Tiny Recursive Language Diffusion Models

Anonymous ACL submission

Abstract

Autoregressive large language models (ARMs) are effective but brittle on tasks where a single wrong token invalidates the full output and where iterative error correction is essential. In parallel, recent work shows that *tiny* networks can perform strong recursive reasoning on hard puzzle-like sequence tasks via deep supervision and latent recursion (Jolicoeur-Martineau, 2025; Wang et al., 2025). Separately, masked diffusion language models (MDMs) demonstrate that diffusion-based, non-autoregressive generation can scale and exhibit core language-model capabilities while enabling iterative refinement (Nie et al., 2025; Austin et al., 2021; Shi et al., 2024; Sahoo et al., 2024; Ou et al., 2024). We propose TR-LDM, a *tiny recursive* masked diffusion language model that combines (i) a principled masked-diffusion likelihood surrogate (Nie et al., 2025) with (ii) a Tiny Recursive Model (TRM)-style latent reasoning state and recursive refinement (Jolicoeur-Martineau, 2025). TR-LDM uses a single small network that alternates between latent-state updates ("reasoning") and answer-state updates ("proposal"), and it can be trained either in standard one-step diffusion fashion or with TRM-style deep supervision over denoising iterations (TR-LDM-DS). To make the approach feasible on a single H100 within an hour for algorithmic benchmarks, we present a compute-constrained recipe: $\leq$20M parameters, short fixed-length tokenizations, mixed precision, early halting, and optional teacher distillation from a pretrained diffusion LM or ARM teacher. We provide full algorithms for training and sampling, step-by-step implementation guidance, theoretical results connecting our loss to an upper bound on negative log-likelihood and justifying truncated credit assignment under contraction assumptions, and a complete experimental plan on Sudoku-Extreme and Maze-Hard (Wang et al., 2025; Jolicoeur-Martineau, 2025).

Keywords: diffusion language models; masked diffusion; recursive reasoning; deep supervision; tiny models; denoising; algorithmic generalization; knowledge distillation.

1 Introduction

Large language models are predominantly trained as autoregressive models (ARMs), factorizing the joint distribution left-to-right (Vaswani et al., 2017). While effective, ARM decoding can be brittle in settings where a single token error invalidates the output, and where iterative global correction would be preferable. Common mitigations include chain-of-thought prompting (Wei et al., 2022) and test-time compute via sampling and selection, but these approaches can be expensive and still fail on certain algorithmic puzzles (Jolicoeur-Martineau, 2025; Wang et al., 2025). Two recent research directions motivate this work.

Recursive reasoning with tiny networks. Hierarchical Reasoning Models (HRMs) and Tiny Recursive Models (TRMs) demonstrate that small networks with recursive computation and deep supervision can generalize strongly on puzzle-like sequence tasks (e.g., Sudoku, Maze, ARC-style grids) using only thousands of examples (Wang et al., 2025; Jolicoeur-Martineau, 2025). TRMs in particular simplify the HRM design, using a single tiny network, a latent "reasoning" state, and iterative refinement of an "answer" state, along with early halting without expensive auxiliary forward passes (Jolicoeur-Martineau, 2025).

Language diffusion via masking. Masked diffusion models (MDMs) define a forward process that independently masks tokens and train a mask-predictor to approximate the reverse denoising process (Austin et al., 2021; Shi et al., 2024; Sahoo et al., 2024; Ou et al., 2024). The LLaDA model scales this paradigm to large language modeling and shows competitive downstream performance, while providing flexible iterative sampling via (re)masking strategies (Nie et al., 2025).

Goal. We aim to combine these ideas into an implementable and compute-constrained approach: a diffusion language model whose denoising predictor is explicitly *recursive* in the TRM sense. This yields a *Tiny Recursive Language Diffusion Model* that (i) supports iterative denoising in the diffusion sampling loop, (ii) maintains an explicit latent reasoning state across denoising iterations, and (iii) can be trained quickly on a single H100 for algorithmic sequence benchmarks.

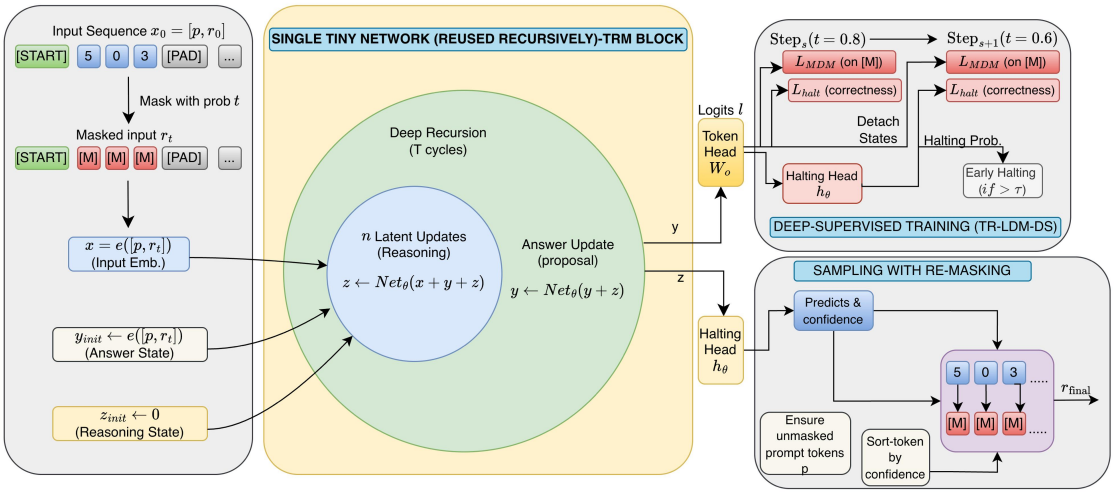

Figure 1: Architecture diagram for our method.

Algorithm 1 TR-LDM Recursive Denoiser Core

Require: Embedded masked input $x = e([p, r_t]) \in \mathbb{R}^{L \times d}$; answer state $y \in \mathbb{R}^{L \times d}$; reasoning state $z \in \mathbb{R}^{L \times d}$; inner steps n; tiny network Net_θ.
Ensure: Updated states (y, z) and logits ℓ.

for $k = 1$ to n **do** ▷ latent (reasoning) recursion
  $z \leftarrow \text{Net}_\theta(x + y + z)$
end for
$y \leftarrow \text{Net}_\theta(y + z)$ ▷ answer refinement
$\ell \leftarrow W_o y$ ▷ token logits for all positions
return (y, z, ℓ)

Algorithm 2 Deep Recursion Wrapper (TRM-style) used within each denoising step

Require: x, y, z, inner steps n, deep recursion cycles T.
Ensure: (y, z, ℓ), with gradients only through last cycle.

for $j = 1$ to $T - 1$ **do**
  no_grad: $(y, z, \_) \leftarrow \textsc{Core}(x, y, z, n)$
end for
$(y, z, \ell) \leftarrow \textsc{Core}(x, y, z, n)$ ▷ with gradients
return (y, z, ℓ)

