CATS: INFERENCE-ALIGNED SFT FOR DIFFUSION LLMS VIA CONTEXT-SENSITIVITY AWARE TRAJECTORY SAMPLING

Seunghyuk Oh[1] **Minjae Lee**[1] **Kevin Galim**[1] **Minseo Kim**[1] **Hyung Il Koo**[1]
Wonjun Kang[1] **Hanbaek Lyu**[2] **Kangwook Lee**[3,4]

[1] FuriosaAI [2] UW-Madison [3] KRAFTON [4] Ludo Robotics

ABSTRACT

Diffusion large language models (dLLMs) are trained to denoise randomly masked sequences, yet in practice, they are commonly decoded by progressively unmasking tokens in order of model confidence. Consequently, the masking patterns used in supervised fine-tuning (SFT) often diverge from those encountered during inference, resulting in suboptimal training signals. We propose Context-sensitivity Aware Trajectory Sampling (CATS), which constructs inference-aligned training trajectories directly from ground-truth targets. We use an initial model to iteratively categorize ground-truth tokens into groups based on how much context the model needs to confidently predict each one. By training on trajectories sampled in this order, the model learns masking patterns closer to what it would actually produce during inference. Across *Sudoku*, *Countdown*, and *Trip Planning*, our approach outperforms standard SFT, yielding accuracy gains across diverse settings. These findings demonstrate that aligning training trajectories with inference-time unmasking enables more reliable SFT of dLLMs.

1 INTRODUCTION

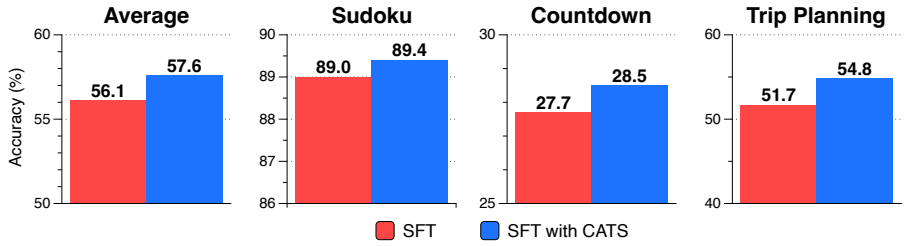


Figure 1: **Main results.** Accuracy of SFT and SFT with CATS (ours) on *Sudoku*, *Countdown*, and *Trip Planning*. SFT with CATS outperforms SFT on average across all tasks.

In contrast to autoregressive (AR) large language models (LLMs) that generate tokens sequentially from left to right, diffusion LLMs (dLLMs) (Nie et al., 2025; Ye et al., 2025b) begin with a fully masked sequence and incrementally unmask tokens in non-AR order. Because each revealed token serves as context for subsequent predictions, the unmasking order directly affects the difficulty of the remaining predictions. Most dLLMs employ confidence-guided decoding, prioritizing the unmasking of high-confidence tokens. Recent studies (Kang et al., 2025; Gong et al., 2025) demonstrate that this unmasking order substantially affects reasoning and planning performance.

However, supervised fine-tuning (SFT) of dLLMs is typically performed using a random forward process that uniformly masks token positions (Nie et al., 2025; Ye et al., 2025b). As a result, many masking patterns encountered during SFT may be misaligned with those arising at inference-time under confidence-guided decoding (Kim et al., 2025a), leading to suboptimal training signals. For instance, trivially predictable tokens can often be recovered with minimal context, whereas task-

