

Pivot-SD: Efficient Self-Distillation for Masked Diffusion Language Models

Seo Hyun Kim^{1,2,*‡} Sunwoo Hong^{1,2,*‡} Younwoo Choi² Chen-Hao Chao²
Se-Young Yun^{1,†} Rahul G. Krishnan^{2,†}

¹KAIST AI ²University of Toronto & Vector Institute
{shkimsally, sunwoo}@kaist.ac.kr

Abstract

Masked diffusion language models (dLMs) offer a promising parallel alternative to autoregressive models for complex reasoning. However, they face a unique credit-assignment challenge, since reasoning failures typically stem from a few critical, early commitments that cascade downstream. Existing post-training overlooks this dynamic: supervised fine-tuning (SFT) weights all tokens uniformly, while on-line reinforcement learning (RL) attributes a single trajectory-level reward to every position. We introduce **Pivot-SD**, an efficient offline self-distillation framework that supervises only these high-impact bottlenecks (pivots). Pivot-SD isolates pivots using an information-gain metric measuring uncertainty reduction over the remaining masked block. It applies bidirectional supervision: successful pivots are reinforced with cross-entropy, while critical errors in failed trajectories are suppressed via targeted unlikelihood, preserving valid surrounding sub-steps. Using only 200 questions and four rollouts each, Pivot-SD improves LLaDA-8B-Instruct over full-sequence SFT and budget-matched diffusion RL baselines across math and code benchmarks.

1 Introduction

Large language models (LLMs) have achieved remarkable progress on complex reasoning tasks, largely driven by post-training methodologies such as supervised fine-tuning (SFT) and reinforcement learning (RL) (Ouyang et al., 2022; Shao et al., 2024; Guo et al., 2025). This progress has been anchored to sequential autoregressive models (ARMs), where generations unfold left-to-right. Recently, masked diffusion language models (dLMs) have emerged as a compelling parallel, coarse-to-fine alternative, relaxing the sequential

bottleneck where early mistakes inevitably propagate and collapse the reasoning trace (Sahoo et al., 2024; Ye et al., 2025; Nie et al., 2025).

While dLMs introduce a fundamentally different generation paradigm, they also present a unique credit-assignment challenge. This challenge, however, reveals a highly efficient path for training: if a trajectory’s success is dictated by a handful of decisive token commitments, it suffices to supervise these critical bottlenecks rather than the entire sequence.

To date, post-training for dLMs has largely adapted practices from autoregressive modeling (Zhao et al., 2025; Wang et al., 2025; Rojas et al., 2026; Tang et al., 2026). While ARMs support selective supervision (e.g., process reward models), they inherently equate a token’s spatial position with its temporal generation step. Masked diffusion breaks this equivalence: a commitment made early, while much of the block is still masked, constrains far more of the trajectory than the same token written late, once the surrounding context is fixed. Effective supervision in this regime must satisfy three requirements simultaneously, yet existing recipes each miss at least one. It must be *selective*, but standard SFT weights all tokens uniformly (Nie et al., 2025); refinements such as entropy-based reweighting select on the final sequence, where the step at which a token was committed is no longer visible (Xu et al., 2025). It must *learn from failure*, but rejection-based self-distillation keeps only successful traces and discards the rest (Zelikman et al., 2022; Singh et al., 2024; Dou et al., 2024). Finally, it must be *local*, but online RL attributes a single trajectory-level reward to every position, penalizing the valid syntactic structures and algebraic fragments in a failed trace alongside the decisive error (Zhao et al., 2025; Wang et al., 2025; Tang et al., 2026).

In this work, we propose a fundamental shift in perspective for dLM post-training: **not all tokens**

*Equal contribution. †Corresponding authors. ‡Work done as visiting researchers at the University of Toronto and the Vector Institute.

require uniform supervision. Within a single denoising trajectory, a small number of commitments sharply reduce the model’s uncertainty over the positions that remain masked, while most others leave it largely unchanged. These are the decisions that fix the shape of the rest of the block, and they are where the trajectory’s outcome is effectively determined. If the outcome depends on these few decisions, supervision should be placed there rather than spread across every position. This holds in both directions: a successful trajectory tells us which commitments to reinforce, and a failed one tells us where the reasoning broke, without condemning the valid fragments that surround it.

To realize this, we introduce **Pivot-SD**, an offline self-distillation framework that shifts the unit of credit assignment from final sequence tokens to structural reasoning bottlenecks. Pivot-SD employs a normalized information-gain metric to dynamically isolate candidate pivots, which are commitments that maximally reduce uncertainty over the remaining masked block. Once identified, we apply a localized update: successful pivots are reinforced with cross-entropy, while high-impact errors from failed trajectories are suppressed via targeted unlikelihood training (Welleck et al., 2019; Ni et al., 2025). Pivot-SD is highly data-efficient, requiring only 200 training questions and four rollouts per question. Under this budget it consistently improves over full-sequence SFT and budget-matched diffusion RL baselines across mathematical reasoning and code-generation benchmarks.

In summary, our main contributions are as follows:

- **Step-Conditioned Credit Assignment:** We formulate reasoning supervision for discrete masked diffusion language models around denoising commitments, fundamentally shifting the unit of credit assignment from final sequence tokens to structural reasoning bottlenecks.
- **Information-Guided Localized Self-Distillation:** We introduce Pivot-SD, a novel offline objective that selects candidate bottlenecks by measuring counterfactual uncertainty reduction. It extracts dense supervision by reinforcing successful pivots with cross-entropy and suppressing high-impact failed pivots with localized unlikelihood, preserving locally valid sub-steps.
- **Data-Efficient Offline Post-Training:** We empirically demonstrate that Pivot-SD substantially improves structured reasoning using only a minimal cached trajectory pool, consistently outperforming same-data and same-step online RL-style baselines.

2 Related Work

Diffusion language models. Diffusion models have been studied as alternatives to autoregressive language modeling. Early work explored continuous diffusion over word embeddings (Li et al., 2022), and D3PMs extended denoising diffusion to discrete state spaces (Austin et al., 2021a). Later work improved discrete and masked diffusion with score-entropy objectives and masked training recipes (Lou et al., 2024; Sahoo et al., 2024), and LLaDA scaled the paradigm to an 8B instruction-following model trained with forward masking and reverse masked-token prediction (Nie et al., 2025). These models introduce credit-assignment questions because generation unfolds over iterative denoising states rather than a fixed token order (e.g., left-to-right.)

Our work focuses on this denoising-time structure. Instead of treating a sampled response as only a final sequence of tokens, we treat the unmask trajectory itself as a source of supervision. Our central idea is that a token is important not merely because it appears in the final solution, but because it may be committed at a particular denoising step and thereby constrain subsequent masked positions.

Reasoning with masked diffusion language models. Several recent methods adapt these models for reasoning. d1 combines masked SFT with diffu-GRPO, an RL algorithm for masked diffusion (Zhao et al., 2025). DCoLT treats intermediate reverse-diffusion steps as latent thinking actions and optimizes trajectories with outcome-based RL (Huang et al., 2025). d2 proposes policy-gradient techniques that improve reasoning without supervised fine-tuning (Wang et al., 2025), and GDPO revisits likelihood estimation and uses variance-reduced ELBO-based policy optimization (Rojas et al., 2026). These methods rely on online RL. Pivot-SD instead samples trajectories once from a frozen model and converts trajectory outcomes into localized supervision over selected denoising commitments. This gives a stable offline alternative that can use both successful and failed trajectories.

Process supervision and reasoning credit assignment. A common approach to improving mathematical reasoning is to supervise intermediate steps. Lightman et al. (2024) show that step-level feedback can outperform outcome supervision on MATH, and Math-Shepherd automatically constructs process labels and trains process reward models (Wang et al., 2024). These methods score textual reasoning steps in autoregressive solutions.

Pivot-SD shares the goal of fine-grained credit assignment but changes the unit of supervision. Instead of scoring a sentence or derivation step, it supervises a denoising commitment (t, p, y_p) inside a masked diffusion trajectory. This matters because masked diffusion models commit tokens out of order, and a single commitment can reshape the distribution over many remaining positions.

Token importance and weighted fine-tuning. A related direction reweights tokens during SFT. For diffusion language models, GIFT assigns entropy-based importance weights to tokens (Xu et al., 2025), showing that emphasizing some tokens helps fine-tuning. Pivot-SD differs in what is being weighted: our scores rank tokens by their impact as commitments at specific denoising steps, not by their importance in the final sequence. To separate these effects, our ablations compare pivot-only supervision, selected-step uniform supervision, entropy-selected pivots, and random pivots.