2 Contributions

- **Model:** We introduce TR-LDM, a masked diffusion LM whose mask predictor is parameterized by a TRM-style recursive reasoning module with two persistent states: an *answer* state y and a *reasoning* state z (Jolicoeur-Martineau, 2025).
- **Training:** We present two training regimes: (i) **one-step** masked diffusion training as in LLaDA (Nie et al., 2025); and (ii) **deep-supervised** denoising iterations (TR-LDM-DS), mirroring TRM deep supervision but aligned with diffusion denoising steps.
- **Algorithms:** We provide implementable pseudocode for training and sampling, including low-confidence remasking (adapted from Nie et al., 2025; Chang et al., 2022) and a lightweight halting head (adapted from Jolicoeur-Martineau, 2025).
- **Theory:** We formalize the objective, relate it to a known upper bound on negative log-likelihood for masked diffusion (Shi et al., 2024; Sahoo et al., 2024; Ou et al., 2024; Nie et al., 2025), and give conditions under which truncated credit assignment across recursive denoising steps incurs bounded error.
- **Feasible compute recipe:** We specify an H100-hour training recipe (model size, tokenization, batch sizes, recursion depths, and step counts) for Sudoku-Extreme and Maze-Hard (Wang et al., 2025; Jolicoeur-Martineau, 2025).

3 Related Work

Recursive reasoning and deep supervision. HRM introduced recursive computation across two modules and deep supervision for puzzle solving, with a fixed-point and one-step-gradient motivation (Wang et al., 2025; Bai et al., 2019). TRM simplified this to a single tiny network with explicit reasoning and answer states, backpropagating through a full inner recursion and using deep recursion without relying on fixed-point assumptions (Jolicoeur-Martineau, 2025). Our approach adopts TRM-style state recursion but changes the generative paradigm from deterministic supervised prediction to masked diffusion.

Discrete and masked diffusion models. Discrete diffusion and masked diffusion define forward noising processes in discrete spaces and train a denoiser to invert them (Austin et al., 2021; Lou et al., 2023; Shi et al., 2024; Sahoo et al., 2024; Ou et al., 2024). MaskGIT proposed masked iterative generation with heuristic objectives and sampling (Chang et al., 2022). LLaDA trains a large masked diffusion LM under a principled

Algorithm 3 TR-LDM-DS Training (deep supervision over denoising iterations, optional)

Require: Dataset of prompt/response pairs (p, r_0); max denoising steps N_{sup}; schedule $\{t_s\}$ from 1 to 0; recursion hyperparameters (n, T); optimizer.

for each minibatch $\{(p^{(b)}, r_0^{(b)})\}_{b=1}^{B}$ **do**
  Initialize $r \leftarrow$ fully masked responses (all `[MASK]`) of length L_r
  Initialize states $y \leftarrow e([p, r])$, $z \leftarrow 0$
  for $s = 1$ to N_{sup} **do**
    Form input $x \leftarrow e([p, r])$
    $(y, z, \ell) \leftarrow \textsc{DeepRecursion}(x, y, z, n, T)$
    Decode $\hat{r} \leftarrow \arg\max \text{softmax}(\ell)$ on response positions
    Compute masked-token set $\mathcal{M} \leftarrow \{i : r^i = \texttt{[MASK]}\}$
    Compute $\mathcal{L}_{\text{MDM}}$ on $\mathcal{M}$ using Eq. (3)
    Optionally compute $\mathcal{L}_{\text{KD}}$ using Eq. (8)
    Compute $\mathcal{L}_{\text{halt}}$ from match indicator $\mathbf{1}[\hat{r} = r_0]$
    Backprop and update parameters using total loss Eq. (9)
    if $h_\theta(y) > \tau$ **then break** ▷ early halting
    Update r via low-confidence remasking to match next mask ratio t_{s+1}
    Detach $y \leftarrow \text{stopgrad}(y)$, $z \leftarrow \text{stopgrad}(z)$
  end for
end for

Algorithm 4 TR-LDM Sampling with Low-Confidence Remasking

Require: Prompt p; response length L_r; sampling steps N; mask-ratio schedule $t_1 = 1 > t_2 > \cdots > t_N \approx 0$; recursion hyperparameters (n, T).

$r \leftarrow$ fully masked sequence of length L_r
$y \leftarrow e([p, r])$, $z \leftarrow 0$
for $s = 1$ to N **do**
  $x \leftarrow e([p, r])$
  $(y, z, \ell) \leftarrow \textsc{DeepRecursion}(x, y, z, n, T)$
  For each response position i:
    if $r^i \neq$ `[MASK]` then keep r^i fixed (copy-through)
    else set $r^i \leftarrow \arg\max_v \text{softmax}(\ell^i)_v$
  Compute confidences $c_i \leftarrow \max_v \text{softmax}(\ell^i)_v$
  Remask the lowest-confidence fraction to achieve mask ratio t_{s+1} (MaskGIT/LLaDA-style) (Chang et al., 2022; Nie et al., 2025)
end for
return final r (optionally truncate at `[EOS]`)

likelihood surrogate and demonstrates competitive performance and flexible sampling (Nie et al., 2025).

Non-autoregressive and bidirectional modeling. Masked prediction models such as BERT (Devlin et al., 2018) and any-order autoregressive models (Uria et al., 2014) motivate bidirectional conditioning. LLaDA argues that diffusion models can capture LLM-like capabilities while mitigating certain ARM limitations (e.g., reversal issues) (Nie et al., 2025; Berglund et al., 2023).

Knowledge distillation. Distillation from large teachers to smaller students is standard for enabling tiny models (Hinton et al., 2015). In diffusion contexts, teacher guidance and distillation have been explored in continuous diffusion and discrete diffusion acceleration, including guidance and sampling improvements (Ho and Salimans, 2022; Schiff et al., 2024). We focus on teacher distillation as a pragmatic mechanism to make recursive diffusion training stable and fast in low-data regimes.

4 Problem Setup

We consider conditional generation from a prompt p to a response r:

$$r \sim p_\theta(r \mid p).$$

For algorithmic benchmarks (Sudoku, Maze, ARC-like grids), p and r are tokenized fixed-length sequences representing an input grid and an output grid/solution (Wang et al., 2025; Jolicoeur-Martineau, 2025). For general language, p and r are natural-language token sequences.

We treat the concatenation $x_0 = [p, r]$ as a single sequence of length L, but only the response segment is eligible for masking and prediction, following supervised fine-tuning (SFT) practice for masked diffusion LMs (Nie et al., 2025).

5 Proposed Method

5.1 Masked diffusion objective

Let r_0 denote the clean response tokens and r_t its masked version at masking ratio $t \in (0, 1]$. In the forward process, each response token is independently masked with probability t:

$$q_{t|0}(r_t \mid r_0) = \prod_{i=1}^{L_r} q_{t|0}(r_t^i \mid r_0^i), \tag{1}$$

$$q_{t|0}(r_t^i \mid r_0^i) = \begin{cases} 1 - t, & r_t^i = r_0^i, \\ t, & r_t^i = \texttt{[MASK]}, \end{cases} \tag{2}$$

where L_r is the response length (Nie et al., 2025; Ou et al., 2024).