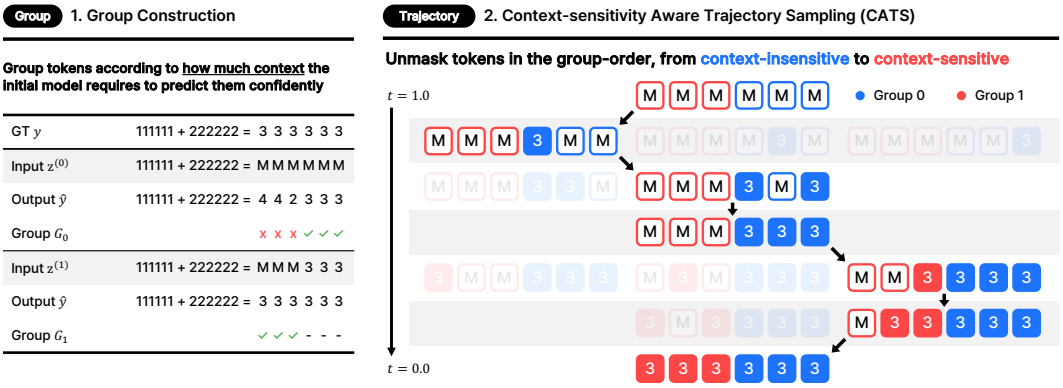


Figure 2: **Overview.** (Left) We categorize tokens into groups by *context-sensitivity*, which measures how much context each token requires for prediction. (Right) During training, we sample unmasking trajectories that reveal tokens in group order, ranging from context-insensitive to context-sensitive.

critical tokens in reasoning and planning may remain uncertain until the surrounding structure is restored. Thus, training should prioritize masked sequences that mimic inference-time behavior: resolving trivial tokens early and progressively revealing more difficult ones.

Previous research has looked at token-level noise scheduling and loss reweighting to address this gap (see Appendix A for more details). However, none of these approaches change the forward masking process itself to match inference-time trajectories. A common solution is to obtain inference-aligned trajectories via model rollouts (He et al., 2025; Kim et al., 2025b). Since these rollouts often diverge from ground-truth targets, they are incompatible with standard SFT and require reward-based methods like rejection fine-tuning (Yuan et al., 2023) or reinforcement learning (RL). Most prior work on training-inference mismatch in dLLMs has focused on RL-based solutions (He et al., 2025), while inference-aligned SFT with ground-truth supervision remains largely underexplored.

To address this, we introduce *Context-sensitivity Aware Trajectory Sampling* (CATS), a method that constructs inference-aligned training trajectories from ground-truth SFT targets without requiring expensive rollouts. The key idea is to iteratively categorize ground-truth tokens into ordered groups based on how much context the initial model needs to confidently predict each one. Training trajectories are then sampled in this order, so the model learns masking patterns closer to inference-time behavior while retaining ground-truth supervision (Figure 2). We mix these trajectories with standard random masking to maintain robustness.

In *Sudoku* (Zhao et al., 2025), *Countdown* (Ye et al., 2025a), and *Trip Planning* (Zheng et al., 2024), this mixed training approach outperforms standard SFT across diverse settings. Specifically, employing a 50:50 random-to-ordered mix increases average accuracy from 56.1% to 57.6% (+1.5%p) as summarized in Figure 1.

We make the following contributions:

- We identify a misalignment between training and inference in SFT for dLLMs.
- We propose *Context-sensitivity Aware Trajectory Sampling* (CATS), a lightweight method for inference-aligned SFT (Section 2).
- We show consistent gains over standard SFT on reasoning and planning benchmarks (Section 3).

2 Method

2.1 Grouping by context-sensitivity

Let (x, y) denote an input–output pair, where $y = (y_1, \ldots, y_L)$ is the target token sequence of length L over vocabulary $\mathcal{V}$. We consider masked dLLMs that operate on sequences containing a special `[MASK]` token. Given a partially masked state $z \in (\mathcal{V} \cup \{\texttt{[MASK]}\})^L$, the model produces token distributions $p_\theta(\cdot \mid x, z, j)$ over vocabulary items for each masked position j.

Confidence-gated correctness We build groups using a fixed initial model M_{initial} with parameters θ_{initial}. For each masked position j in the current state z, define the model's *top-1* prediction $\hat{y}_j$ and its confidence c_j:

$$\hat{y}_j = \arg\max_{v \in \mathcal{V}} p_{\theta_{\text{initial}}}(v \mid x, z, j)$$
$$c_j = p_{\theta_{\text{initial}}}(\hat{y}_j \mid x, z, j)$$

A masked position j is deemed *solvable* if

$$\hat{y}_j = y_j \quad \text{and} \quad c_j \geq \gamma. \tag{1}$$

The group threshold $\gamma \in [0, 1]$ controls the granularity of the resulting groups: a permissive setting (low γ) yields fewer but larger groups, while a conservative setting (high γ) restricts groups to only the most confident predictions, resulting in finer-grained groups and more frequent fallback steps.