Positioning of our work. In masked diffusion, spatial position and decision time decouple: the exact same token at the same position can be a trivial fill if committed late, yet reshape the entire trajectory if committed early. Pivot-SD leverages this decoupling by shifting the unit of credit assignment from final-sequence tokens to the realized commitment event. This captures not just *what* token is written and *where*, but *when* it is committed under a partially denoised state.

Pivot-SD applies standard objectives to a fundamentally different object. Unlike process supervision that scores textual steps (Lightman et al., 2024; Wang et al., 2024) or entropy-based SFT weighting over final sequences (Xu et al., 2025), Pivot-SD supervises the denoising commitment itself. Furthermore, rather than assigning broad rewards over denoising intervals (Chen et al., 2025; Xie et al., 2026; Oba et al., 2026) or evaluating terminal blocks (Horvitz et al., 2025), it localizes credit specifically to early, trajectory-internal bottlenecks. This allows Pivot-SD to exploit both successful

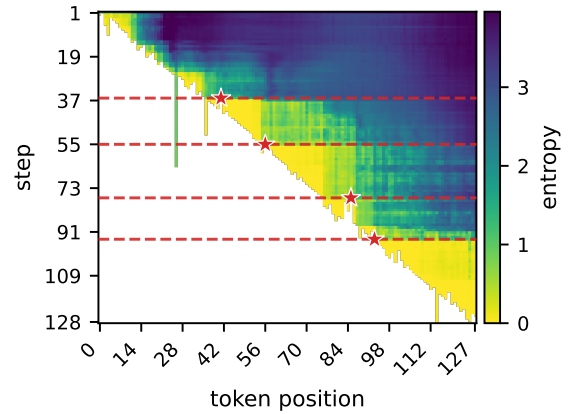


Figure 1: **Predictive Entropy and Pivot Selection.** An entropy heatmap of a masked denoising trajectory for a MATH problem (showing the first 128 tokens). The x-axis represents the token position, and the y-axis denotes the denoising step progressing downward. The color scale indicates the predictive entropy of the active masked positions. Red dashed lines and stars denote the critical denoising steps and the specific pivot tokens selected by our Information Gain metric, respectively.

and failed trajectories without broad sequence-level suppression (Table 6).

3 Methodology

In this section, we present Pivot-SD, an offline self-distillation framework for masked diffusion language models. We first establish the formal mechanics of masked diffusion language models and standard supervised fine-tuning (§3.1). We then introduce our information-gain selection metric and training objectives (§3.2). We detail the complete offline algorithm and contrast its structural properties with standard SFT and online reinforcement learning paradigms (§3.3, §3.4).

3.1 Preliminaries and Problem Setup

We begin by formalizing the notation for masked diffusion language models and establishing the baseline training objective.

Setup Let x represent a conditioning prompt, and let $y = (y_1, \dots, y_L)$ denote a target response sequence of length L . A masked diffusion language model generates y conditioned on x by iteratively denoising an initially fully masked response sequence over T steps. Let M_t represent the complete sequence state at denoising step t (consisting of the unmasked prompt x concatenated with the partially masked response), indexed in the sampler’s reverse chronological order. To formalize the

active response positions, let $\mathcal{I}_{\text{resp}}$ denote the set of coordinate indices corresponding exclusively to the response tokens. The active response positions that remain masked at step t are defined as

$$\mathcal{B}_t = \{i \in \mathcal{I}_{\text{resp}} : M_{t,i} = [\text{MASK}]\}. \quad (1)$$

At each denoising step, the model parameterizes a distribution over the clean token for each active masked position:

$$P_\theta(x_0^{(i)} \mid M_t), \quad i \in \mathcal{B}_t, \quad (2)$$

where $x_0^{(i)}$ is the clean-token random variable at response position i , and its final sampled value in the finalized trajectory is y_i .

Standard SFT Standard masked SFT trains the parameters θ by minimizing the uniform per-position cross-entropy loss:

$$\mathcal{L}_{\text{SFT}}(\theta) = -\mathbb{E}_{((x,y),t,M_t)} \left[\sum_{i \in \mathcal{B}_t} \log P_\theta(y_i \mid M_t) \right]. \quad (3)$$

The baseline loss in Eq. (3) weights every active masked position uniformly. It fails to distinguish between trivial token fills and critical structural commitments that dictate the downstream coherence of the response. Pivot-SD addresses this limitation by restricting supervision to a sparse set of highly informative step-position commitments, where the directional sign of the parameter update is gated by the overall quality of the trajectory outcome.

3.2 The Pivot-SD Framework

In this section, we introduce the core components of Pivot-SD. We outline how we extract localized, step-conditioned units of supervision from raw diffusion trajectories (§3.2.1), isolate high-impact commitments using an information-gain formulation (§3.2.2), and finally the training objective (§3.2.3).

3.2.1 Trajectory Generation and Candidate Pivots

Sampling a complete response from the frozen base model θ_0 yields a discrete denoising trajectory:

$$\tau = (M_t, \mathcal{B}_t, \mathcal{U}_t)_{t=1}^T, \quad (4)$$

where $\mathcal{U}_t \subseteq \mathcal{B}_t$ is the specific subset of positions selected for unmasking and commitment by the sampler at step t . A position p is committed at step

t when the sampler selects $p \in \mathcal{U}_t$ and writes the sampled token value y_p into the sequence state. For every committed position $p \in \mathcal{U}_t$, the trajectory generates a candidate pivot c , which we group into a trajectory-wide candidate set:

$$\mathcal{C}(\tau) = \{(t, p, y_p) : p \in \mathcal{U}_t\}. \quad (5)$$

The triple in Eq. (5) records what token was committed, where it was committed, and when in the denoising process it was committed. Thus the supervised object is not merely the token y_p , but the event of committing y_p at position p under state M_t . The same token at the same response position can be low-impact late in denoising but high-impact earlier, when many neighboring positions remain masked. This is what separates Pivot-SD from token-level SFT, which supervises only (p, y_p) , and from trajectory-level RL, which supervises only the trajectory outcome.

3.2.2 Information-Gain Pivot Selection

Pivot selection ranks candidate commitments by a one-step estimate of how much fixing that commitment reduces uncertainty over the remaining masked block. Intuitively, a useful pivot is not merely a token the model is uncertain about, it is a commitment that changes what the frozen model can infer about the rest of the denoising block. We measure local token uncertainty using the Shannon entropy of the frozen base model’s predictions. For any masked state M and index i , the per-position entropy is

$$h_{\theta_0}(i \mid M) = H \left(P_{\theta_0}(x_0^{(i)} \mid M) \right). \quad (6)$$

For a candidate commitment (t, p, y_p) , the baseline pre-commitment uncertainty across the remaining uncommitted active block is defined as

$$H_{\text{pre}}(t, p) = \sum_{i \in \mathcal{B}_t \setminus \{p\}} h_{\theta_0}(i \mid M_t). \quad (7)$$

We then evaluate a counterfactual state where position p is explicitly clamped to its sampled token value y_p , yielding a modified context sequence $M_t^{[p \leftarrow y_p]}$. The post-clamp uncertainty over the identical remaining block is calculated as

$$H_{\text{post}}(t, p) = \sum_{i \in \mathcal{B}_t \setminus \{p\}} h_{\theta_0}(i \mid M_t^{[p \leftarrow y_p]}). \quad (8)$$

By excluding the pivot position p from both terms, this formulation directly isolates how much a localized commitment reduces uncertainty across the

rest of the generation block. To prevent early denoising states, which inherently possess larger absolute entropy scales, from dominating the selection process, we compute a normalized information-gain score:

$$g(t, p) = \frac{H_{\text{pre}}(t, p) - H_{\text{post}}(t, p)}{H_{\text{pre}}(t, p) + \epsilon}, \quad (9)$$

where $\epsilon > 0$ is a small constant ensuring numerical stability. For each trajectory τ , candidates are ranked by $g(t, p)$, and only the top- K scoring pivots are selected for training. We fix $K = 10$ a priori without tuning; Appendix F shows performance is flat over $K = 5$ to 20 and degrades only when K grows large enough that selection ceases to be selective. The metric $g(t, p)$ is used exclusively as a filtering mechanism and does not scale the loss magnitude during optimization.

Figure 1 visualizes this selection process via the entropy dynamics of a single denoising trajectory. At critical structural bottlenecks, committing a single token drastically reduces the uncertainty of the remaining sequence. Our Information Gain metric effectively captures these moments of sharp entropy collapse (marked by red dashed lines and stars). This confirms that Pivot-SD isolates the decisive commitments that dictate the logical outcome, rather than uniformly weighting the entire sequence.