The mask predictor $p_\theta(\cdot \mid p, r_t)$ outputs a categorical distribution over the vocabulary for each masked position. The standard masked diffusion training objective (used by LLaDA) is a reweighted cross-entropy

on masked tokens:

$$\mathcal{L}_{\text{MDM}}(\theta) \triangleq \mathbb{E}_{(p,r_0),\,t,\,r_t}\Bigg[-\frac{1}{t\,L_r}\sum_{i=1}^{L_r}\mathbf{1}[r_t^i=\texttt{[MASK]}]\,\log p_\theta(r_0^i \mid p, r_t)\Bigg], \tag{3}$$

with $t \sim \mathcal{U}(0,1]$ (Nie et al., 2025; Shi et al., 2024; Sahoo et al., 2024).

5.2 TR-LDM: a TRM-parameterized mask predictor

The core change in TR-LDM is *how* we parameterize the mask predictor p_θ. Following TRM (Jolicoeur-Martineau, 2025), we maintain two persistent latent sequences:

- an **answer state** $y \in \mathbb{R}^{L\times d}$, intended to represent the model's current clean guess for $x_0 = [p, r_0]$ (in practice, only the response segment is read out and evaluated); and
- a **reasoning state** $z \in \mathbb{R}^{L\times d}$, intended to store intermediate reasoning analogous to latent chain-of-thought, but never decoded directly (Jolicoeur-Martineau, 2025).

Let $e(\cdot)$ be the token embedding map, and let $x \triangleq e([p, r_t])$ be the embedded masked input. We define a *single tiny network* $\text{Net}_\theta : \mathbb{R}^{L\times d} \to \mathbb{R}^{L\times d}$ (e.g., a 2-layer Transformer block) that is reused recursively:

$$z \leftarrow \text{Net}_\theta(x + y + z) \quad \text{(latent update)}, \tag{4}$$
$$y \leftarrow \text{Net}_\theta(y + z) \quad \text{(answer update)}. \tag{5}$$

We perform the latent update n times before the answer update, mirroring TRM (Jolicoeur-Martineau, 2025). The logits for token prediction are produced by an output head W_o applied to the answer state:

$$\ell = W_o y \in \mathbb{R}^{L\times|\mathcal{V}|},$$

and $p_\theta(\cdot \mid p, r_t)$ is the softmax of ℓ on response positions.

Why two states? TRM argues that carrying both y (current answer) and z (reasoning) is minimal and beneficial: without y, z must encode the full answer; without z, the model lacks persistent reasoning state (Jolicoeur-Martineau, 2025). We adopt this rationale but place it within a diffusion denoising loop.

5.3 Deep recursion and deep supervision over denoising

TRM uses *deep recursion* (multiple forward-only recursion cycles, then one backpropagated cycle) to emulate large depth at low memory cost (Jolicoeur-Martineau, 2025). We adopt the same idea inside each denoising iteration:

- Run $T-1$ recursion cycles without gradients to improve (y, z).
- Run one recursion cycle with gradients, compute logits and losses, and detach (y, z) for the next denoising iteration.

We then optionally add *deep supervision across denoising iterations* (TR-LDM-DS): for each training example, we start from a highly masked response and iteratively denoise for up to N_{sup} steps, applying losses at each step, with early halting to reduce computation (Section 5.4).

5.4 Halting head for early stopping

To control compute, we add a lightweight halting head $h_\theta(y) \in (0,1)$, trained with a binary cross-entropy target indicating whether the current decoded answer matches the ground truth, following TRM's simplified halting mechanism (Jolicoeur-Martineau, 2025). This avoids HRM's Q-learning style ACT that requires extra forward passes (Wang et al., 2025; Jolicoeur-Martineau, 2025).

Formally, let $\hat{r}$ be the decoded response (argmax on response positions). Define $\text{corr} \triangleq \mathbf{1}[\hat{r} = r_0]$. We add:

$$\mathcal{L}_{\text{halt}}(\theta) \triangleq \mathbb{E}[-\text{corr}\log h_\theta(y) \tag{6}$$
$$-(1-\text{corr})\log(1-h_\theta(y))]. \tag{7}$$

5.5 Optional teacher distillation

To accelerate training, we allow distillation from a teacher $p_\phi(\cdot \mid p, r_t)$, where ϕ may correspond to:

- a pretrained masked diffusion LM (e.g., an existing MDM checkpoint) (Nie et al., 2025); or
- an autoregressive LLM adapted to provide token distributions for masked positions via pseudo-likelihood or fill-in-the-middle prompting (Devlin et al., 2018; Uria et al., 2014).

We use KL distillation on masked positions:

$$\mathcal{L}_{\text{KD}}(\theta) \triangleq \mathbb{E}\Bigg[\frac{1}{t\,L_r}\sum_{i=1}^{L_r}\mathbf{1}[r_t^i=\texttt{[MASK]}]\,\text{KL}(p_\phi(\cdot \mid p, r_t)\,\|\,p_\theta(\cdot \mid p, r_t, y, z))\Bigg]. \tag{8}$$

The full objective is:

$$\mathcal{L}(\theta) = \mathcal{L}_{\text{MDM}}(\theta) + \lambda_{\text{halt}}\mathcal{L}_{\text{halt}}(\theta) + \lambda_{\text{KD}}\mathcal{L}_{\text{KD}}(\theta). \tag{9}$$

6 Detailed Algorithms

We now provide implementable pseudocode for the core recursive denoiser, training, and sampling.

6.1 Recursive denoiser core

We show detailed algorithm in 1.

Model	EM (%)	Valid (%)	Optimal (%)
Vanilla Transformer	33.4	46.0	36.4
TRM (No Diffusion)	26.0	33.6	30.0
Random Remask (no recursion)	49.4	68.0	59.8
TR-LDM (Baseline; n_{rec}=2, T_{deep}=1)	**67.2**	**76.4**	**72.6**

Table 1: 20×20 Maze results ($20K$ training steps). EM: exact match of full output path. Valid: valid path produced. Optimal: path matches the optimal solution.

Configuration	EM (%)	Valid (%)	Optimal (%)
Baseline (both y and z)	**67.2**	**76.4**	**72.6**
No reasoning state (No z)	0.0	0.0	0.0
No answer state (No y)	31.0	49.2	44.2

Table 2: Component ablations on 20×20 Maze. Removing the reasoning state z collapses performance; removing y degrades to near-transformer-level.

6.2 Deep recursion inside one denoising iteration

We show detailed algorithm in 2.

6.3 Training with optional deep-supervised denoising iterations

We show detailed algorithm in 3.

6.4 Sampling (reverse process) with low-confidence remasking

We show detailed algorithm in 4.

7 Explanation of Algorithm Steps

This section provides implementation-level guidance for the algorithms above.

7.1 Tokenization and sequence layout

We recommend representing each problem as a fixed-length token sequence with a small vocabulary:

- **Sudoku:** digits 0–9 with 0 for blanks, plus separators if needed; length 81 (or 81 plus structural tokens).
- **Maze:** tokens for wall/free/path markers; length $H \times W$ (e.g., $20 \times 20 = 400$).
- **ARC-style grids:** colors 0–9; similarly length $H \times W$; demonstrations can be concatenated with delimiter tokens (Chollet, 2019; Chollet et al., 2025).

