
- Better reasoning through iterative correction

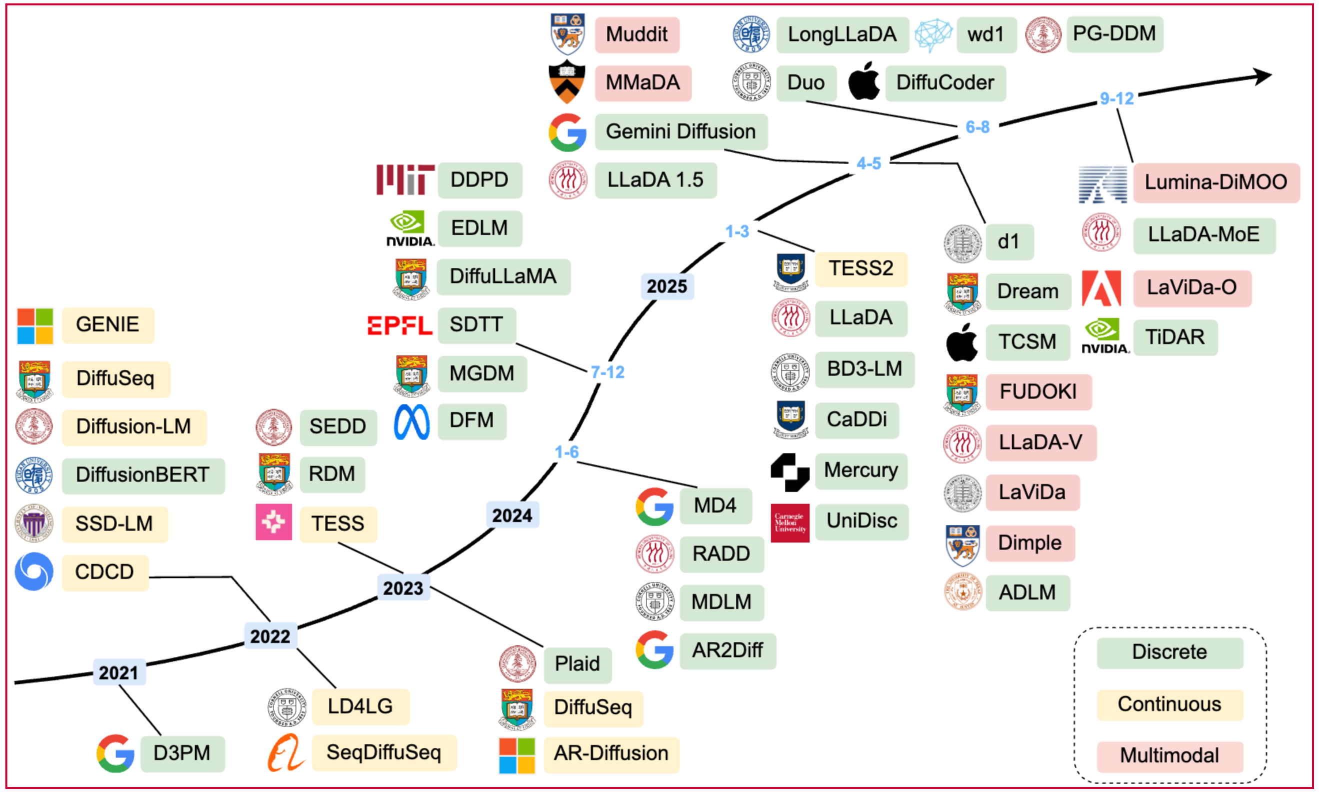



Figure: Timeline of Diffusion Language Models.[1]

[1] Li, Tianyi, et al. "A survey on diffusion language models." *arXiv preprint arXiv:2508.10875* (2025).