3.2.3 Training objective

Each selected pivot is mapped to a trajectory outcome score $s(\tau) \in \{+1, -1\}$, which evaluates the correctness of the final response sequence y generated by trajectory τ :

$$s(\tau) = \begin{cases} +1, & \text{if } y \text{ is correct,} \\ -1, & \text{otherwise.} \end{cases} \quad (10)$$

Every chosen pivot yields a training tuple $z = (M_t, y, p, y_p, s(\tau))$. For a selected pivot token y_p , we define the positive cross-entropy loss as

$$\ell_p^+(\theta; M_t, y_p) = -\log P_\theta(y_p | M_t), \quad (11)$$

and the token-level unlikelihood penalty as

$$\ell_p^-(\theta; M_t, y_p) = -\log(1 - P_\theta(y_p | M_t)). \quad (12)$$

During training, the predicted probability $P_\theta(y_p | M_t)$ in Eq. (12) is clipped away from 1 to maintain numerical stability. Failed trajectories often contain structurally sound intermediate sequences,

Algorithm 1: Pivot-SD: Offline Outcome-Conditioned Pivot Self-Distillation

Input: Training prompts $\mathcal{X}$, frozen base model θ_0 , trajectories per prompt G , pivots per trajectory K

Output: Fine-tuned model θ

```

1 Initialize pivot dataset  $\mathcal{D}_{\text{pivot}} \leftarrow \emptyset$ 
2 foreach prompt  $x \in \mathcal{X}$  do
3   for  $j \leftarrow 1$  to  $G$  do
4     Sample a denoising trajectory
        $\tau = \{(M_t, \mathcal{B}_t, \mathcal{U}_t)\}_{t=1}^T$  and response  $y$ 
       from  $\theta_0$ 
5     Assign trajectory outcome  $s(\tau)$  using
       Eq. (10)
6     Construct candidate pivots
        $\mathcal{C}(\tau) = \{(t, p, y_p) : p \in \mathcal{U}_t\}$ 
7     foreach candidate  $(t, p, y_p) \in \mathcal{C}(\tau)$  do
8       Compute information-gain score  $g(t, p)$ 
       using Eq. (9)
9     Select the top- $K$  candidates by  $g(t, p)$ 
10    foreach selected pivot  $(t, p, y_p)$  do
11      Add  $(M_t, y, p, y_p, s(\tau))$  to  $\mathcal{D}_{\text{pivot}}$ 
12 Fine-tune  $\theta$  on  $\mathcal{D}_{\text{pivot}}$  using Eq. (14)
13 return  $\theta$ 

```

such as valid syntactic phrasing or correct localized algebraic steps. To avoid penalizing these helpful components, Pivot-SD applies negative supervision exclusively to the highest-impact pivots extracted from failed generations. The complete outcome-conditioned pivot loss for a single tuple z is formulated as

$$L_{\text{pivot}}(z) = \mathbf{1}[s(\tau) = +1] \ell_p^+(\theta; M_t, y_p) + \lambda_{\text{neg}} \mathbf{1}[s(\tau) = -1] \ell_p^-(\theta; M_t, y_p), \quad (13)$$

where λ_{neg} is a hyperparameter scaling the unlikelihood penalty. Let $\mathcal{D}_{\text{pivot}}$ be the offline dataset compiled by executing this selection process across all training prompts $\mathcal{X}$ and generated trajectories. The final fine-tuning objective minimizes the expected pivot loss over this curated dataset:

$$\min_{\theta} \frac{1}{|\mathcal{D}_{\text{pivot}}|} \sum_{z \in \mathcal{D}_{\text{pivot}}} L_{\text{pivot}}(z). \quad (14)$$

We initialize θ from the base model θ_0 and optimize Eq. (14) using standard gradient-based optimization.

3.3 Algorithmic Implementation

We synthesize the full training pipeline in Algorithm 1, detailing the offline pipeline from data collection to fine-tuning.

Table 1: **Performance comparison of LLaDA-8b-instruct on math and coding tasks.** We report the accuracy evaluated at generation lengths of 256 and 512. The baselines include both SFT and RL methods. All non-base methods are trained under the same controlled setup whenever applicable. Methods marked with ‡ are evaluated over three fully re-randomized runs at the 256-token setting; we report standard deviations in Appendix D.

Method	MATH		GSM8K		HumanEval+		MBPP+	
	256	512	256	512	256	512	256	512
Base	31.40	38.8	75.13	83.40	32.32	42.07	43.12	44.44
SFT-GT‡	35.47	38.6	76.94	83.17	33.94	36.59	46.32	42.86
SFT-SD‡	31.73	38.6	73.90	81.96	35.78	40.85	45.92	44.97
diffu-GRPO (budget-matched)‡	33.47	38.2	76.04	82.11	32.52	42.07	43.39	45.24
diffu-GRPO	35.20	40.6	77.63	83.62	34.15	40.85	42.59	43.65
wdl++ (budget-matched)‡	33.27	39.2	76.93	82.71	33.54	40.24	43.12	43.92
wdl++	35.40	39.4	76.65	82.71	34.15	38.41	44.97	44.44
Pivot-SD (ours)‡	37.47	42.6	79.51	84.38	40.43	41.46	47.12	48.15

Table 2: **Ablation study of pivot selection and credit assignment.** “IG” denotes denoising states selected by our information-gain criterion. “All” applies the loss to all supervised masked tokens in the selected state, while “Pivot” applies the loss only to the selected pivot token. CE denotes cross-entropy and UL denotes unlikelihood. Methods marked with ‡ are evaluated over three fully re-randomized runs; standard deviations are in Appendix D.

Method	State	Target	Pos.	Neg.	MATH	GSM8K	HumanEval+	MBPP+
Positive All-token	IG	All	CE	-	31.60	74.37	36.59	39.68
Positive Pivot	IG	Pivot	CE	-	32.20	72.48	35.37	43.65
Entropy pivot‡	Entropy	Pivot	CE	-	32.27	74.75	35.57	44.80
Pos+Neg All-token‡	IG	All	CE	State-UL	35.00	74.44	36.80	44.54
Random-Step Pivot‡	Random	Pivot	CE	Pivot-UL	29.93	76.51	39.00	44.80
Random-Token Pivot‡	IG	Random	CE	Pivot-UL	34.87	76.47	34.78	45.24
Pivot-SD (ours)‡	IG	Pivot	CE	Pivot-UL	37.47	79.51	40.43	47.12

3.4 Relation to SFT and Online RL

Pivot-SD occupies a distinct methodological space between standard SFT and online RL. Standard masked SFT applies uniform token-level supervision, ignoring the variance in causal impact across different denoising positions, while online RL utilizes trajectory-level rewards that lack localized precision, and rejection-based self-distillation discards failed trajectories entirely.

In contrast, Pivot-SD executes an outcome-to-commitment projection, translating a single terminal outcome into K state-conditioned constraints. The pipeline deliberately separates identification from optimization: information gain isolates the high-leverage pivots, the verifier dictates the update direction, and localized losses apply it without broad credit or blame. Consequently, Pivot-SD uniquely extracts explicit localized negative targets from failed trajectories, suppressing decisive errors while preserving surrounding valid reasoning.

The simplicity of the cross-entropy and unlikelihood formulation is a deliberate design choice: our core contribution is constructing these valid local-

ized targets from trajectory-level outcomes, rather than devising a novel scalar loss. Finally, because all trajectories are sampled statically from the initial frozen base model θ_0 , the compiled dataset $\mathcal{D}_{\text{pivot}}$ remains invariant, allowing for reproducible offline hyperparameter tuning without the instabilities typical of online policy-gradient methods (Appendix H).

4 Experiments

We evaluate whether outcome-conditioned pivot self-distillation can improve masked diffusion language models under a small post-training budget. Our main experiments use LLaDA-8B-Instruct as the backbone.

Training data. For each math reasoning dataset, we sample 200 training questions from the corresponding training split: GSM8K for GSM8K evaluation (Cobbe et al., 2021) and MATH for MATH500 (Hendrycks et al., 2021) evaluation. For code tasks, we sample 200 training questions from AceCode (Zeng et al., 2025) and evaluate on HumanEval+ and MBPP+ (Chen et al., 2021; Liu et al.,

2023; Austin et al., 2021b). For each training question, we generate four denoising trajectories from the frozen base model, yielding only 800 sampled trajectories per dataset. Each trajectory is labeled as successful or failed using the task verifier: exact-answer matching for math and unit-test execution for code.