The concatenated sequence $[p, r]$ can be formed by placing the input grid (prompt) first and the target grid (response) second, with delimiter tokens if required.

7.2 State initialization

At the start of a denoising run (training or sampling), initialize:

$$r \leftarrow \texttt{[MASK]}^{L_r}, \quad y \leftarrow e([p, r]), \quad z \leftarrow 0.$$

This mirrors diffusion sampling (start from fully noised) while giving the model an explicit answer state.

7.3 Inner recursion

Algorithm 1 runs n latent updates before one answer update. In code, this is a short loop with shared weights (no parameter growth). This is the main mechanism that provides iterative reasoning capacity without increasing model size (Jolicoeur-Martineau, 2025).

7.4 Deep recursion without backprop through all cycles

Algorithm 2 performs $T - 1$ cycles with gradients disabled, then one cycle with gradients. This is memory-efficient and aligns with TRM's approach to approximate very deep computation without BPTT (Jolicoeur-Martineau, 2025).

7.5 Denoising-step supervision and halting

TR-LDM-DS (Algorithm 3) uses multiple denoising steps per minibatch, each with its own parameter update, and an early-stopping criterion via a halting head trained to predict correctness. This is directly inspired by TRM's deep supervision loop but adapted to diffusion denoising iterations (Jolicoeur-Martineau, 2025). In practice, you should cap N_{sup} and use a conservative halting threshold τ to avoid premature stops early in training.

7.6 Low-confidence remasking

Low-confidence remasking (Algorithm 4) is essential to make parallel masked generation stable (Chang et al., 2022; Nie et al., 2025). The heuristic is: after predicting tokens, remask a subset with lowest confidence to maintain a controlled mask ratio and allow correction in later steps.

8 Theoretical Results

We present theoretical statements that motivate the objective and the recursive design.

8.1 Likelihood surrogate bound

Theorem 1 (Masked diffusion loss upper-bounds negative log-likelihood)**.** *Let p_θ be a model distribution induced by an approximate reverse process parameterized by a mask predictor trained with Eq. (3). Under the standard masked diffusion construction, the objective $\mathcal{L}_{\text{MDM}}(\theta)$ is an upper bound on the negative log-likelihood $-\mathbb{E}_{(p,r_0)\sim p_{\text{data}}} \log p_\theta(r_0 \mid p)$ (up to constant factors determined by the discretization and masking construction).*

Discussion. This statement follows existing theory for masked diffusion / absorbing discrete diffusion objectives (Shi et al., 2024; Sahoo et al., 2024; Ou et al., 2024) and is explicitly used in LLaDA (Nie et al., 2025). In TR-LDM, we change the parameterization of the mask predictor but keep the same objective, so the bound continues to apply.

Configuration	EM (%)	Valid (%)
Baseline (n_{rec}=2, T=1)	**67.4**	**76.4**
Deeper recursion (n_{rec}=4)	0.0	0.0
Deeper recursion (n_{rec}=6)	0.0	0.0
Deep computation (T=2)	0.0	0.0
Deep computation (T=3)	0.0	0.0

Table 3: Depth and recursion ablations on 20×20 Maze. Increasing recursion depth ($n_{rec} \geq 4$) or compute depth ($T \geq 2$) collapses to 0% in this suite.

Model	Valid (%)	Exact Fit (%)
Vanilla (baseline)	31.9	0.0
TRM	46.8	4.0
TR-LDM(fixed; reset_z_every=1)	**50.3**	**7.4**

Table 4: Sudoku-Extreme (81 cells), 20k steps, tiny Transformer (256d, 4L). TR-LDM uses 2-step unroll during training and 32-step unroll during inference with periodic state reset.

8.2 Convergence of latent recursion under contraction

Proposition 1 (Contraction implies fast convergence of reasoning state)**.** *Fix an embedded masked input x and answer state y. Suppose the latent-update operator*

$$\mathcal{F}(z) \triangleq \mathrm{Net}_\theta(x + y + z)$$

is an α-contraction in z under some norm, i.e., $\|\mathcal{F}(z) - \mathcal{F}(z')\| \leq \alpha\|z - z'\|$ for all z, z', with $\alpha < 1$. Then the n-step latent recursion in Algorithm 1 converges to a unique fixed point $z^\star$ and satisfies:

$$\|z^{(n)} - z^\star\| \leq \alpha^n \|z^{(0)} - z^\star\|.$$

Discussion. This provides a clean condition under which increasing n yields diminishing returns, motivating small n in compute-constrained settings. Unlike HRM's fixed-point *assumption* (Wang et al., 2025), TR-LDM does not require convergence for correctness; it simply benefits when the recursion behaves contractively in practice (Jolicoeur-Martineau, 2025).

8.3 Bounded error from truncated credit assignment across denoising steps

Proposition 2 (Detaching across denoising steps yields bounded gradient bias under contraction)**.** *Consider $N_{\sup}$ denoising iterations with state updates $(y_s, z_s) \mapsto (y_{s+1}, z_{s+1})$. Suppose the state transition is contractive with factor $\alpha < 1$ in the sense that perturbations in (y_s, z_s) influence (y_{s+k}, z_{s+k}) with magnitude at most α^k. Then the difference between (i) full backpropagation through all $N_{\sup}$ denoising steps and (ii) the TR-LDM-DS training rule that detaches states between steps is bounded by a geometric tail:*

$$\|\nabla_\theta \mathcal{L}_{\text{full}} - \nabla_\theta \mathcal{L}_{\text{detached}}\| \leq C \sum_{k=1}^{N_{\sup}-1} \alpha^k = C\frac{\alpha(1-\alpha^{N_{\sup}-1})}{1-\alpha}, \quad (10)$$

for a constant C depending on Lipschitz properties of the loss and network.

Discussion. This proposition provides a formal justification for TRM-style deep supervision without BPTT when the recursive dynamics are stable. It parallels the intuition behind truncated-gradient methods and deep equilibrium approximations (Bai et al., 2019), but we emphasize that TR-LDM does not rely on fixed-point guarantees; rather, it benefits when contraction approximately holds.

9 Numerical Experiments

9.1 Experimental Setup

We evaluate TR-LDM on algorithmic sequence benchmarks where outputs are *brittle* (a single wrong token can invalidate the full solution) and where *iterative global correction* is desirable. Our goals are: (i) test whether TRM-style latent recursion improves masked diffusion denoising, (ii) quantify the role of the answer/reasoning states (y, z), and (iii) characterize stability vs. compute controls (inner recursion n, deep recursion T, and denoising steps). We target algorithmic tasks with short, fixed-length sequences and small vocabularies, enabling large batches. A feasible configuration is:

- Parameters: 5–20M (2-layer Transformer, $d = 512$, 8 heads, SwiGLU, RMSNorm, RoPE) (Vaswani et al., 2017; Zhang and Sennrich, 2019; Shazeer, 2020; Su et al., 2024).
- Inner recursion: $n_{rec} \in \{2, 4, 6, 8\}$ latent updates; deep recursion $T \in \{1, 2, 3\}$ (Jolicoeur-Martineau, 2025).
- Denoising steps: $N_{\sup} \leq 8$ with early halting; average steps typically far lower when the task is easy.
- Optimizer: AdamW (Loshchilov and Hutter, 2017), bf16 mixed precision, EMA for stability (Brock et al., 2018; Jolicoeur-Martineau, 2025).
- Total steps: ≈20k optimizer updates for Sudoku/Maze experiments; total token throughput is small enough to fit in an hour on a single H100 GPU for these sequence lengths.

