Learned Relay Representations for Forward-Thinking Discrete Diffusion Models

Benjamin Rozonoyer*[1] Jacopo Minniti*[2] Dhruvesh Patel*[1] Neil Band[3]
Avishek Joey Bose[4,5] Tim G. J. Rudner[2,6] Andrew McCallum[1]

[1] UMass Amherst [2] U Toronto [3] Stanford [4] Imperial [5] Mila [6] Vijil

❶ Problem: The Hard Reset

MDMs discard all intermediate computation between denoising steps.

Hard Reset: Each forward pass computes rich hidden states—including at still-masked positions—then **throws them away** before the next step.

How can the sequential unmasking structure of MDMs support recurrent computation that carries richer information across steps?

❷ Learned Relay Representations

A Relay channel passes hidden states across steps for the benefit of future denoising steps.

Augment the state with last-layer hidden states: $\mathbf{s}_k = (\boldsymbol{x}_{t_k}, \boldsymbol{h}_k)$, passed forward via a differentiable **relay channel**:

$$\boldsymbol{h}_{k+1} = f_\theta\big(\mathrm{Emb}(\boldsymbol{x}_{t_k}) + R_\theta(\boldsymbol{h}_k)\big)$$

with logits read out as $\boldsymbol{\ell}_k = \mathrm{UnEmb}(\boldsymbol{h}_{k+1})$.

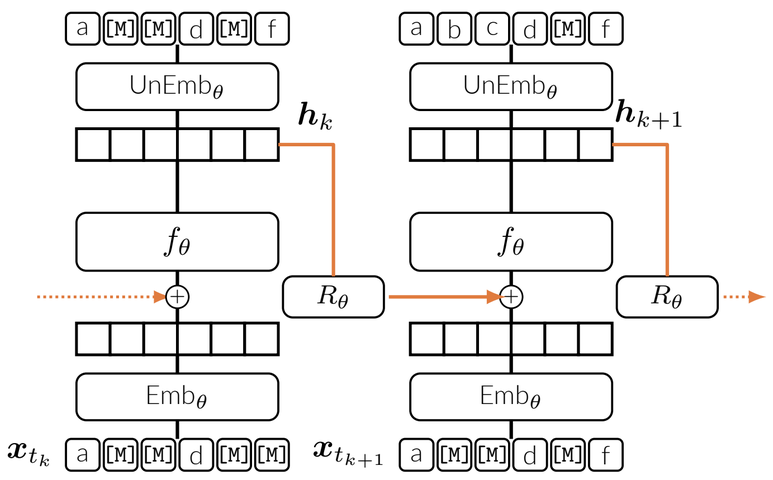


Figure 1: Relay schematic: latent state persists across denoising steps.

Architecture-agnostic: decoding is unchanged except for forwarding the relay.

❸ Training: Truncated BPTT

Gradients flow through the relay across steps; K=2 suffices for strong gains.

Algorithm 1: Relay Training

Input: $f_\theta, R_\theta, K, u, N, \eta$

for $t \in \{1, \dots, N\}$ **do**
  if $t = 1$ or $\mathcal{M}(\boldsymbol{z}) = \emptyset$ **then**
    $\boldsymbol{x}_0 \sim p_{\text{data}},\ \boldsymbol{z} \leftarrow \{\texttt{[M]}\}^L,\ \boldsymbol{h} \leftarrow \mathbf{0}$
  $L \leftarrow 0$
  for $k \in \{0, \dots, K-1\}$ **do**
    $\boldsymbol{h} \leftarrow f_\theta(\mathrm{Emb}_\theta(\boldsymbol{z}) + R_\theta(\boldsymbol{h}))$
    $\boldsymbol{\ell} \leftarrow \mathrm{UnEmb}_\theta(\boldsymbol{h}), \quad L \leftarrow L + \mathcal{L}(\boldsymbol{\ell}, \boldsymbol{x}_0)$
    $\mathcal{U} \sim u(\cdot \mid \boldsymbol{\ell}, \boldsymbol{z})$
    $z^i \leftarrow x_0^i\ \forall i \in \mathcal{U}$
  $\theta \leftarrow \theta - \eta \nabla_\theta L$
return θ

BPTT: gradients flow through $\boldsymbol{h}$ across K-step window; discrete unmasking is fixed.