Baselines. We compare against offline SFT baselines and online RL-style baselines. SFT-GT trains on ground-truth solutions with standard random masking. SFT-SD uses the same generated rollout pool as Pivot-SD, but trains on successful generated solutions with ordinary masked SFT. For RL-style baselines, we include diffu-GRPO (Zhao et al., 2025) and wd1++ (Tang et al., 2026). Here wd1++ denotes the step-wise variant proposed by Tang et al. (2026), which extends their base wd1 objective to reuse the intermediate clean completions produced at each denoising step; we adopt it as the stronger of the two variants. Their budget-matched variants use the same 200-question, four-rollout, 1,000-step budget as Pivot-SD, while the larger variants use 2,000 questions and 5,000 optimizer steps.

Evaluation. All rollout-based methods collect training trajectories with a maximum generation length of 256 tokens. SFT-GT does not use model rollouts and instead trains on budget-feasible ground-truth solutions with the same response budget. We evaluate all trained models on MATH500, GSM8K, HumanEval+, and MBPP+. Unless otherwise stated, all methods use the same decoding configuration and answer-extraction or unit-test evaluation pipeline. The 256-token setting is our primary controlled setting, while 512-token evaluation is used to measure length transfer. Methods trained under this controlled budget are evaluated over three fully re-randomized runs at 256 tokens; we report means in the main tables, with per-run standard deviations and the full protocol in Appendix D. Detailed experimental configurations and hyperparameters are provided in Appendix C.

4.1 Main Results

Table 1 reports the main comparison. We use the 256-token response budget as the primary controlled setting, since it matches the budget used to collect pivot supervision. Under this small-budget setting, Pivot-SD achieves the best mean on all four benchmarks.

Table 3: **Compute accounting.** Wall-clock hours measured on a 2-GPU setup. Unlike RL baselines that interleave costly online rollouts, Pivot-SD cleanly decouples the pipeline into offline generation and lightweight offline training. RL baseline times are averaged across evaluated tasks.

Method	Steps	Gen. (h)	Train (h)	Total (h)
diffu-GRPO (matched)	1000	0	5.4	5.4
diffu-GRPO	5000	0	22.9	22.9
wd1++ (matched)	1000	0	3.6	3.6
wd1++	5000	0	17.3	17.3
Pivot-SD (ours)	1000	2.5	0.3	2.8

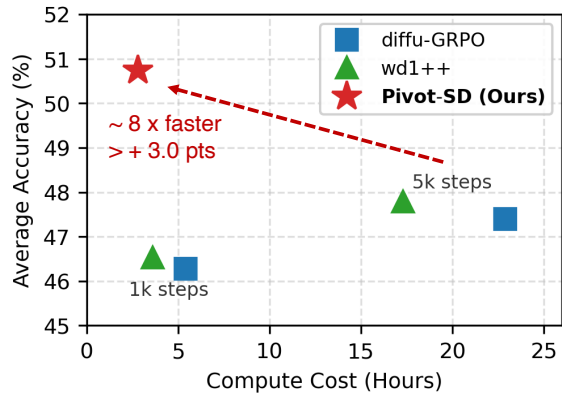


Figure 2: **Accuracy vs. Compute Cost.** Average reasoning accuracy versus total 2-GPU wall-clock hours. Pivot-SD achieves a > 3.0 absolute point improvement while requiring only a fraction of the compute time compared to fully converged RL baselines.

The key comparison is with methods trained under the same budget. SFT-SD uses the same self-generated rollout pool as Pivot-SD but applies ordinary masked SFT to successful generated solutions. Budget-matched diffu-GRPO and wd1++ use the same 200 training questions and 1,000 optimizer steps. Pivot-SD outperforms these budget-matched baselines overall, indicating that localizing credit and blame to high-information pivots is a more efficient use of the small rollout pool. The comparison is not specific to the 200-question budget: Pivot-SD remains strong from 50 to 2,000 training questions (Appendix E).

The 512-token columns evaluate length transfer. Pivot-SD continues to perform strongly on math reasoning and obtains the best overall average, but the code results are mixed, with HumanEval+ slightly favoring other 512-token baselines. We therefore use the 512-token results as supporting length-transfer evidence, while our main claim is based on the matched 256-token setting.

Table 4: **Dream generalization results.** We keep the same baseline categories as in the LLaDA experiments and report the same evaluation metrics.

Method	MATH	GSM	HE	MBPP
Base (Dream-Instruct-7B)	37.97	79.55	54.27	60.58
SFT-GT	35.40	61.49	54.27	59.26
SFT-SD	40.80	70.74	48.78	61.38
Pivot-SD (ours)	42.20	81.88	58.54	61.64

Table 5: **Average out-of-domain (OOD) performance.** Each column reports the mean accuracy over the benchmarks not used for training: MATH- and GSM8K-trained models are evaluated on the other three benchmarks, AceCode-trained models on MATH and GSM8K. Column composition therefore differs, and “Overall” is the micro-average over all source-target pairs rather than the mean of the three columns.

Method	Training domain			Overall
	MATH	GSM8K	AceCode	
Base	50.19	35.61	53.27	45.49
SFT-GT	53.57	39.46	51.80	47.83
SFT-SD	51.13	35.44	51.21	45.27
diffu-GRPO (matched)	50.89	36.45	55.49	46.62
diffu-GRPO	53.02	37.99	54.66	47.79
wd1++ (matched)	50.71	38.06	53.76	46.73
wd1++	52.55	37.60	53.81	47.25
Pivot-SD (ours)	52.87	39.82	55.42	48.61

4.2 What Makes Pivot-SD Work?

Ablation setup. Table 2 isolates two design choices: the denoising state used for training and the token locations that receive credit (via cross-entropy, CE) or blame (via unlikelihood, UL). **Positive All-token** applies CE to all supervised tokens in IG-selected states from successful trajectories. **Positive Pivot** applies CE only to the IG pivot from successful trajectories. **Entropy pivot** replaces the IG criterion with local uncertainty, selecting the top- K highest-entropy committed tokens within successful trajectories and applying CE to them. **Pos+Neg All-token** applies full-state CE/UL on IG-selected states. **Random-Step Pivot** applies the same pivot-local CE/UL objective to a token decoded at a random denoising step. **Random-Token Pivot** keeps the IG-selected state but replaces the IG pivot with a randomly selected decoded token. **Pivot-SD (ours)** applies pivot-local CE/UL to the selected high-information pivot.

Pivot-local credit assignment matters. Table 2 ablates where the supervision is applied. Positive All-token and Pos+Neg All-token spread the loss

over all supervised masked tokens in the selected denoising state. In contrast, Pivot-SD applies credit and blame only to the selected pivot token. The gains over all-token variants show that trajectory feedback should not be distributed uniformly over the whole state; concentrating the update on high-information pivot decisions is more effective.

Information-gain pivot selection matters. The random pivot controls test whether any decoded token can serve as a useful local update. Random-Step Pivot chooses pivots from random denoising steps, while Random-Token Pivot keeps the IG-selected state but replaces the selected pivot with a random decoded token. Pivot-SD outperforms these controls on the overall trend, indicating that both the denoising state and the selected pivot token matter. Entropy pivot tests the selection criterion itself. It underperforms Pivot-SD by a wide margin and lands close to Positive Pivot, showing that local uncertainty is not the signal that matters.

4.3 Compute Efficiency

Offline pivot distillation avoids online rollout overhead. Table 3 compares the wall-clock training costs measured on a 2-GPU setup. RL-style baselines inherently interleave generation and optimization, making their online training loop substantially expensive (up to 22.9 hours for diffu-GRPO at 5,000 steps). In contrast, Pivot-SD cleanly decouples the pipeline into a small offline rollout collection stage (2.5 hours) followed by an extremely lightweight offline fine-tuning phase (0.3 hours). This brings the total time to 2.8 hours, making the method substantially cheaper than online RL-style training under the same hardware setup.

Superior compute-performance trade-off. As illustrated in Figure 2, this structural efficiency does not come at the cost of performance. Pivot-SD significantly shifts the Pareto frontier of accuracy versus compute cost. It improves the average score while requiring roughly $8\times$ less wall-clock time than the larger diffu-GRPO run. This shows that a small offline pool of 800 verifier-labeled trajectories can provide a favorable compute-performance trade-off compared with repeated online rollout collection.