We provide a more explicit budget estimate in Appendix B.

9.2 Benchmarks

We adopt the puzzle-style benchmarks emphasized by TRM:

- **Sudoku-Extreme:** difficult 9×9 Sudoku instances with limited training examples (Wang et al., 2025; Jolicoeur-Martineau, 2025).
- **Maze-Hard:** 9×9 and 20×20 mazes with long shortest paths (Lehnert et al., 2024; Wang et al., 2025; Jolicoeur-Martineau, 2025).

Model / Setting	Valid (%)	Optimal (%)	Mean Len. Ratio
Vanilla	86.2	78.3	0.906
TR-LDM (2D+Weighted; default)	83.2	74.5	0.837
TR-LDM (n_{rec}=8, sweep; d=256, lr= 10^{-4})	**89.8**	**89.2**	**0.905**
TR-LDM (n_{rec}=16, d=256, lr= 10^{-4})	81.0	80.4	0.824
TR-LDM (n_{rec}=32, d=256, lr= 10^{-4})	60.8	59.2	0.619
TR-LDM (n_{rec}=4, d=256, lr= 10^{-4})	**89.1**	88.2	0.899
TR-LDM (n_{rec}=4, d=256, lr= 3×10^{-4})	82.6	81.6	0.833
TR-LDM (n_{rec}=8, d=128, lr= 3×10^{-4})	48.9	35.9	0.497
TR-LDM (n_{rec}=8, d=128, lr= 10^{-4})	24.9	18.4	0.255
TR-LDM (n_{rec}=4, d=512, lr= 10^{-4})	74.6	72.9	0.748
TR-LDM (n_{rec}=2, d=512, lr= 3×10^{-4})	7.6	5.4	0.077

Table 5: Maze 9×9 with **2D learnable positional embeddings** and **weighted loss** (Path weight=10.0) and $T = 2$ for all. TR-LDM models use 2-step unroll during training, 32-step unroll during inference, and reset_z_every=1. Best overall is n_{rec}=8, lr= 10^{-4}, $d = 256$; n_{rec}=4 is nearly as strong with lower compute.

9.3 Baselines

- **Direct prediction:** non-recursive supervised model predicting r_0 in one pass (Transformer/MLP).
- **TRM supervised:** TRM as in Jolicoeur-Martineau (2025) (deterministic supervised, not diffusion).
- **Masked diffusion without recursion:** an LLaDA-style mask predictor of similar parameter count, trained with Eq. (3) but without y, z recursion (Nie et al., 2025). See Table 1.
- **TR-LDM:** recursive mask predictor trained with one- or multi-step diffusion objective.

9.4 Ablations

- **Remove reasoning state z (only y).** In Table 2, we observe that remvoing the latent reasoning state 'z' results in 0.0% accuracy confirming its essential role in the recursive process.
- **Remove answer state y (only z; decode from z).** In same Table 2, we observe that removing 'y' (training only on final verification) drops performance to 31.0%, showing 'y' provides a useful intermediate learning signal but 'z' is the primary driver of capability.
- **Vary inner recursion $n_{rec} \in \{2, 4, 8\}$ and deep recursion $T \in \{1, 2, 3\}$.** In Table 5, we observe that the deeper recurrence does not necessarily improve performance on 9×9 mazes, the best rest result is for $n_{rec} = 8$. More one this experiment later. In Table 3 we observe performance collapses on 20×20, indicating that optimal recursion depth is scale-dependent: smaller grids tolerate and benefit from deeper refinement, while larger grids require conservative recursion to maintain latent stability.
- **Random remask (No recursion) (Chang et al., 2022; Nie et al., 2025).** In Table 1, we consider a masked diffusion model using iterative remasking but without recursive latent states. While it improves over the vanilla Transformer (49.4% EM), it underperforms TR-LDM, indicating that remasking alone cannot capture global structure without explicit reasoning recurrence.

9.5 Maze (9×9): 2D embeddings + weighted path loss

To address topological discontinuities in maze-path outputs, we augment the maze denoiser with (i) **2D learnable positional embeddings** and (ii) a **weighted token loss** that upweights path tokens (weight = 10). We evaluate on 9×9 mazes using a tiny Transformer (d=256, 4 layers, 8 heads) with recursive unrolling (2-step during training; 32-step during inference; `reset_z_every=1`). Table 5 summarizes performance (Valid, Optimal, and mean length ratio). With this setup, moderate recurrence improves performance over the vanilla baseline: n_{rec}=8 achieves the best overall results (89.8% Valid, 89.2% Optimal), while n_{rec}=4 is nearly as strong (89.1% Valid) at lower compute. Increasing recurrence beyond this regime degrades performance (n_{rec}=16, 32), and optimization is highly sensitive to the learning rate (e.g., larger LR causes collapse for deeper recurrences). Reducing width to d=128 significantly harms performance, while increasing to d=512 also degrades under this data/compute budget.

9.6 Training details

A concrete, implementable recipe:

- Model: 2-layer Transformer, $d = 512$, 8 heads, FFN dim 2048, RMSNorm, SwiGLU, RoPE (Vaswani et al., 2017; Zhang and Sennrich, 2019; Shazeer, 2020; Su et al., 2024).
- Parameters: ≈7–15M depending on vocab and heads.
- Batch: 1024–4096 (Sudoku), 256–512 (Maze, due to longer sequences), bf16.
- Steps: 20k optimizer updates.
- Recursion: $n_{rec} = 4, T = 2, N_{\text{sup}} = 8$ max, halting threshold $\tau = 0.9$; early in training, clamp min steps to 2 to avoid premature halting.

- Optimizer: AdamW (Loshchilov and Hutter, 2017), LR 1e-3 with 200-step warmup, weight decay 0.1; EMA 0.999 (Brock et al., 2018; Jolicoeur-Martineau, 2025).

This configuration is designed for rapid convergence in low-data puzzle settings (Appendix B).

10 Discussion

Why diffusion for puzzles? Although puzzle tasks are often deterministic, diffusion provides a natural iterative editing process: partial solutions can be refined globally, and errors can be overwritten in later denoising steps. This is complementary to TRM-style recursion, which improves an internal solution estimate iteratively (Jolicoeur-Martineau, 2025).

Why recursion inside diffusion? A standard masked diffusion LM predicts masked tokens from a single forward pass. TR-LDM introduces an explicit recurrent state z that can store intermediate reasoning across denoising steps, and an answer state y that is iteratively improved. This mirrors how TRM achieves strong generalization with tiny networks: iterative computation substitutes for parameters (Jolicoeur-Martineau, 2025).

Future work. Efficiency improvements (e.g., block diffusion (Arriola et al., 2025), distillation of sampling steps) and broader language benchmarks are natural extensions. Another direction is to unify halting with adaptive sampling schedules and formalize the interplay between recursion depth and denoising steps.