# Background: The Challenges of Few-Step Decoding

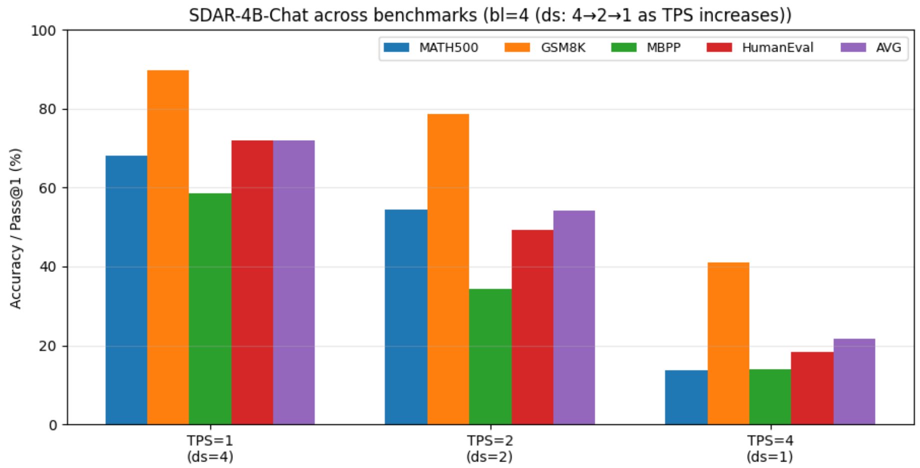



**Figure: Performance degradation of SDAR-4B-Chat as tokens-per-step (TPS) increases.** As more tokens are predicted per step (with fewer decoding steps per block), accuracy consistently drops across MATH500, GSM8K, MBPP, and HumanEval. The decline becomes especially severe at TPS=4, indicating that **aggressive parallel token prediction substantially harms generation quality**.

**Why Few-Step Fails:**

- Train/Test mismatch
- Factorization Error Amplification
- …

**Why Solving Few-Step Matters**:

- 🚀 Lower latency
- 💰 Lower compute cost
- 📈 Higher throughput
- …

# Method

Recall the mean-field assumption:

**(Mean Field Assumption)** *The denoising process factorizes independently across tokens*

$$p_\theta(\mathbf{z}_s^{1:L} \mid \mathbf{z}_t^{1:L}) = \prod_{\ell=1}^{L} p_\theta(\mathbf{z}_s^{\ell} \mid \mathbf{z}_t^{1:L})$$

- This means we assume all tokens are independent
- **Not correct** in natural language: there should be high correlation between tokens in a sentence
- It will lead a huge approximation error in few-step generation

*Can we reduce mean-field approximation error under tight step budgets for DLLMs?*

# Method

Some previous works explored this mean-field error:

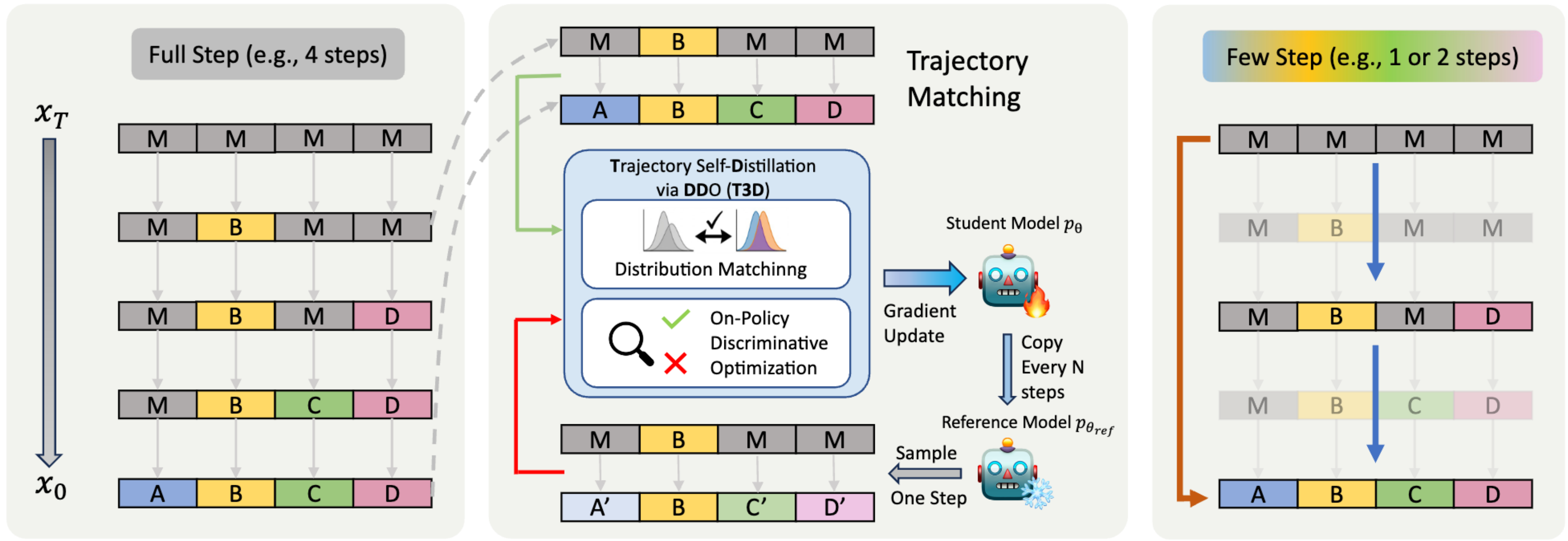
- **ReDi** (Rectified Discrete Flow[1]). **Not compatible with MDLM**
- **EDLM** (Energy-Based Diffusion Language Models For Text-Generation[2]). Need MCMC rejection sampling. **High Cost**