4.4 Backbone Generalization

Pivot-SD transfers to another diffusion backbone. Table 4 evaluates the same training recipe on the Dream-v0-Instruct-7B base model (Ye et al.,

2025). Pivot-SD improves over the Dream base model and the SFT baselines on all four benchmarks. This suggests that the benefit of pivot-local credit assignment is not specific to LLaDA, but transfers to another masked diffusion language model.

4.5 Out-of-Domain Transfer

Table 5 reports transfer to benchmarks unseen during training, grouped by training domain. Models trained on MATH or GSM8K are evaluated on the remaining three benchmarks, while AceCode-trained models are evaluated on MATH and GSM8K. The Overall score is the micro-average across all source-target pairs.

Pivot-SD transfers most consistently. Using the same small 200-question budget, Pivot-SD achieves the best overall average. Although not the top method for every individual pair, it has the best average rank across the three training domains and never falls below third. This indicates that pivot-local credit assignment learns transferable denoising preferences rather than merely overfitting to the source domain. The full source-target breakdown is in Appendix G.

5 Conclusion

We introduced Pivot-SD, a data-efficient offline self-distillation framework for masked diffusion LMs that resolves the credit-assignment bottleneck via step-conditioned denoising commitments. By localizing structural bottlenecks through counterfactual information-gain, Pivot-SD applies targeted unlikelihood to erroneous pivots, converting failed trajectories into precise training signals while preserving valid reasoning sub-steps. Empirical results demonstrate that Pivot-SD outperforms sample-intensive online RL and uniform SFT baselines using only 200 training prompts, establishing a robust, compute-efficient paradigm for structured reasoning post-training.

6 Limitations

While Pivot-SD demonstrates strong data efficiency, several aspects of the design remain open. Our objective includes a few hyperparameters that are set per configuration rather than learned, and some adjustment across domains and backbones is expected. An adaptive rule based on step-wise entropy collapse would let supervision follow task complexity without this step. Our formulation also

operates within a denoising block, so extending step-conditioned credit assignment across block boundaries would be needed for long-context reasoning. Finally, our experiments span two masked diffusion backbones at a comparable scale, and applying pivot-local supervision as dLMs grow is a natural next step.

Acknowledgements

RGK gratefully acknowledges support from the Canada Research Chairs Program (CRC-2022-00049) and the Canada CIFAR AI Chairs Program. This research was funded in part by an NFRF Special Call Award (NFRFR-2022-00526) and NSERC Discovery Grant (RGPIN-2022-04546). This work was also partly supported by the “Advanced GPU Utilization Support Program” funded by the Government of the Republic of Korea (Ministry of Science and ICT) (05-26-04-0100), and by Institute for Information & communications Technology Planning & Evaluation (IITP) grants funded by the Korea government (MSIT) (RS-2022-00143911, AI Excellence Global Innovative Leader Education Program; RS-2026-25522672, Development of Unified Reasoning Technology Mimicking Human Cognition for Hierarchical Understanding and Unbounded Problem Solving; RS-2019-II190075, Artificial Intelligence Graduate School Support Program (KAIST)).

References

- Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. 2021a. [Structured denoising diffusion models in discrete state-spaces](#). In *Advances in Neural Information Processing Systems*.
- Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and Charles Sutton. 2021b. [Program synthesis with large language models](#). *Preprint*, arXiv:2108.07732.
- Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, and 39 others. 2021. [Evaluating large language models trained on code](#). *Preprint*, arXiv:2107.03374.
- Ranfei Chen, Ming Chen, and Kaifei Wang. 2025. [Reasoning in diffusion large language models is concentrated in dynamic confusion zones](#). *Preprint*, arXiv:2511.15208.

- Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. [Training verifiers to solve math word problems](#). *Preprint*, arXiv:2110.14168.
- Zi-Yi Dou, Cheng-Fu Yang, Xueqing Wu, Kai-Wei Chang, and Nanyun Peng. 2024. [Re-ReST: Reflection-reinforced self-training for language agents](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 15394–15411, Miami, Florida, USA. Association for Computational Linguistics.
- Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, and 175 others. 2025. [Deepseek-r1 incentivizes reasoning in llms through reinforcement learning](#). *Nature*, 645(8081):633–638.
- Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. [Measuring mathematical problem solving with the MATH dataset](#). In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.
- Zachary Horvitz, Raghav Singhal, Hao Zou, Carles Domingo-Enrich, Zhou Yu, Rajesh Ranganath, and Kathleen McKeown. 2025. [Rethinking reasoning with mdlms: Early exits, post-hoc reasoning, and beyond](#). *Preprint*, arXiv:2510.19990.
- Zemin Huang, Zhiyang Chen, Zijun Wang, Tiancheng Li, and Guo-Jun Qi. 2025. [Reinforcing the diffusion chain of lateral thought with diffusion language models](#). In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.
- Seo Hyun Kim, Sunwoo Hong, Hojung Jung, Youngrok Park, and Se-Young Yun. 2025. [Klass: Kl-guided fast inference in masked diffusion models](#). In *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pages 92267–92301. Curran Associates, Inc.
- Xiang Lisa Li, John Thickstun, Ishaan Gulrajani, Percy Liang, and Tatsunori Hashimoto. 2022. [Diffusion-LM improves controllable text generation](#). In *Advances in Neural Information Processing Systems*.
- Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2024. [Let’s verify step by step](#). In *The Twelfth International Conference on Learning Representations*.
- Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. 2023. [Is your code generated by chat-GPT really correct? rigorous evaluation of large language models for code generation](#). In *Thirty-seventh Conference on Neural Information Processing Systems*.
- Aaron Lou, Chenlin Meng, and Stefano Ermon. 2024. Discrete diffusion modeling by estimating the ratios of the data distribution. In *Proceedings of the 41st International Conference on Machine Learning, ICML’24*. JMLR.org.
- Tianwei Ni, Allen Nie, Sapana Chaudhary, Yao Liu, Huzefa Rangwala, and Rasool Fakoor. 2025. [Of-line learning and forgetting for reasoning with large language models](#). *Preprint*, arXiv:2504.11364.
- 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](#). In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.
- Daisuke Oba, Hiroki Furuta, and Naoaki Okazaki. 2026. [Diffusion-state policy optimization for masked diffusion language models](#). *Preprint*, arXiv:2602.06462.
- Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Gray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. [Training language models to follow instructions with human feedback](#). In *Advances in Neural Information Processing Systems*.
- Kevin Rojas, Jiahe Lin, Kashif Rasul, Anderson Schneider, Yuriy Nevmyvaka, Molei Tao, and Wei Deng. 2026. [Improving reasoning for diffusion language models via group diffusion policy optimization](#). In *The Fourteenth International Conference on Learning Representations*.
- Subham Sekhar Sahoo, Marianne Arriola, Aaron Gokaslan, Edgar Mariano Marroquin, Alexander M Rush, Yair Schiff, Justin T Chiu, and Volodymyr Kuleshov. 2024. [Simple and effective masked diffusion language models](#). In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.
- Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. [Deepseekmath: Pushing the limits of mathematical reasoning in open language models](#). *Preprint*, arXiv:2402.03300.
- Avi Singh, John D Co-Reyes, Rishabh Agarwal, Ankesh Anand, Piyush Patil, Xavier Garcia, Peter J Liu, James Harrison, Jaehoon Lee, Kelvin Xu, Aaron T Parisi, Abhishek Kumar, Alexander A Alemi, Alex Rizkowsky, Azade Nova, Ben Adlam, Bernd Bohnet, Gamaleldin Fathy Elsayed, Hanie Sedghi, and 21 others. 2024. [Beyond human data: Scaling self-training for problem-solving with language models](#). *Transactions on Machine Learning Research*. Expert Certification.