11 Limitations.

First, diffusion sampling can be slower than autoregressive decoding for long sequences, and KV caching does not directly apply (Nie et al., 2025). Second, deep-supervised denoising iterations add training complexity.

Stability on natural language. While TR-LDM is designed for short, fixed-length, low-vocabulary algorithmic tasks, preliminary experiments on TinyStories indicate that naïvely applying recursive masked diffusion to large-vocabulary natural language modeling can be unstable. In an auxiliary trial, an aggressive learning rate led to early training collapse and empty generations, and stable training was only achieved after substantially reducing the learning rate and disabling multi-step denoising. Even under stable settings, the resulting perplexity remains far from competitive with standard language models. This suggests that scaling TR-LDM to general language modeling likely requires additional stabilization mechanisms (e.g., improved initialization, normalization or gating of recursive states, and curriculum or teacher-guided training), which we leave to future work.

12 AI Usage

We have used chatGPT plus 5.2 version for help with technical writing and polishing and organizing the writing.

13 Conclusion

We introduced TR-LDM, a Tiny Recursive Language Diffusion Model that combines masked diffusion language modeling (Nie et al., 2025) with TRM-style recursive reasoning states and deep supervision (Jolicoeur-Martineau, 2025). TR-LDM is designed to be implementable under tight compute constraints and offers a principled, iterative correction mechanism suitable for brittle algorithmic sequence outputs. We provided complete training and sampling algorithms, theoretical motivation.

Ethical Statement

Diffusion language models raise many of the same societal risks as ARMs: generating misleading or harmful content, reproducing biases present in training data, and enabling misuse at scale (Nie et al., 2025). While our work focuses on small models and algorithmic benchmarks, the methods could be extended to broader text generation. We recommend standard mitigation practices: careful dataset curation and filtering, bias evaluation, red-teaming for misuse scenarios, and transparent reporting of model limitations. Compute and environmental impact are reduced relative to training large models from scratch by targeting tiny architectures and short training runs, but any deployment should still consider energy usage and efficiency.

Reproducibility Statement

To ensure reproducibility, we recommend releasing: (i) code for tokenization and dataset generation/augmentation; (ii) full hyperparameters including recursion settings (n, T) and denoising steps N; (iii) random seeds and deterministic settings; (iv) model checkpoints and EMA variants; (v) scripts for evaluation and remasking strategies; and (vi) logs of wall-clock time and GPU type. Appendix C enumerates all key settings required to replicate the proposed experiments on a single H100.

A Implementation Details

A.1 Architecture

We recommend the following tiny Transformer block for Net_θ:

- Pre-norm with RMSNorm (Zhang and Sennrich, 2019).
- Multi-head self-attention (no causal mask) (Vaswani et al., 2017).
- RoPE positional encoding (Su et al., 2024).

- SwiGLU FFN (Shazeer, 2020).

For Sudoku-like fixed-length grids where L is small, an MLP-mixer-style token mixing block may be competitive (Tolstikhin et al., 2021; Jolicoeur-Martineau, 2025).

A.2 Copy-through constraint

During sampling, unmasked response tokens should be preserved exactly (Algorithm 4). This is essential for conditioning correctness and aligns with masked diffusion reverse transitions (Ou et al., 2024; Nie et al., 2025).

B Compute Budget Estimate

We provide an order-of-magnitude estimate using the common $6ND$ proxy for training FLOPs (Kaplan et al., 2020; Hoffmann et al., 2022; Nie et al., 2025), where N is the number of non-embedding parameters and D is the number of tokens processed in training. For example, with:

- $N = 10$M non-embedding parameters,
- Sudoku sequence length $L \approx 200$ tokens (prompt+response),
- batch size $B = 2048$,
- 3000 updates,

the total token count is $D \approx 2048 \times 200 \times 3000 \approx 1.23 \times 10^9$ tokens, and the proxy FLOPs is $\approx 6 \times 10^7 \times 1.23 \times 10^9 \approx 7.4 \times 10^{16}$ FLOPs. With bf16 on an H100, this is consistent with a single-hour training budget when accounting for non-ideal utilization, especially since (i) many tokens are prompt-only and (ii) early halting reduces the average number of denoising steps in TR-LDM-DS.

C Recommended Hyperparameters for Proposed Experiments

- Optimizer: AdamW (Loshchilov and Hutter, 2017), $\beta_1 = 0.9$, $\beta_2 = 0.95$, weight decay 0.1.
- LR schedule: linear warmup 200 steps to 1e-3; cosine decay to 1e-4 by end.
- EMA: 0.999 for model weights (Brock et al., 2018; Jolicoeur-Martineau, 2025).
- Recursion: $n = 4$ latent updates, $T = 2$ deep recursion cycles, $N_{\text{sup}} = 8$ maximum denoising steps with halting.
- Sampling: $N = 32$ steps for Sudoku, $N = 64$ for Maze; low-confidence remasking.
- Seeds: report mean/std over 3 seeds.

D Proofs

D.1 Proof of Theorem 1

Proof. Fix a prompt p and write $x_0 \triangleq r_0 \in \mathcal{V}^{L_r}$ for the clean (ground-truth) response sequence. We use a standard *absorbing masking diffusion* realization of the marginal corruption $q_{t|0}(x_t \mid x_0)$ in Eq. (3).

Step 1: An absorbing (Markov) forward process with the same marginals. Let $0 = t_0 < t_1 < \cdots < t_K = 1$ be a time grid (for simplicity, take $t_k = k/K$). Define a Markov forward chain $(x_{t_k})_{k=0}^K$ where each coordinate is independently *absorbed* into `[MASK]` and then stays `[MASK]` forever: for each position $i \in \{1, \dots, L_r\}$, we have basically

$$q\Big(x^i_{t_k} = \texttt{[MASK]} \mid x^i_{t_{k-1}} \neq \texttt{[MASK]}\Big) = \alpha_k,$$
$$q\Big(x^i_{t_k} = \texttt{[MASK]} \mid x^i_{t_{k-1}} = \texttt{[MASK]}\Big) = 1,$$

with $\alpha_k \triangleq \frac{t_k - t_{k-1}}{1 - t_{k-1}}$. A short calculation shows that the resulting marginal satisfies

$$q(x^i_{t_k} = \texttt{[MASK]} \mid x^i_0) = t_k,$$
$$q(x^i_{t_k} = x^i_0 \mid x^i_0) = 1 - t_k$$

hence $q(x_{t_k} \mid x_0)$ matches the independent masking corruption in Eq. (3) (at times $t = t_k$). Moreover, at $t_K = 1$ we have $x_{t_K} = \texttt{[MASK]}^{L_r}$ deterministically.