**T3D:** Few-Step Diffusion Language Models via **T**rajectory Self-**D**istillation with **D**irect **D**iscriminative Optimization[3]

- A training framework for **efficient** diffusion language models.
- **T3D** uses **mode-seeking objective** to do training.
- Aggressive few-step generation while **preserving the diffusion capabilities**.
- Totally **label-free** training.

[1] Yoo, Jaehoon, et al. "ReDi: Rectified Discrete Flow." *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.
[2] Xu, Minkai, et al. "Energy-Based Diffusion Language Models for Text Generation." *The Thirteenth International Conference on Learning Representations*.
[3] Zhang, Tunyu, et al. "T3D: Few-Step Diffusion Language Models via Trajectory Self-Distillation with Direct Discriminative Optimization." *arXiv preprint arXiv:2602.12262* (2026).

# T3D: Overview




(a) Teacher Full Trajectory (b) Trajectories Objective Learning of **T3D** (c) Few-Step Decoding

Figure: Overview of T3D for enabling few-step diffusion decoding.[1]

(a) **Teacher full trajectory.** A pretrained diffusion language model generates sequences through a full diffusion trajectory.
(b) **Trajectory-level objective learning.** T3D trains a few-step student model via trajectory self-distillation on teacher rollout trajectories, where ***Direct Discriminative Optimization (DDO)***[1] is used to perform mode-seeking trajectory matching.
(c) **Few-step decoding.** After training, the student can decode using significantly fewer diffusion steps (e.g., 1–2 steps per block)

[1] Zheng, Kaiwen, et al. "Direct Discriminative Optimization: Your Likelihood-Based Visual Generative Model is Secretly a GAN Discriminator." *Forty-second International Conference on Machine Learning*.

# T3D: Trajectory Self-Distillation

Given a pretrained teacher model $p_\phi$, which always do full-step diffusion generation, we train a few-step student model $p_\theta$ :

1. We sample from $p_\phi$ to collect real trajectory $(x_0, x_1, \ldots, x_T)$
2. We can draw a clean/noisy pair from this trajectory: $(x_0, x_t)$
3. We need learn the joint distribution of $(x_0, x_t)$

# T3D: Trajectory Self-Distillation

Given a pretrained teacher model $p_\phi$, which always do full-step diffusion generation, we train a few-step student model $p_\theta$ :

1. We sample from $p_\phi$ to collect real trajectory $(x_0, x_1, \ldots, x_T)$
2. We can draw a clean/noisy pair from this trajectory: $(x_0, x_t)$
3. We need learn the joint distribution of $(x_0, x_t)$

Takeaway

- On-policy training: train on what you execute
- Remove distribution mismatch
- No architecture change required
- Label free

# T3D: DDO style Objective Training

Direct Discriminative Optimization (DDO)[1]

- A GAN-inspired objective for likelihood-based generative models
- Implicitly parameterizes the discriminator
- Reverse KL-style Objective

We adopt DDO in our framework:

*KL Objective*

$$L_{\text{traj}}(\theta) = -\mathbb{E}_{p_\phi(x_t)}\mathbb{E}_{x_0 \sim p_\phi(\cdot|x_t)}\left[\log p_\theta(x_0 \mid x_t)\right]$$