- Xiaohang Tang, Rares Dolga, Sangwoong Yoon, and Ilija Bogunovic. 2026. [wd1: Weighted policy optimization for reasoning in diffusion language models](#). In *The Fourteenth International Conference on Learning Representations*.
- Guanghan Wang, Gilad Turok, Yair Schiff, Marianne Arriola, and Volodymyr Kuleshov. 2025. [d2: Improved techniques for training reasoning diffusion language models](#). *Preprint*, arXiv:2509.21474.
- Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. 2024. [Math-shepherd: Verify and reinforce LLMs step-by-step without human annotations](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9426–9439, Bangkok, Thailand. Association for Computational Linguistics.
- Sean Welleck, Ilia Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston. 2019. [Neural text generation with unlikelihood training](#). *Preprint*, arXiv:1908.04319.
- Shaoan Xie, Lingjing Kong, Xiangchen Song, Xinshuai Dong, Guangyi Chen, Eric P. Xing, and Kun Zhang. 2026. [Advancing reasoning in diffusion language models with denoising process rewards](#). In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 42703–42720, San Diego, California, United States. Association for Computational Linguistics.
- Guowei Xu, Wenxin Xu, Jiawang Zhao, and Kaisheng Ma. 2025. [Gift: Guided importance-aware fine-tuning for diffusion language models](#). *Preprint*, arXiv:2509.20863.
- Kaisen Yang, Jayden Teoh, Kaicheng Yang, Yitong Zhang, and Alex Lamb. 2026. [Improving sampling for masked diffusion models via information gain](#). *Preprint*, arXiv:2602.18176.
- Jiacheng Ye, Zhihui Xie, Lin Zheng, Jiahui Gao, Zirui Wu, Xin Jiang, Zhenguo Li, and Lingpeng Kong. 2025. [Dream 7b: Diffusion large language models](#). *Preprint*, arXiv:2508.15487.
- Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. 2022. [STaR: Bootstrapping reasoning with reasoning](#). In *Advances in Neural Information Processing Systems*, volume 35, pages 15476–15488. Curran Associates, Inc.
- Huaye Zeng, Dongfu Jiang, Haozhe Wang, Ping Nie, Xiaotong Chen, and Wenhui Chen. 2025. [Acecoder: Acing coder rl via automated test-case synthesis](#). *Preprint*, arXiv:2502.01718.
- Siyan Zhao, Devaansh Gupta, Qinqing Zheng, and Aditya Grover. 2025. [d1: Scaling reasoning in diffusion large language models via reinforcement learning](#). In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

A Credit-Assignment Landscape

Table 6 details the credit-assignment landscape across a broad set of related methods, including process reward models, the training-free Info-Gain Sampler, and answer-aware selection (Horvitz et al., 2025; Yang et al., 2026). This expanded landscape highlights the joint uniqueness of Pivot-SD. First, its unit of supervision is the realized commitment (t, p, y_p) evaluated under the partially denoised state M_t . Second, it actively constructs localized negative targets from failed trajectories. As shown across the entries, no prior method combines commitment-level supervision with an explicit, token-level unlikelihood target derived directly from trajectory outcomes.

B Illustration of Information-Gain Pivot Selection

Figure 3 illustrates how Pivot-SD selects a pivot during masked denoising. Among the positions committed at a given step, the one whose commitment most reduces the predictive entropy over the remaining masked block is selected as the pivot (t, p, y_p) , and the localized CE or UL update is applied only there.

C Experimental Details and Configurations

In this section, we provide detailed configurations for our training and evaluation pipelines to ensure reproducibility.

Hardware Infrastructure. All training, trajectory generation, and evaluation pipelines across all methods were executed on a hardware setup consisting of two NVIDIA L40 GPUs.

Generation and Rollout Limits. For all training methodologies—including our offline trajectory sampling for Pivot-SD, the success-only sampling for SFT-SD, and the online rollouts for the RL baselines (diffu-GRPO and wd1++)—we strictly enforce a maximum generation length of **256 tokens**. This ensures a compute-matched and controlled environment during the training phase.

Optimization and Hyperparameters. All offline methods (SFT-GT, SFT-SD, and Pivot-SD) are trained for 1,000 optimizer steps with a batch size of 8. The baseline RL methods are trained under both a budget-matched setting (1,000 steps) and

an extended setting (5,000 steps) to observe convergence scaling. For Pivot-SD, rather than tuning the negative unlikelihood weight λ_{neg} , we set it from the positive-to-negative trajectory ratio of the offline pool so that the two branches contribute comparably. We set $\lambda_{\text{neg}} = 1$ for MATH, and $\lambda_{\text{neg}} = 5$ for GSM8K to account for an approximate 5:1 ratio of positive to negative trajectories. Coding is the one exception: since models are trained on Ace-Code but evaluated on HumanEval+ and MBPP+, the ratio-matched value risks over-suppression under distribution shift, so we lower the penalty to $\lambda_{\text{neg}} = 0.02$ along a log scale on a held-out split of roughly 100 examples.

Evaluation Protocol and Decoding Strategy.

During the evaluation phase on MATH500, GSM8K, HumanEval+, and MBPP+, we evaluate the models under two different maximum generation length constraints: **256 tokens** and **512 tokens**. The 256-token evaluation tests the models’ performance under the exact same constraints seen during training, while the 512-token evaluation assesses length generalization and the models’ ability to sustain reasoning over longer contexts without suffering from degradation. We evaluate and report the performance at intermediate checkpoints of 500 and 1,000 optimization steps. To guarantee fully deterministic and reproducible evaluation results, all evaluations using the LLaDA-8B-Instruct backbone employ top-1 confidence decoding with a temperature of 0.

D Multi-Seed Evaluation Protocol

Scope. We repeat every method trained under our controlled budget (200 questions, 1,000 optimizer steps) three times and report means in the main tables. The base model is not trained and is evaluated with deterministic decoding (temperature 0, top-1), so seed variance is undefined for it. The extended-budget RL baselines are too slow to repeat three times and are reported as single runs. The 512-token columns are likewise single runs, since they measure length transfer rather than the training-matched setting.

What is re-randomized. Each run independently re-samples (i) the 200-question training set, (ii) the trajectory pool, re-generated from the frozen base model at temperature 0.9, and (iii) the training seed. The runs are therefore not seed replicates of a fixed dataset: they resample the entire offline

Method	Local unit	Localization signal	Training regime	Localized Failure Target
STaR / rejection SD (Zelikman et al., 2022)	Full successful rationale	Final correctness	Offline SFT	No; failed trace filtered or retried
Process supervision (PRMs) (Lightman et al., 2024; Wang et al., 2024)	Textual reasoning step	Human or separately constructed process label	PRM / SFT / RL	Possible, but requires process-label construction
GIFT (Xu et al., 2025)	Final-response token position	Local predictive entropy	Offline weighted SFT	No
Denosing Process Reward / ATPO (Xie et al., 2026; Chen et al., 2025)	Denosing interval or step	Outcome-progress reward or trajectory uncertainty	Online RL	No explicit token-level UL target
DiSPO (Oba et al., 2026)	State-conditioned mask filling / newly filled tokens	Branched terminal rewards	Policy optimization	No cached realized-pivot UL target
Info-Gain Sampler / KLASS (Yang et al., 2026; Kim et al., 2025)	Candidate inference action	Effect on future-mask uncertainty	Inference only	N/A
Horvitz et al. (Horvitz et al., 2025)	Answer block / reasoning trace	Answer entropy or gold-answer probability	Inference / trace self-distillation	No localized failed-pivot UL
Pivot-SD (ours)	Realized commitment (t, p, y_p) under M_t	Counterfactual entropy reduction over other masks	Cached offline CE / UL	Yes

Table 6: **Credit-assignment landscape.** Across the closest prior work, Pivot-SD is the only method whose supervised unit is the realized denoising commitment (t, p, y_p) and that constructs a localized negative target from failed trajectories.

pipeline that produces $\mathcal{D}_{\text{pivot}}$. Because evaluation is deterministic, all reported variance originates from training, none from decoding.

Reading the standard deviations. Evaluation-set size alone induces substantial spread on the code benchmarks: HumanEval+ contains only 164 problems, giving a binomial standard error of about 3.8 points. The larger standard deviations there are consistent with this floor and are not specific to any method.

Separation between methods. On MATH, GSM8K, and HumanEval+, the intervals spanned by Pivot-SD and by the strongest competing method do not overlap at one standard deviation. On MBPP+ they do: Pivot-SD attains the higher mean (47.12 ± 1.20 against 46.32 ± 0.45 for SFT-GT), but the two are not separated at this sample size, and we do not claim an improvement there. Table 8 also shows that the random-selection controls are unstable in a way the single-run numbers concealed: Random-Step Pivot has a standard deviation of 3.13 on MATH, where its mean falls below the untrained base model, and 5.51 on Hu-

manEval+. Random commitments occasionally land on useful tokens, but they do not constitute a dependable supervision signal.

E Training Budget Sweep

Our main experiments fix the training budget at $q = 200$ questions. To verify that the comparison is not specific to that choice, we vary $q \in \{50, 200, 400, 2000\}$ and retrain every method under the same protocol in Table 9.

The advantage holds at the largest budget. diffu-GRPO and wd1++ regenerate rollouts online at every optimizer step, so the largest budget is where online exploration should pay off most. At $q2000$, Pivot-SD is nonetheless best on all four benchmarks while training offline, and the gap widens further at the largest budget since the RL baselines there use five times the optimizer steps and ten times the data.