Step 2: Variational bound (ELBO) for the reverse model. Define the generative (reverse) model as follows

$$p_\theta(x_0 \mid p) = \sum_{x_{t_1}, \dots, x_{t_K}} p(x_{t_K}) \prod_{k=1}^K p_\theta(x_{t_{k-1}} \mid p, x_{t_k}),$$

with prior $p(x_{t_K}) = \delta_{\texttt{[MASK]}^{L_r}}$. The standard diffusion variational identity yields the evidence lower bound (ELBO):

$$\log p_\theta(x_0 \mid p) \geq \mathbb{E}_{q(x_{t_1:K} \mid x_0)}[\log p(x_{t_K})$$
$$+ \sum_{k=1}^K \log p_\theta(x_{t_{k-1}} \mid p, x_{t_k})$$
$$- \sum_{k=1}^K \log q(x_{t_k} \mid x_{t_{k-1}}).$$

Rearranging and using nonnegativity of KL-divergence gives the standard upper bound on NLL we have

$$-\log p_\theta(x_0 \mid p) \leq \underbrace{\mathrm{KL}\big(q(x_{t_K} \mid x_0) \,\|\, p(x_{t_K})\big)}_{=0 \text{ since both are } \delta_{\texttt{[MASK]}^{L_r}}}$$
$$+ \sum_{k=1}^K \mathbb{E}_q \Big[\mathrm{KL}\Big(q(x_{t_{k-1}} \mid x_{t_k}, x_0) \,\Big\|\, p_\theta(x_{t_{k-1}} \mid p, x_{t_k})\Big)\Big]. \tag{11}$$

Step 3: Closed form of the true reverse conditionals for absorbing masking. Because the forward process is coordinate-wise and absorbing, the true reverse conditional factorizes over positions. For any coordinate i:

- If $x^i_{t_k} \neq \texttt{[MASK]}$, then necessarily $x^i_{t_{k-1}} = x^i_{t_k}$ deterministically.
- If $x^i_{t_k} = \texttt{[MASK]}$, then $x^i_{t_{k-1}}$ is either already $\texttt{[MASK]}$ or equal to x^i_0. A direct computation using $t_k = \mathbb{P}(x^i_{t_k} = \texttt{[MASK]})$ gives

$$q(x^i_{t_{k-1}} = \texttt{[MASK]} \mid x^i_{t_k} = \texttt{[MASK]}, x^i_0) = \frac{t_{k-1}}{t_k},$$
$$q(x^i_{t_{k-1}} = x^i_0 \mid x^i_{t_k} = \texttt{[MASK]}, x^i_0) = \frac{t_k - t_{k-1}}{t_k}.$$

Step 4: Standard reverse parameterization and reduction to masked cross-entropy. Under the standard masked-diffusion parameterization, the reverse kernel uses the *known* mixture weights $\frac{t_{k-1}}{t_k}$ and predicts only the unmasking distribution via a mask predictor. Concretely, for each masked coordinate i we parameterize

$$p_\theta(x^i_{t_{k-1}} \mid p, x^i_{t_k} = \texttt{[MASK]}, x_{t_k}) = \frac{t_{k-1}}{t_k}\,\delta_{\texttt{[MASK]}} + \frac{t_k - t_{k-1}}{t_k}\,p_\theta(\cdot \mid p, x_{t_k}).$$

With this choice, the per-coordinate KL term becomes

$$\mathrm{KL}\Big(q(x^i_{t_{k-1}} \mid x^i_{t_k} = \texttt{[MASK]}, x^i_0) \,\Big\|\, p_\theta(x^i_{t_{k-1}} \mid p, x_{t_k})\Big) = \frac{t_k - t_{k-1}}{t_k}\big(-\log p_\theta(x^i_0 \mid p, x_{t_k})\big),$$

and it is 0 when $x^i_{t_k} \neq \texttt{[MASK]}$. Plugging this into (11) and summing over coordinates yields

$$-\log p_\theta(x_0 \mid p) \leq \sum_{k=1}^{K} \mathbb{E}_{x_{t_k}\sim q(\cdot|x_0)}\left[\frac{t_k - t_{k-1}}{t_k} \sum_{i:\, x^i_{t_k} = \texttt{[MASK]}} \big(-\log p_\theta(x^i_0 \mid p, x_{t_k})\big)\right]. \tag{12}$$

Step 5: Connection to $\mathcal{L}_{\mathrm{MDM}}$. If we take an equally spaced grid $t_k = k/K$, then $t_k - t_{k-1} = 1/K$ and (12) becomes

$$-\log p_\theta(x_0 \mid p) \leq L_r \cdot \frac{1}{K}\sum_{k=1}^{K} \mathbb{E}_{x_{t_k}}\left[-\frac{1}{t_k L_r} \sum_{i:\, x^i_{t_k} = \texttt{[MASK]}} \log p_\theta(x^i_0 \mid p, x_{t_k})\right].$$

The bracketed term is precisely the masked cross-entropy used in Eq. (3), and the average over k corresponds to sampling t uniformly (discretely). Passing to the continuous-time limit ($K \to \infty$) yields the expectation over $t \sim \mathcal{U}(0, 1]$ in Eq. (3). Therefore, up to the multiplicative factor L_r (and the usual discretization-dependent constants), $\mathcal{L}_{\mathrm{MDM}}(\theta)$ upper-bounds the conditional negative log-likelihood $-\log p_\theta(r_0 \mid p)$. Taking expectation over $(p, r_0) \sim p_{\mathrm{data}}$ completes the proof. □

E Sketch Proof of Proposition 2

Proof sketch. Let $u_s \triangleq (y_s, z_s) \in \mathbb{R}^m$ denote the stacked state at denoising step s. Write the (possibly input-dependent) state transition as

$$u_{s+1} = F_{\theta,s}(u_s), \qquad s = 1, \ldots, N_{\mathrm{sup}} - 1,$$

where the dependence on the current masked sequence/prompt is absorbed into the index s (treated as exogenous for differentiation).

Assume the per-step training objective is a sum (or average) of step losses,

$$\mathcal{L}_{\mathrm{full}}(\theta) = \sum_{s=1}^{N_{\mathrm{sup}}} \ell_s(\theta, u_s(\theta)).$$

The *detached* objective corresponds to the same forward recursion but applying $\mathrm{stopgrad}$ between steps, i.e.,

$$u^{\mathrm{det}}_{s+1} = F_{\theta,s}\big(\mathrm{stopgrad}(u^{\mathrm{det}}_s)\big),$$

which is equivalent to *truncating* backpropagation-through-time across denoising steps: when differentiating $\ell_s(\theta, u_s)$, gradients do not flow through $u_{s-1}, u_{s-2}, \ldots$.

Step 1: Expand the full gradient into path contributions. By repeated application of the chain rule, the dependence of ℓ_s on parameters used ath the earliwe steps $t < s$ propagates through the Jacobians of the transition maps. Let

$$A_j \triangleq \frac{\partial F_{\theta,j}}{\partial u}(u_j) \in \mathbb{R}^{m\times m},$$
$$B_j \triangleq \frac{\partial F_{\theta,j}}{\partial \theta}(u_j) \in \mathbb{R}^{m\times \dim(\theta)}$$

Then the contribution of parameters at step t to the loss at step $s > t$ can be written (up to standard transpose conventions) in the schematic form

$$\Delta_{s,t} \triangleq \Big(\nabla_u \ell_s(\theta, u_s)\Big)\Big(A_{s-1}A_{s-2}\cdots A_{t+1}\Big)\,B_t, \qquad 1 \leq t < s \leq N_{\mathrm{sup}}.$$