*Reverse KL Objective*

$$L_{\text{traj-DDO}}(\theta) = \mathbb{E}_{x_t \sim p_\phi(x_t)}\left[l(\theta)\right]$$

$$l(\theta) = -\log \sigma\left(\mathbb{E}_{x_0 \sim p_\phi(\cdot|x_t)}\left[\log \frac{p_\theta(x_0 \mid x_t)}{p_{\theta_{\text{ref}}}(x_0 \mid x_t)}\right]\right) - \log\left(1 - \sigma\left(\mathbb{E}_{x_0 \sim p_{\theta_{\text{ref}}}(\cdot|x_t)}\left[\log \frac{p_\theta(x_0 \mid x_t)}{p_{\theta_{\text{ref}}}(x_0 \mid x_t)}\right]\right)\right)$$

[1] Zheng, Kaiwen, et al. "Direct Discriminative Optimization: Your Likelihood-Based Visual Generative Model is Secretly a GAN Discriminator." *Forty-second International Conference on Machine Learning*.

# T3D: DDO style Objective Training

Direct Discriminative Optimization (DDO)[1]

- A GAN-inspired objective for likelihood-based generative models
- Implicitly parameterizes the discriminator
- Reverse KL-style Objective

We adopt DDO in our framework:

*KL Objective* ✗

$$L_{\text{traj}}(\theta) = -\mathbb{E}_{p_\phi(x_t)}\mathbb{E}_{x_0\sim p_\phi(\cdot|x_t)}\left[\log p_\theta(x_0 \mid x_t)\right]$$

*Reverse KL Objective* ✓

$$L_{\text{traj-DDO}}(\theta) = \mathbb{E}_{x_t\sim p_\phi(x_t)}\left[\, l(\theta)\,\right]$$

$$l(\theta) = -\log\sigma\left(\mathbb{E}_{x_0\sim p_\phi(\cdot|x_t)}\left[\log\frac{p_\theta(x_0\mid x_t)}{p_{\theta_{\text{ref}}}(x_0\mid x_t)}\right]\right) - \log\left(1-\sigma\left(\mathbb{E}_{x_0\sim p_{\theta_{\text{ref}}}(\cdot|x_t)}\left[\log\frac{p_\theta(x_0\mid x_t)}{p_{\theta_{\text{ref}}}(x_0\mid x_t)}\right]\right)\right)$$

## Takeaway

- Forward KL → mode covering
- Few-step needs confident mode-seeking
- Reverse KL / DDO encourages commitment

[1] Zheng, Kaiwen, et al. "Direct Discriminative Optimization: Your Likelihood-Based Visual Generative Model is Secretly a GAN Discriminator." *Forty-second International Conference on Machine Learning*.

# T3D: Path Consistency Regularization

We introduced a path consistency regularization:

- This is KL-style objective combined with DDO loss
- Pay more attention to early decoded tokens
- Improve overall decoding stability

$$L_{\text{path}}(\theta) = -\mathbb{E}_{p_\phi(x_t)}\mathbb{E}_{x_0 \sim p_\phi(\cdot|x_t)}\left[\sum_i w_i \left(\log p_\theta(x_0^i \mid x_t^{(i)})\right)\right]$$

$$w_i = \frac{B - \pi_i + 1}{B}$$

where $w_i$ is the decode order of token $x^i$

# T3D: Overview

**Algorithm 1** `T3D` Training

**Require:** Teacher model $p_\phi$
**Require:** Student model $p_\theta$ (initialized from teacher)
**Require:** Path regularization weight $\lambda$
1: Sample trajectory pairs $(\boldsymbol{x}_0, \boldsymbol{x}_t) \sim p_\phi$
2: **repeat**
3: Set reference model $p_{\theta_{ref}} \leftarrow \mathrm{StopGrad}(p_\theta)$
4: Compute trajectory DDO loss $\mathcal{L}_{\mathrm{traj-DDO}}$
5: Compute path consistency loss $\mathcal{L}_{\mathrm{path}}$
6: Update student model using

$$\mathcal{L} = \mathcal{L}_{\mathrm{traj-DDO}} + \lambda \mathcal{L}_{\mathrm{path}}$$