Small budgets already beat large ones. With only 200 questions, Pivot-SD matches or exceeds every baseline at any budget on GSM8K and HumanEval+, including baselines trained on ten times

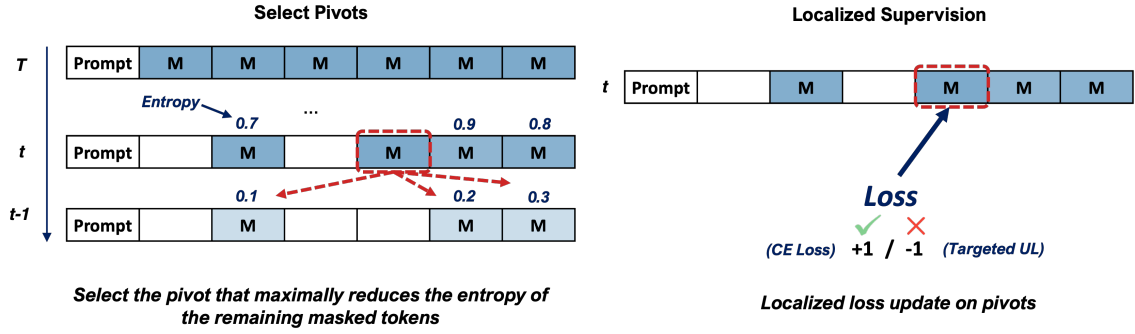


Figure 3: **Information-gain pivot selection.** Schematic of how a candidate commitment is scored. The red dashed box marks the selected pivot, whose commitment maximally reduces the predictive entropy over the remaining masked block; the localized loss is applied only at that position.

Table 7: **Main comparison with per-run standard deviations.** Mean $\pm$ standard deviation over three fully re-randomized runs, 256-token evaluation. Rows correspond to the $\ddagger$ -marked entries of Table 1; RL baselines are the budget-matched variants.

Method	MATH	GSM8K	HumanEval+	MBPP+
SFT-GT	35.47 $\pm$ 0.23	76.94 $\pm$ 1.05	33.94 $\pm$ 2.32	46.32 $\pm$ 0.45
SFT-SD	31.73 $\pm$ 0.58	73.90 $\pm$ 0.89	35.78 $\pm$ 1.54	45.92 $\pm$ 1.46
diffu-GRPO	33.47 $\pm$ 1.03	76.04 $\pm$ 0.80	32.52 $\pm$ 1.27	43.39 $\pm$ 0.27
wd1++	33.27 $\pm$ 0.50	76.93 $\pm$ 0.59	33.54 $\pm$ 2.80	43.12 $\pm$ 0.27
Pivot-SD (ours)	37.47 $\pm$ 1.21	79.51 $\pm$ 1.30	40.43 $\pm$ 2.75	47.12 $\pm$ 1.20

the data. The gain is therefore not an artifact of the budget we selected for the main experiments.

Exceptions. Baselines lead by at most 1.7 points, and only at the smallest budgets ($q50$ on GSM8K; $q50$ and $q200$ on MBPP+). The one larger gap is a non-monotone wd1++ spike at MATH $q400$ (39.2, against its own 35.4 at $q2000$ and 33.27 in the multi-seed comparison of Table 7), which is consistent with the instability of online RL in this regime.

F Pivot Budget K

The pivot budget K is the only quantity in Pivot-SD that controls how much of a trajectory receives supervision. We fixed $K = 10$ a priori rather than tuning it, and report here the sweep $K \in \{5, 10, 20, 40, 80\}$ for Pivot-SD and for the two random-selection controls, all trained on the same trajectory pool. The results are shown in Table 10.

$K = 10$ sits inside a flat plateau. Pivot-SD is best or tied at every $K \in \{5, 10, 20\}$ on all four benchmarks, so the value we fixed in advance falls well inside the stable range rather than at a tuned optimum.

Large K degrades by design. Performance drops from $K = 40$ on MATH and on both code benchmarks. A large K admits low-information commitments into the pivot set and applies unlikelyhood at positions that did not shape the trajectory, so the update is no longer selective. This is the same mechanism our main claim rests on, seen from the other side, and we make no claims outside the range $K = 5$ to 20.

Sparsity alone does not explain the gain. Random-Step Pivot supervises exactly as few positions as Pivot-SD but collapses on MATH across the operating range (17.6 to 26.6, against 37.8 to 38.6 for Pivot-SD). Which sparse subset is selected is what matters, not that the subset is sparse.

G Out-of-Domain Generalization

To understand the robustness of our learned representations, we evaluate the Out-of-Domain (OOD) generalization of models trained on a single domain and tested on unseen domains. The results are summarized in Table 11.

As shown in the table, Pivot-SD maintains highly competitive performance across unseen domains. By avoiding the destruction of locally valid representations during negative updates, our surgical

Table 8: **Ablation with per-run standard deviations.** Mean $\pm$ standard deviation over three fully re-randomized runs, 256-token evaluation.

Method	MATH	GSM8K	HumanEval+	MBPP+
Pos+Neg All-token	35.00 $\pm$ 1.25	74.44 $\pm$ 0.70	36.80 $\pm$ 0.35	44.54 $\pm$ 2.37
Random-Step Pivot	29.93 $\pm$ 3.13	76.51 $\pm$ 0.74	39.00 $\pm$ 5.51	44.80 $\pm$ 0.85
Random-Token Pivot	34.87 $\pm$ 1.94	76.47 $\pm$ 1.68	34.78 $\pm$ 1.07	45.24 $\pm$ 0.70
Entropy pivot	32.27 $\pm$ 1.10	74.75 $\pm$ 0.82	35.57 $\pm$ 1.76	44.80 $\pm$ 1.10
Pivot-SD (ours)	37.47 $\pm$1.21	79.51 $\pm$1.30	40.43 $\pm$2.75	47.12 $\pm$1.20

Table 9: **Accuracy across training budgets** (256-token evaluation). Column headers give the number of training questions q . Step budgets are fixed within each group ($q = 50/200$: 1,000 steps; $q = 400$: 2,000; $q = 2000$: 5,000), matching the extended RL baselines. All entries are single-run evaluations under the stated budget; the corresponding multi-seed means (for $q = 200$) are reported in Table 7.

Method	MATH				GSM8K				HumanEval+				MBPP+			
	50	200	400	2000	50	200	400	2000	50	200	400	2000	50	200	400	2000
SFT-GT	36.0	35.6	34.2	35.0	76.7	75.8	77.9	75.1	32.9	32.9	32.9	33.5	46.6	46.6	45.8	46.8
SFT-SD	31.6	31.4	33.0	31.6	72.4	74.9	72.7	73.2	37.8	35.4	37.2	37.8	45.5	45.8	44.7	45.5
diffu-GRPO	34.0	34.6	34.6	35.2	76.4	75.7	76.4	77.6	31.7	31.1	32.9	34.2	42.3	43.7	43.9	42.6
wd1++	32.4	33.8	39.2	35.4	75.9	77.1	79.0	76.7	32.3	32.9	31.1	34.2	44.7	43.1	45.8	45.0
Pivot-SD (ours)	36.2	38.6	36.0	37.2	75.0	79.5	80.1	78.9	42.1	40.9	40.2	38.4	46.0	46.0	46.6	47.1

credit assignment prevents catastrophic forgetting of domain-agnostic syntax and structural knowledge.

H Addressing Reward Sparsity in Low-Data RL: The wd1++ Bootstrap Analysis

A critical challenge when applying online reinforcement learning (RL) to discrete diffusion models in low-data regimes is reward sparsity. During our experiments with the wd1++ baseline on the 200-prompt AceCode subset ($s200$), we observed a pathological training collapse.

Because AceCode is a highly challenging domain, the base model frequently failed to pass any unit tests across all generated samples within a batch. Consequently, the group reward was entirely zero, rendering the advantage uniformly zero. Under the wd1++ dual-objective formulation (which combines positive weighted likelihood and negative weighted unlikelihood), a zero-advantage batch causes the softmax weights to become uniform, perfectly canceling out the positive and negative gradients. As a result, the model experiences a complete gradient collapse, evidenced by identical checkpoint hashes and completely stagnant evaluation metrics between step 500 and 1000 (see “w/o bootstrap” in Table 12).

To ensure a rigorous and fair baseline comparison—and to verify that our proposed Pivot-SD’s

sample efficiency is not merely an artifact of a broken RL baseline—we implemented a **Bootstrap Reward** shaping strategy for the $s200$ wd1++ runs. Specifically, if the unit test pass rate was zero, we injected a small dense structural reward (up to +0.05) if the generated code met basic syntactic requirements (e.g., proper def initialization, valid return statements, and general syntax compliance).