A differentiable relay channel gives MDMs cross-step latent memory trainable with BPTT for temporal credit assignment.

❹ Sudoku: Progressive Ablations

Relay advances the accuracy–NFE Pareto frontier at every confidence threshold.

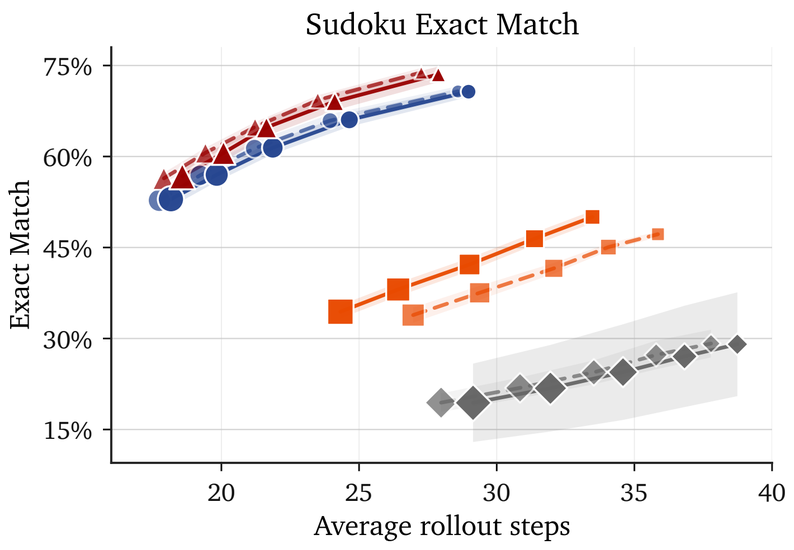
RQ1 Does BPTT improve performance *and* latency over stop-gradient?

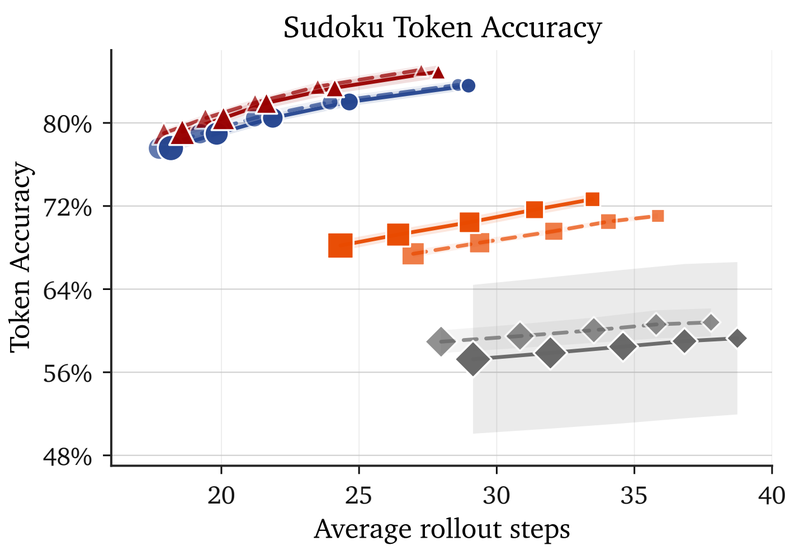
RQ2 Does weight-tying Emb$_\theta$/UnEmb$_\theta$ affect Relay, since the first layer must consume the UnEmb$_\theta$-aligned relay $\boldsymbol{h}$?

RQ3 Can state-of-the-art DLMs be efficiently adapted to use relay representations?

- ◆ MLM — Uniform masks; no rollout, no relay.
- ■ Rollout — K=2 teacher-forced unmasking; no relay.
- ● Relay (sg) — Relay but stop-gradient blocks temporal credit.
- ▲ Relay — Full method: K=2 BPTT through the relay.