7: **until** convergence
**output** $p_\theta$

# Experiments: Few-Step Decoding

| TokPS | Method | SD | Block Size = 4 | | | | Block Size = 8 | | | | Average | Improvement (%) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | MATH500 | GSM8K | MBPP | HumanEval | MATH500 | GSM8K | MBPP | HumanEval | | |
| **SDAR-1.7B-Chat** | | | | | | | | | | | | |
| 2 | Original Model | - | 39.40 | 63.00 | 30.40 | 32.93 | 33.60 | 55.88 | 27.80 | 37.20 | 40.03 | - |
| | SFT | ✗ | 43.00 | 61.79 | 30.00 | 34.76 | 36.80 | 62.55 | 27.20 | 37.80 | 41.74 | ↑ 4.28 |
| | ReDi | ✓ | 40.60 | 63.99 | 13.20 | 16.46 | 36.40 | 62.17 | 12.80 | 13.41 | 32.38 | ↓ 19.11 |
| | dParallel | ✓ | 43.40 | 68.23 | 22.20 | 24.39 | 45.20 | 67.70 | 23.20 | **26.83** | 40.14 | ↑ 0.29 |
| | Naive TD | ✓ | 43.00 | 66.19 | 21.20 | 19.51 | 39.00 | 63.99 | 17.20 | 18.29 | 36.05 | ↓ 9.94 |
| | `T3D` (Ours) | ✓ | **47.00** | **70.96** | **27.20** | **30.49** | **47.80** | **68.84** | **26.60** | 25.61 | **43.06** | **↑ 7.59** |
| 4 | Original Model | - | 5.00 | 13.34 | 10.60 | 12.20 | 4.80 | 12.74 | 10.20 | 10.37 | 9.91 | - |
| | SFT | ✗ | 22.40 | 36.62 | 6.20 | 5.49 | 20.00 | 39.65 | 4.40 | 7.93 | 17.84 | ↑ 80.05 |
| | ReDi | ✓ | 15.00 | 32.45 | 3.40 | 5.49 | 12.80 | 29.72 | 4.00 | 4.88 | 13.47 | ↑ 35.95 |
| | dParallel | ✓ | 22.80 | 45.26 | **10.20** | 12.20 | 25.40 | **42.91** | **10.40** | 11.59 | **22.60** | ↑ 128.09 |
| | Naive TD | ✓ | 24.20 | **46.02** | 6.00 | **15.24** | **26.60** | 39.73 | 9.00 | 9.76 | 22.07 | ↑ 122.78 |
| | `T3D` (Ours) | ✓ | **25.60** | 42.91 | 9.40 | **15.24** | 24.40 | 37.38 | 9.20 | **14.02** | 22.27 | **↑ 124.79** |
| **SDAR-4B-Chat** | | | | | | | | | | | | |
| 2 | Original Model | - | 54.40 | 78.77 | 34.20 | 49.39 | 49.60 | 72.33 | 33.40 | 46.95 | 52.38 | - |
| | SFT | ✗ | 54.60 | 54.60 | 26.80 | 37.20 | 54.44 | 77.41 | 25.60 | 29.88 | 46.76 | ↓ 10.73 |
| | ReDi | ✓ | 41.00 | 73.62 | 20.00 | 21.95 | 23.60 | 71.87 | 19.20 | 23.17 | 36.80 | ↓ 29.74 |
| | dParallel | ✓ | 52.60 | 76.57 | 23.80 | 39.63 | 51.20 | 75.97 | 18.20 | 28.66 | 45.83 | ↓ 12.51 |
| | Naive TD | ✓ | 50.80 | 72.78 | 21.40 | 24.39 | 47.00 | 68.01 | 19.60 | 22.56 | 40.82 | ↓ 22.07 |
| | `T3D` (Ours) | ✓ | **60.00** | **83.85** | **38.80** | **51.83** | **61.60** | **81.96** | **37.00** | **56.10** | **58.89** | **↑ 12.43** |
| 4 | Original Model | - | 13.80 | 41.09 | 14.00 | 18.29 | 16.80 | 41.02 | 10.00 | 16.46 | 21.43 | - |
| | SFT | ✗ | 39.00 | 48.14 | 9.00 | 15.85 | 40.20 | 55.42 | 8.80 | 11.59 | 28.50 | ↑ 32.98 |
| | ReDi | ✓ | 25.40 | 53.30 | 5.00 | 7.32 | 20.20 | 47.84 | 6.80 | 6.71 | 21.57 | ↑ 0.65 |
| | dParallel | ✓ | 34.20 | 45.94 | 13.20 | 20.73 | 40.80 | 53.83 | 9.60 | 20.12 | 29.80 | ↑ 39.05 |
| | Naive TD | ✓ | 39.00 | 57.92 | 10.40 | 17.07 | 36.20 | 50.80 | 12.40 | 9.00 | 28.92 | ↑ 34.95 |
| | `T3D` (Ours) | ✓ | **47.80** | **69.90** | **22.60** | **23.78** | **44.80** | **63.99** | **21.20** | **23.17** | **39.66** | **↑ 85.02** |