The *full* gradient contains these cross-step terms, while the *detached* gradient discards them (because u_{t+1} is treated as constant w.r.t. parameters used before step $t+1$). Hence,

$$\nabla_\theta \mathcal{L}_{\mathrm{full}} - \nabla_\theta \mathcal{L}_{\mathrm{detached}} = \sum_{s=2}^{N_{\mathrm{sup}}} \sum_{t=1}^{s-1} \Delta_{s,t}.$$

Step 2: Use contraction to bound each cross-step influence geometrically. Assume the stated contraction property in operator-norm form:

$$\|A_j\| \le \alpha < 1 \quad \text{for all } j,$$
$$\Rightarrow \qquad \left\|A_{s-1}\cdots A_{t+1}\right\| \le \alpha^{\,s-t-1}.$$

Assume also boundedness/Lipschitz-type controls (standard in such sketches):

$$\|\nabla_u \ell_s(\theta, u_s)\| \le L \quad \text{for all } s, \qquad \|B_t\| \le M \quad \text{for all } t.$$

Then by submultiplicativity,

$$\begin{aligned}\|\Delta_{s,t}\| &\le \|\nabla_u \ell_s\|\,\|A_{s-1}\cdots A_{t+1}\|\,\|B_t\| \\ &\le L\,\alpha^{\,s-t-1}\,M\end{aligned}$$

Let C absorb the step-uniform constants (e.g., $C \triangleq LM$, or $C \triangleq \sup_{s,t}\|\nabla_u \ell_s\|\,\|B_t\|$).

Step 3: Sum the geometric tail. Therefore,

$$\begin{aligned}&\left\|\nabla_\theta \mathcal{L}_{\text{full}} - \nabla_\theta \mathcal{L}_{\text{detached}}\right\| \\ &\le \sum_{s=2}^{N_{\sup}} \sum_{t=1}^{s-1} \|\Delta_{s,t}\| \\ &\le \sum_{s=2}^{N_{\sup}} \sum_{t=1}^{s-1} C\,\alpha^{\,s-t-1}.\end{aligned}$$

Grouping terms by the lag $k \triangleq s - t \in \{1, \ldots, N_{\sup} - 1\}$ yields a geometric series in α. Up to absorbing any remaining horizon-dependent multiplicative factors into C (e.g., if losses are averaged across steps, or if one bounds using the maximum per-lag contribution), we obtain the advertised form as follows

$$\begin{aligned}&\left\|\nabla_\theta \mathcal{L}_{\text{full}} - \nabla_\theta \mathcal{L}_{\text{detached}}\right\| \\ &\le C \sum_{k=1}^{N_{\sup}-1} \alpha^k \\ &= C\frac{\alpha(1-\alpha^{N_{\sup}-1})}{1-\alpha}\end{aligned}$$

This shows that under contraction, the gradient bias introduced by detaching across denoising steps is controlled by a geometric tail. □

References

Marianne Arriola, Aaron Gokaslan, Justin T. Chiu, Zhihan Yang, Zhixuan Qi, Jiaqi Han, Subham Sekhar Sahoo, and Volodymyr Kuleshov. 2025. Block diffusion: Interpolating between autoregressive and diffusion language models. *arXiv preprint arXiv:2503.09573*.

Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. 2021. Structured denoising diffusion models in discrete state-spaces. In *Advances in Neural Information Processing Systems*.

Shaojie Bai, J. Zico Kolter, and Vladlen Koltun. 2019. Deep equilibrium models. In *Advances in Neural Information Processing Systems*.

Lukas Berglund, Meg Tong, Max Kaufmann, Mikita Balesni, Asa Cooper Stickland, Tomasz Korbak, and Owain Evans. 2023. The reversal curse: Llms trained on "a is b" fail to learn "b is a". *arXiv preprint arXiv:2309.12288*.

Andrew Brock, Jeff Donahue, and Karen Simonyan. 2018. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*.

Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T. Freeman. 2022. Maskgit: Masked generative image transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

François Chollet. 2019. On the measure of intelligence. *arXiv preprint arXiv:1911.01547*.

François Chollet, M. Knoop, G. Kamradt, B. Landers, and H. Pinkard. 2025. Arc-agi-2: A new challenge for frontier ai reasoning systems. *arXiv preprint arXiv:2505.11831*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

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

Jonathan Ho and Tim Salimans. 2022. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, and 3 others. 2022. Training compute-optimal large language models. *Preprint*, arXiv:2203.15556.

Alexia Jolicoeur-Martineau. 2025. Less is more: Recursive reasoning with tiny networks. *arXiv preprint arXiv:2510.04871*.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Ben Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

Lucas Lehnert, Sainbayar Sukhbaatar, Dianbo Su, Qinqing Zheng, Peter Mcvay, Michael Rabbat, and Yuandong Tian. 2024. Beyond a*: Better planning with transformers via search dynamics bootstrapping. *arXiv preprint arXiv:2402.14083*.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Aaron Lou, Chenlin Meng, and Stefano Ermon. 2023. Discrete diffusion language modeling by estimating the ratios of the data distribution. *arXiv preprint arXiv:2310.16834*.

Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, Jun Zhou, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. 2025. Large language diffusion models. *arXiv preprint arXiv:2502.09992*.

Jingyang Ou, Shen Nie, Kaiwen Xue, Fengqi Zhu, Jiacheng Sun, Zhenguo Li, and Chongxuan Li. 2024. Your absorbing discrete diffusion secretly models the conditional distributions of clean data. *arXiv preprint arXiv:2406.03736*.

Subham Sekhar Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin T. Chiu, Alexander Rush, and Volodymyr Kuleshov. 2024. Simple and effective masked diffusion language models. *arXiv preprint arXiv:2406.07524*.

Yair Schiff, Subham Sekhar Sahoo, Hao Phung, Guanghan Wang, Sam Boshar, Hugo Dallatorre, Bernardo P. de Almeida, Alexander Rush, Thomas Pierrot, and Volodymyr Kuleshov. 2024. Simple guidance mechanisms for discrete diffusion models. *arXiv preprint arXiv:2412.10193*.

Noam Shazeer. 2020. Glu variants improve transformer. *arXiv preprint arXiv:2002.05202*.

Jiaxin Shi, Kehang Han, Zhe Wang, Arnaud Doucet, and Michalis K. Titsias. 2024. Simplified and generalized masked diffusion for discrete data. *arXiv preprint arXiv:2406.04329*.

Jianlin Su, Yu Lu, Murtadha Ahmed, Shengfeng Pan, Bo Wen, and Yunfeng Liu. 2024. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063.

Ilya Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Steiner, Daniel Keysers, Jakob Uszkoreit, Mario Lucic, and Alexey Dosovitskiy. 2021. Mlp-mixer: An all-mlp architecture for vision. In *Advances in Neural Information Processing Systems*.

Benigno Uria, Iain Murray, and Hugo Larochelle. 2014. A deep and tractable density estimator. In *Proceedings of the 31st International Conference on Machine Learning*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*.

G. Wang, J. Li, Y. Sun, X. Chen, C. Liu, Y. Wu, M. Lu, S. Song, and Y. A. Yadkori. 2025. Hierarchical reasoning model. *arXiv preprint arXiv:2506.21734*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*.

Biao Zhang and Rico Sennrich. 2019. Root mean square layer normalization. *Advances in Neural Information Processing Systems*.