Iterative group construction We construct an ordered list of disjoint groups $G_0, G_1, \ldots, G_{K-1}$ such that $\bigcup_k G_k = \{1, \ldots, L\}$, where lower-indexed groups contain more *context-insensitive* tokens and higher-indexed groups contain more *context-sensitive* ones. We initialize $z^{(0)}$ by masking all target positions. At iteration k, let $\mathcal{M}^{(k)}$ be the set of currently masked positions in $z^{(k)}$. We compute the *solvable* set

$$S^{(k)}(\gamma) \;=\; \{j \in \mathcal{M}^{(k)} \;:\; j \text{ satisfies Equation 1}\}.$$

If $S^{(k)}(\gamma)$ is non-empty, we set $G_k = S^{(k)}(\gamma)$ and unmask these positions to the ground truth: $z_j^{(k+1)} \leftarrow y_j$ for all $j \in G_k$. See Algorithm 1 for the complete group construction procedure.

If $S^{(k)}(\gamma)$ is empty, no position is both correct and confident under the current context. To guarantee progress, we take a **fallback** step and select the single position whose ground-truth token receives the highest probability:

$$j^\star \;=\; \arg\max_{j \in \mathcal{M}^{(k)}} p_{\theta_{\text{initial}}}(y_j \mid x, z^{(k)}, j). \tag{2}$$

We then set $G_k = \{j^\star\}$ and unmask that position to $y_{j^\star}$. This fallback rule yields a well-defined partition even when the initial model cannot confidently solve any remaining token.

2.2 TRAINING

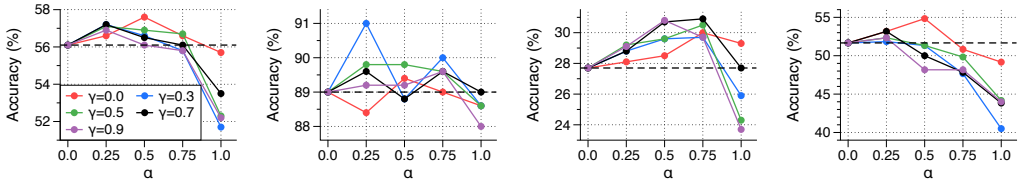
Group-ordered forward process Given ordered groups $(G_0, G_1, \ldots, G_{K-1})$ from Section 2.1, we define a forward process that corrupts the target y into a masked state $z^{(t)}$ by masking tokens in reverse group order. Context-sensitive tokens are masked first and context-insensitive tokens last, so the noised states mirror what the model encounters at inference-time using confidence-guided unmasking. Let m_t be the number of positions to mask at noise level t. We mask every token in G_{K-1}, then G_{K-2}, and so on until the cumulative count reaches m_t. The last group partially or fully affected by this process is the boundary group, denoted by the index k^*. All groups $G_{>k^*}$ are fully masked and the remaining $r_t = m_t - \sum_{i>k^*} |G_i|$ positions are sampled uniformly from G_{k^*} without replacement. Groups below the boundary stay unmasked. See Algorithm 2 for the pseudocode.

Mixing Group-based and Random Forward Processes At inference-time, the model's unmasking trajectory will not exactly follow the group trajectories, and errors can accumulate through decoding steps. For robustness, we mix the group-based forward process with a random forward process. Each training sample uses the group-based process with probability α (group selection ratio), and the random forward process otherwise.

The model is trained with the standard masked-token cross-entropy loss over positions in $\mathcal{M}(z)$ (see Appendix D.1 for the full objective). We construct groups using M_{initial} and a fixed γ for the training dataset. During SFT, masked states z are drawn by reusing these groups and sampling intra-group permutations on the fly.