As shown in Table 12, injecting the bootstrap reward successfully revived the learning process. The model broke out of the flatline, achieving notable performance gains by step 1000, most prominently a +3.97 point improvement on MBPP+ (256 length) and consistent gains in the 512-length evaluations.

We confirmed that online RL baselines like wd1++ are inherently fragile in low-data, sparse-reward environments and require aggressive reward shaping to function. Even after we explicitly rescue the baseline with structural dense rewards, its peak performance does not surpass Pivot-SD. On MBPP+, the shaped baseline reaches 47.09, which is statistically indistinguishable from Pivot-SD’s 47.12 ± 1.20 ; consistent with Appendix D, we do not claim an improvement on this benchmark. On the remaining benchmarks the gap is clear: Pivot-SD leads on HumanEval+ (40.43 vs. 33.54), as well as on MATH and GSM8K, all without any such reward engineering. Taken together, these re-

Table 10: **Effect of the pivot budget K** ($q = 200$, 256-token evaluation). K is the number of pivots supervised per trajectory. Random-Step and Random-Token are the two random-selection controls of Table 2. We use $K = 10$ and scope our claims to $K \in \{5, 20\}$; $K = 40$ and 80 are reported to show the failure mode. Entries are single-run evaluations on a shared trajectory pool; the multi-seed mean for $K = 10$ is reported in Table 7.

K	MATH			GSM8K			HumanEval+			MBPP+		
	Random Step	Random Token	Ours	Random Step	Random Token	Ours	Random Step	Random Token	Ours	Random Step	Random Token	Ours
5	22.8	34.2	37.8	74.3	75.1	79.6	39.0	34.8	39.0	42.6	44.2	45.0
10	26.6	33.2	38.6	76.9	75.7	79.5	35.4	37.8	40.9	46.0	45.8	46.0
20	17.6	36.4	38.2	73.8	76.0	76.8	39.6	37.2	41.5	41.5	45.5	45.5
40	22.0	34.6	22.8	73.0	77.0	79.5	39.0	43.3	39.6	41.0	44.4	39.7
80	20.4	33.0	17.6	69.8	77.6	76.7	37.2	42.7	30.5	34.7	41.8	37.6

Table 11: **Out-of-Domain (OOD) generalization results.** Models are trained on one domain and evaluated on others. “budget-matched” denotes models trained with the 1,000-step budget configuration matching Pivot-SD.

Method	MATH Trained				GSM8K Trained				AceCode Trained			
	GSM8K	HumanEval+	MBPP+	Avg.	MATH	HumanEval+	MBPP+	Avg.	MATH	GSM8K	Avg.	
SFT-GT	76.35	37.80	46.56	53.57	35.40	37.20	45.77	39.46	30.20	73.39	51.80	
SFT-SD	75.74	32.93	44.71	51.13	30.60	30.49	45.24	35.44	29.40	73.01	51.21	
diffu-GRPO (budget-matched)	75.21	33.54	43.92	50.89	33.80	31.10	44.44	36.45	34.40	76.57	55.49	
diffu-GRPO	76.35	37.20	45.50	53.02	35.80	32.93	45.24	37.99	33.80	75.51	54.66	
wdl++ (budget-matched)	76.42	32.32	43.39	50.71	35.00	36.59	42.59	38.06	32.60	74.91	53.76	
wdl++	78.09	35.37	44.18	52.55	35.60	33.54	43.65	37.60	32.40	75.21	53.81	
Pivot-SD (ours)	78.09	36.59	43.92	52.87	36.60	38.41	44.44	39.82	34.80	76.04	55.42	
<i>Ablation Variants (Budget-matched)</i>												
Positive Pivot	73.46	33.54	41.27	49.42	31.40	36.59	41.01	36.33	30.80	73.92	52.36	
Positive All-token	75.36	34.76	42.86	50.99	29.60	36.59	43.92	36.70	32.60	73.77	53.19	
Pos+Neg All-token	75.44	38.41	43.39	52.41	33.00	34.76	44.44	37.40	33.60	75.44	54.52	
Random-Token Pivot	77.10	36.59	46.03	53.24	31.80	37.80	45.77	38.46	34.60	74.07	54.34	
Random-Step Pivot	75.06	32.93	37.57	48.52	33.40	29.27	41.27	34.65	35.00	76.42	55.71	

sults indicate that Pivot-SD matches or exceeds a reward-shaped RL baseline while relying on a simpler offline signal, supporting the view that high-impact pivot selection provides a robust learning signal without online reward engineering.

I Artifact Licenses and Intended Use

In our experiments, we utilize publicly available mathematical and code generation benchmarks, including GSM8K (MIT License), MATH (MIT License), HumanEval (MIT License), and MBPP (CC-BY-4.0). We also utilize the pre-trained masked diffusion language model, LLaDA, which is distributed under the Llama 3 Community License.

Our use of these artifacts strictly aligns with their intended use cases as standard structural reasoning and evaluation benchmarks for language modeling research. These datasets do not contain personally identifying information or offensive content. Furthermore, the trajectories and evaluation scripts generated during the development of Pivot-SD will be open-sourced under the MIT License to facilitate reproducibility and future research in masked

diffusion language models.

J Potential Risks

While Pivot-SD significantly improves the compute efficiency and reasoning capabilities of masked diffusion language models, it carries certain potential risks that merit consideration:

- **Bias and Hallucination Amplification:** As an offline self-distillation framework, Pivot-SD relies on trajectories generated by the frozen base model. Although our method filters for successful outcomes, it may still inherit and potentially amplify subtle biases or hallucinations deeply ingrained in the base model’s pre-trained representations.
- **Dependence on Automated Verifiers:** Our trajectory labeling currently relies on exact-match checking for mathematics and unit-test execution for code. Applying this framework to open-ended generation tasks without clear objective verifiers may lead to reward hacking or optimization toward superficial proxies of quality.

Variant	Math		Code (256)		Code (512)	
	MATH	GSM	MBPP+	HE+	MBPP+	HE+
<i>w/o bootstrap</i>						
Step 500	32.60	74.91	43.12	32.93	45.50	40.24
Step 1000	-	-	43.12	32.93	-	-
<i>w/ bootstrap</i>						
Step 500	33.20	74.53	44.71	31.10	46.56	39.63
Step 1000	31.20	74.45	47.09	33.54	46.03	40.85

Table 12: **Effect of Bootstrap Reward on wd1++ trained on 200 AceCode prompts.** GSM and HE+ denote GSM8K and HumanEval+, respectively. Without bootstrap, the model completely stagnates between step 500 and 1000 due to zero-advantage gradient cancellation. With the bootstrap reward, training resumes on code, yielding a +3.97 improvement on MBPP+ at step 1000, though MATH does not benefit. The wd1++ figures in the main tables (Tables 1 and 7) are the standard runs *without* bootstrap shaping; the 47.09 reported here is the single-run peak obtained *after* bootstrap shaping and is shown separately because it is not part of the controlled multi-seed protocol.

K Broader Impacts

The development of Pivot-SD offers practical benefits for the computational efficiency and accessibility of reasoning research in language models. Standard post-training methods, particularly online reinforcement learning (RL), are highly compute-intensive and require extensive hardware resources. By demonstrating that step-conditioned pivot supervision can extract dense learning signals from a minimal data budget (e.g., 200 prompts), Pivot-SD significantly reduces the computational overhead required to improve reasoning capabilities. In our empirical setting, this yielded an approximate 8x reduction in compute time compared to conventional RL baselines. This efficiency supports the principles of Green AI and lowers the barrier to entry, enabling researchers in resource-constrained academic institutions to conduct experiments on diffusion language model alignment and reasoning.

Additionally, the localized credit assignment mechanism presents useful properties for model alignment and safety. Standard rejection fine-tuning or full-sequence unlikelihood training risks over-correcting the model by suppressing locally valid syntactic fragments alongside undesirable behavior. Pivot-SD’s localized unlikelihood objective suggests a path for isolating and suppressing only the specific denoising commitments that lead to logical failures or policy violations, preserving the model’s general capabilities. While enhancing

reasoning and code synthesis naturally carries dual-use risks—such as the potential generation of malicious code or sophisticated misinformation—the offline, verifier-dependent nature of Pivot-SD is compatible with existing safety guardrails. Its sample efficiency also facilitates the rapid, low-cost deployment of defensive alignment updates to mitigate such downstream risks.