T3D consistently achieves the best or near-best performance among all self-distillation baselines

# Experiments: Full-Step Decoding Preserving

## Experimental Setting

- Study potential **diffusion behavior forgetting** after few-step distillation
- Revert distilled models back to **full-step diffusion**
- Decode using **static one-token-per-step schedule**
- Evaluate performance **without any retraining**

| Method | MATH500 | GSM8K | MBPP | HumanEval |
|---|---|---|---|---|
| **SDAR-1.7B-Chat** | | | | |
| Original Model | 59.40 | 80.59 | 45.20 | 59.76 |
| SFT | 52.00 | 73.09 | 44.20 | 60.37 |
| ReDi | 47.00 | 73.77 | 27.60 | 31.10 |
| dParallel | 0.40 | 0.23 | 34.60 | 43.29 |
| Naive TD | 49.80 | 72.40 | 35.20 | 32.93 |
| **T3D (Ours)** | **56.80** | **78.01** | **41.20** | **57.32** |
| **SDAR-4B-Chat** | | | | |
| Original Model | 68.00 | 89.84 | 58.60 | 71.95 |
| SFT | 60.20 | 86.05 | 50.20 | 69.51 |
| ReDi | 50.40 | 82.03 | 34.00 | 37.80 |
| dParallel | 13.20 | 2.88 | 34.00 | 48.17 |
| Naive TD | 57.40 | 82.11 | 37.60 | 43.90 |
| **T3D (Ours)** | **70.00** | **89.31** | **54.20** | **73.78** |

T3D enables few-step generation without sacrificing full diffusion performance

# Experiments: Efficiency of Dynamic Decoding

## Setting

- Evaluate under **dynamic decoding** (confidence-based adaptive decoding)
- Although T3D is trained for **static few-step**, we test real-world inference setting
- Configuration:
    - Block size = 4
    - 4 steps per block
    - Confidence threshold = 0.9
- **No additional retraining**

## Results

- ✅ T3D improves **accuracy**
- ✅ Reduces **average decoding steps**
- ✅ Achieves **lower latency**
- ✅ Delivers **higher throughput**

| Method | TPS↑ | Latency↓ | Avg Steps↓ | Avg Len | Acc↑ |
|---|---|---|---|---|---|
| `MATH500` | | | | | |
| Original | 657.72 | 1.10 | 196.19 | 721.90 | 39.00 |
| ReDi | 715.71 | 1.04 | 198.24 | 757.05 | 27.00 |
| dParallel | 692.08 | 0.95 | 170.22 | 653.98 | 45.80 |
| Naive TD | 693.85 | 0.97 | 177.99 | 678.55 | 44.00 |
| **T3D** (Ours) | **791.23** | **0.66** | **137.95** | 525.50 | **49.40** |
| `GSM8K` | | | | | |
| Original | 580.60 | 0.43 | 71.12 | 249.52 | 61.56 |
| ReDi | 636.58 | 0.49 | 84.63 | 311.99 | 54.89 |
| dParallel | 805.02 | 0.39 | 83.23 | 310.58 | 67.02 |
| Naive TD | 696.99 | 0.47 | 89.78 | 330.82 | 62.40 |
| **T3D** (Ours) | **843.05** | **0.37** | **83.03** | 312.48 | **72.40** |
| `MBPP` | | | | | |
| Original | 262.66 | 0.36 | 27.25 | 93.64 | 23.40 |
| ReDi | 298.83 | 0.21 | 17.11 | 62.57 | 10.00 |
| dParallel | 215.65 | 0.63 | 36.03 | 135.16 | 8.40 |
| Naive TD | **314.99** | 0.31 | 26.43 | 98.80 | 9.80 |
| **T3D** (Ours) | 313.18 | **0.19** | **16.94** | 61.62 | **23.60** |
| `HumanEval` | | | | | |
| Original | 175.48 | 0.73 | 36.56 | 127.54 | 33.54 |
| ReDi | 163.77 | 0.47 | 21.23 | 76.75 | 10.00 |
| dParallel | 130.34 | 0.48 | 17.41 | 62.19 | 23.78 |
| Naive TD | 216.39 | 0.29 | 17.15 | 62.10 | 23.17 |
| **T3D** (Ours) | **222.68** | **0.26** | **16.21** | 58.10 | **29.27** |