Table 1: Evaluation results on *Sudoku*, *Countdown*, and *Trip Planning*. For CATS, γ=0.0 and α=0.5 are used.

	Sudoku	Countdown	Trip Plan.	Average
Few Shots	1.0	17.6	30.3	16.3
SFT	89.0	27.7	51.7	56.1
GIFT	84.8	25.8	44.0	51.5
CATS	**89.4**	**28.5**	**54.8**	**57.6**



(a) Average (b) Sudoku (c) Countdown (d) Trip Planning

Figure 3: **Effect of trajectory mixing α and group granularity γ.** Performance of CATS relative to standard SFT (dotted horizontal line at $\alpha = 0$) across all group selection ratio α and group threshold γ configurations. Mixing trajectories consistently improves robustness, with intermediate γ values yielding the greatest stability.

3 Experiments

Setup We compare against three baselines: few-shot prompting of the initial model without any fine-tuning, standard SFT with uniform random masking, and GIFT (Xu et al., 2025), which reweights the loss based on per-token entropy. We include GIFT as the closest prior work targeting the training-inference gap in dLLM SFT, albeit through loss reweighting rather than trajectory restructuring. GIFT was originally evaluated on instruction-tuning data; we apply it to downstream tasks under the same optimization setup. We evaluate on *Sudoku*, *Countdown*, and *Trip Planning*, which require global constraint satisfaction, strong token interdependence, and sensitivity to generation order (Ye et al., 2025b; Zhao et al., 2025; Ye et al., 2025a; Kang et al., 2025) [1]. Unless otherwise stated, models decode by unmasking the single most confident token per step (b=1). We report the highest accuracy for each configuration after sweeping over learning rates and training steps. Details on dataset characteristics and hyperparameters are provided in Appendix E.

Results Figure 1 and Table 1 show that CATS improves over standard SFT, particularly on *Trip Planning*. At α=0.5 and γ=0.0, the average accuracy improves from 56.1% to 57.6% (+1.5%p). This advantage stems from directly aligning training with the model's inference-time decoding behavior, rather than relying on static token-level reweighting. The gains are also preserved under parallel decoding (b=2, 4; see Appendix F). Full results across γ values and sub-tasks are provided in Tables 5 to 8.

Figure 3 summarizes the performance across all γ and α configurations. Relying solely on group-ordered trajectories (α=1.0) degrades accuracy across nearly all γ values, so mixing in random masking remains necessary. Meanwhile, intermediate γ gives the most stable gains.

4 Conclusion

We propose CATS, a method that constructs inference-aligned training trajectories for dLLM SFT from ground-truth targets. By grouping tokens based on the initial model's context-sensitivity, our approach bridges the training-inference gap without the need for rollouts or reward-based methods. Across a range of benchmarks, CATS improves average accuracy over standard SFT by 1.5%p.

[1] Math/code benchmarks are excluded because small-scale SFT can degrade performance in domains already well covered during pretraining.

ACKNOWLEDGMENTS

We are grateful to the FuriosaAI team for their dedicated support. We want to thank Kang for his invaluable mentorship and supervision during this project. We also appreciate the guidance from Prof. Lyu at UW-Madison and Dr. Lee at KRAFTON.

REFERENCES

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. The language model evaluation harness, 07 2024. URL https://zenodo.org/records/12608602.

Shansan Gong, Ruixiang Zhang, Huangjie Zheng, Jiatao Gu, Navdeep Jaitly, Lingpeng Kong, and Yizhe Zhang. Diffucoder: Understanding and improving masked diffusion models for code generation. *arXiv preprint arXiv:2506.20639*, 2025.

Haoyu He, Katrin Renz, Yong Cao, and Andreas Geiger. Mdpo: Overcoming the training-inference divide of masked diffusion language models. *arXiv preprint arXiv:2508.13148*, 2025.

Zhengfu He, Tianxiang Sun, Qiong Tang, Kuanning Wang, Xuan-Jing Huang, and Xipeng Qiu. Diffusionbert: Improving generative masked language models with diffusion models. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: Long papers)*, pp. 4521–4534, 2023.