Solid: tied Emb$_\theta$/UnEmb$_\theta$. *Dashed*: untied.





Method	Unfiltered Acc (%)	Unfiltered NFE	Deduction-only Acc (%)	Deduction-only NFE
MLM	20.3	13.8	32.2	9.3
Rollout	38.7	12.5	52.9	10.5
Relay (sg)	58.4	7.6	70.8	6.1
Relay	62.7	7.4	76.4	5.9

Tied Emb$_\theta$/UnEmb$_\theta$ weights, τ=0.15. Tying vs. untying has ≤3 pp effect on any row.

Relay also commits more cells per forward pass while keeping the partial board legal: 74.8% fully legal boards vs. 70.7% for Relay (sg) (+4.1 pp), demonstrating that BPTT teaches the relay to unmask more *aggressively and correctly*.

Relay dominates every baseline on the Sudoku accuracy–NFE frontier, with BPTT credit assignment accounting for the final accuracy lift over stop-gradient.

❺ Scaling: Fast-dLLM v2 (1.5B)

Relay surpasses vanilla SFT accuracy with 32% fewer NFEs on HumanEval.

Base model. Fast-dLLM v2 (Qwen2.5, 1.5B) — a state-of-the-art DLM with block-autoregressive decoding and KV caching.

Training. 200 SFT steps (batch 32) on a 60k-example code/math mixture (40/60 split of OpenCodeInstruct + OpenMathInstruct-2).

KV-cache adaptations. (1) Relay rollout runs *only inside the active block*, leaving frozen blocks' inter-block KV cache unchanged. (2) Relay state $\boldsymbol{h}$ is updated *only at still-masked positions*, keeping within-block sub-block KV entries valid.

Evaluation. Confidence-based parallel decoding; τ=0.85, block 32, sub-block 8. NFE counts active denoising forwards per example.

Method	HumanEval Base	HumanEval Plus	HumanEval NFE	MBPP Base	MBPP Plus	MBPP NFE
Fast-dLLM-v2	38.4	35.4	178.1	46.8	39.7	133.0
Vanilla SFT	38.4	34.1	130.7	43.9	38.1	84.8
Relay (sg)	38.4	35.4	104.4	43.1	39.2	80.1
Relay	42.1	37.2	88.3	46.6	41.5	78.8

Relay attains the best accuracy *and* lowest NFE among adapted methods on both benchmarks. On HumanEval it surpasses vanilla SFT accuracy while using 32% fewer NFEs (88 vs. 131).

❻ Memory Overhead

Truncated BPTT does not increase peak GPU memory.

Truncated BPTT does not double peak GPU memory on Fast-dLLM v2: the CE backward pass dominates in both regimes.

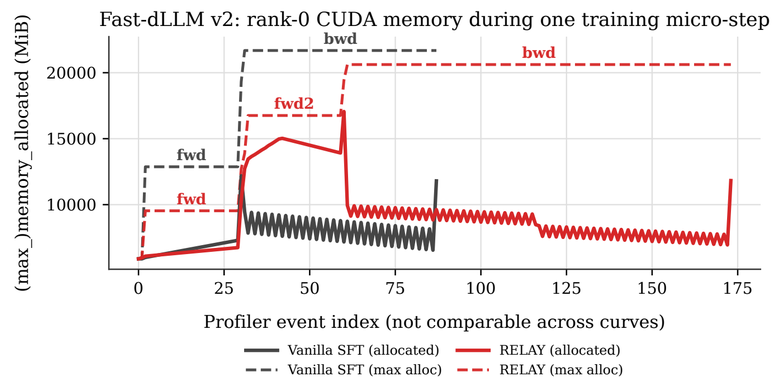


Figure 2: One training micro-step (A100): Relay peak ≈20.1 GiB vs vanilla ≈21.2 GiB.

Paper Code Blog

Relay scales to 1.5B: architecture-agnostic drop-in for MDMs/DLMs, compatible with block diffusion + KV caching, and no extra GPU memory cost.