T3D generalizes beyond static decoding.

# T3D: Theory Justification

**On-policy Optimum**

In **Proposition 4.3**, we defined a *on-policy risk for distillation:*

$$\mathcal{R}_t(\theta) := \mathbb{E}_{\boldsymbol{x}_t \sim p_\phi(\mathbf{x}_t)} \Big[ D\big(p_\phi(\cdot \mid \boldsymbol{x}_t) \,\|\, p_\theta(\cdot \mid \boldsymbol{x}_t)\big) \Big]$$

We can prove the optimal solution of T3D is the optimal solution for this risk:

$$\mathcal{R}_t(\theta^\star_{\mathrm{Tra}}) \;\leq\; \mathcal{R}_t(\theta^\star).$$

## Intuition

- Student is trained under the same intermediate-state distribution encountered at inference time
- This ensures on-policy alignment with the teacher
- Avoids mismatch caused by marginal-only (random mask) training

# T3D: Theory Justification

**Reduce mean-field (factorization) error**

Following the definition in ReDi, the Conditional Total Correlation (TC) measures the factorization error:

$$TC_J(\mathbf{x}_s \mid \mathbf{x}_t) := \mathbb{E}_{\mathbf{x}_t}\left[\mathrm{KL}\left(p(\mathbf{x}_s \mid \mathbf{x}_t) \,\Big\|\, \prod_{i=1}^{N} p(\mathbf{x}_s^i \mid \mathbf{x}_t)\right)\right]$$

In **Theorem 4.5**, we proved T3D can induce lower Conditional Total Correlation

$$\mathbb{E}_t\left[TC_{J_{\theta^*}^{\mathrm{Tra}}}(x_0 \mid x_t) + TC_{J_{\theta^*}^{\mathrm{Tra}}}(x_t \mid x_T)\right] \leq \mathbb{E}_t\left[TC_{J_{\phi}^{\mathrm{Tra}}}(x_0 \mid x_t) + TC_{J_{\phi}^{\mathrm{Tra}}}(x_t \mid x_T)\right]$$

Intuition

- Trains the student on full joint trajectory states
- T3D encourages internal consistency across tokens
- Reduces conditional total correlation

# Future Directions

- Summary
  - **Distribution Alignment** → train on what you execute
  - **Objective Alignment** → optimize for few-step behavior

- Next Steps
  - Task/Domain specific distillation
  - Larger and better datasets
  - Incorporate Reinforcement Learning
  - Scale up to larger and stronger models

# References

[1] Nie, Shen, et al. "Large Language Diffusion Models." The Thirty-ninth Annual Conference on Neural Information Processing Systems.

[2] Li, Tianyi, et al. "A survey on diffusion language models." *arXiv preprint arXiv:2508.10875* (2025).

[3] Yoo, Jaehoon, et al. "ReDi: Rectified Discrete Flow." *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

[4] Xu, Minkai, et al. "Energy-Based Diffusion Language Models for Text Generation." *The Thirteenth International Conference on Learning Representations*.

[5] Zhang, Tunyu, et al. "T3D: Few-Step Diffusion Language Models via Trajectory Self-Distillation with Direct Discriminative Optimization." *arXiv preprint arXiv:2602.12262* (2026).

[6] Zheng, Kaiwen, et al. "Direct Discriminative Optimization: Your Likelihood-Based Visual Generative Model is Secretly a GAN Discriminator." *Forty-second International Conference on Machine Learning*.

# Thanks

RUTGERS
UNIVERSITY