Wonjun Kang, Kevin Galim, Seunghyuk Oh, Minjae Lee, Yuchen Zeng, Shuibai Zhang, Coleman Hooper, Yuezhou Hu, Hyung Il Koo, Nam Ik Cho, et al. Parallelbench: Understanding the tradeoffs of parallel decoding in diffusion llms. *arXiv preprint arXiv:2510.04767*, 2025.

Jaeyeon Kim, Kulin Shah, Vasilis Kontonis, Sham M. Kakade, and Sitan Chen. Train for the worst, plan for the best: Understanding token ordering in masked diffusions. In *Forty-second International Conference on Machine Learning*, 2025a. URL https://openreview.net/forum?id=DjJmre5IkP.

Minseo Kim, Chenfeng Xu, Coleman Hooper, Harman Singh, Ben Athiwaratkun, Ce Zhang, Kurt Keutzer, and Amir Gholami. Cdlm: Consistency diffusion language models for faster sampling. *arXiv preprint arXiv:2511.19269*, 2025b.

Sourab Mangrulkar, Sylvain Gugger, Lysandre Debut, Younes Belkada, Sayak Paul, Benjamin Bossan, and Marian Tietz. PEFT: State-of-the-art parameter-efficient fine-tuning methods. https://github.com/huggingface/peft, 2022.

Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, JUN ZHOU, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. Large language diffusion models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=KnqiC0znVF.

Yu-Yang Qian, Junda Su, Lanxiang Hu, Peiyuan Zhang, Zhijie Deng, Peng Zhao, and Hao Zhang. d3llm: Ultra-fast diffusion llm using pseudo-trajectory distillation. *arXiv preprint arXiv:2601.07568*, 2026.

Subham Sekhar Sahoo, Marianne Arriola, Aaron Gokaslan, Edgar Mariano Marroquin, Alexander M Rush, Yair Schiff, Justin T Chiu, and Volodymyr Kuleshov. Simple and effective masked diffusion language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=L4uaAR4ArM.

Jiaxin Shi, Kehang Han, Zhe Wang, Arnaud Doucet, and Michalis Titsias. Simplified and generalized masked diffusion for discrete data. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=xcqSOfHt4g.

Yuxuan Song, Zheng Zhang, Cheng Luo, Pengyang Gao, Fan Xia, Hao Luo, Zheng Li, Yuehang Yang, Hongli Yu, Xingwei Qu, et al. Seed diffusion: A large-scale diffusion language model with high-speed inference. *arXiv preprint arXiv:2508.02193*, 2025.

Renzhi Wang, Jing Li, and Piji Li. Infodiffusion: Information entropy aware diffusion process for non-autoregressive text generation. In *The 2023 Conference on Empirical Methods in Natural Language Processing*, 2023. URL https://openreview.net/forum?id=8IrFLWRvuW.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 38–45, Online, October 2020. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/2020.emnlp-demos.6.

Guowei Xu, Wenxin Xu, Jiawang Zhao, and Kaisheng Ma. Gift: Guided importance-aware finetuning for diffusion language models. *arXiv preprint arXiv:2509.20863*, 2025.

Jiacheng Ye, Jiahui Gao, Shansan Gong, Lin Zheng, Xin Jiang, Zhenguo Li, and Lingpeng Kong. Beyond autoregression: Discrete diffusion for complex reasoning and planning. In *The Thirteenth International Conference on Learning Representations*, 2025a. URL https://openreview.net/forum?id=NRYgUzSPZz.

Jiacheng Ye, Zhihui Xie, Lin Zheng, Jiahui Gao, Zirui Wu, Xin Jiang, Zhenguo Li, and Lingpeng Kong. Dream 7b: Diffusion large language models. *arXiv preprint arXiv:2508.15487*, 2025b.

Zheng Yuan, Hongyi Yuan, Chengpeng Li, Guanting Dong, Keming Lu, Chuanqi Tan, Chang Zhou, and Jingren Zhou. Scaling relationship on learning mathematical reasoning with large language models, 2023. URL https://arxiv.org/abs/2308.01825.

Siyan Zhao, Devaansh Gupta, Qinqing Zheng, and Aditya Grover. d1: Scaling reasoning in diffusion large language models via reinforcement learning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=7ZVRlBFuEv.

Huaixiu Steven Zheng, Swaroop Mishra, Hugh Zhang, Xinyun Chen, Minmin Chen, Azade Nova, Le Hou, Heng-Tze Cheng, Quoc V Le, Ed H Chi, et al. Natural plan: Benchmarking llms on natural language planning. *arXiv preprint arXiv:2406.04520*, 2024.

A Related Work

Masked diffusion language models (Sahoo et al., 2024; Shi et al., 2024) are typically trained using a random forward masking schedule, where token positions are uniformly masked, and the model is optimized to recover the original sequence from the masked input. However, such random schedules may not provide optimal supervision for learning the backward unmasking process. DiffusionBERT (He et al., 2023) and InfoDiffusion (Wang et al., 2023) introduce token-aware forward diffusion processes that adjust each token's noise level based on corpus-level statistics (e.g., information entropy). While such token-level metrics may suffice for pre-training, they are ill-suited for SFT, particularly when the model must acquire specific reasoning or planning capabilities, because they do not account for trajectory-level dependencies during progressive unmasking. GIFT (Xu et al., 2025) introduces an importance-aware SFT strategy that assigns entropy-based token importance weights, so that high-uncertainty tokens receive stronger training signals during SFT. Although GIFT also aims to bridge the training-inference gap, it operates at the loss-weighting level without restructuring the forward masking process itself. In contrast, our approach directly aligns the training trajectory with the model's intrinsic decoding behavior by reordering which tokens are masked and revealed during training.

Concurrent work Seed Diffusion (Song et al., 2025) is an early attempt at trajectory-aware post-training for diffusion LLMs, but its algorithmic details are not publicly available. Meanwhile, d3LLM (Qian et al., 2026) proposes pseudo-trajectory distillation based on the teacher model's own decoding trajectories, which are not aligned with ground-truth generation paths. In contrast, we construct ground-truth-aligned trajectories for SFT. Moreover, while d3LLM primarily targets faster inference, our focus is on improving accuracy.

B Limitations

CATS constructs groups from the initial model once before training, so the context-sensitivity estimates remain static and may become stale as the model improves. Additionally, group-ordered trajectories alone (α=1.0) degrade accuracy, requiring mixing with random masking and tuning of the additional hyperparameter α. Finally, we evaluate on three reasoning and planning benchmarks; generalization to other domains remains to be verified.

C Future Work

Our current approach constructs groups from the initial model only once before training begins. A natural extension is to re-collect groups at intermediate checkpoints so that context-sensitivity estimates can evolve with the model during training.

Although we focus on SFT, CATS is also applicable to pre-training. One possible direction is to begin with random masking and gradually increase the group selection ratio during training, while iteratively constructing groups from the current checkpoint so that the forward process adapts with the model's predictions.

D Details of CATS

D.1 Training Objective

Given a sampled masked state z and the corresponding ground-truth tokens at masked positions, we apply the standard token-level denoising loss used in discrete diffusion SFT:

$$\mathcal{L}(\theta) = \mathbb{E}_{(x,y)\sim\mathcal{D},z\sim q_{\text{mix}}(\cdot|x,y)}\left[\sum_{j\in\mathcal{M}(z)} -\log p_\theta(y_j \mid x, z, j)\right], \tag{3}$$

where q_{mix} denotes the mixture of random and group-ordered forward processes and $\mathcal{M}(z)$ is the set of masked positions in state z.

D.2 INTERPRETATION OF GROUP CONSTRUCTION

By construction, groups are ordered from context-insensitive to context-sensitive positions: G_0 contains tokens solvable with minimal context, and later groups contain tokens that become solvable only after earlier groups are revealed. In practice, trivial suffix regions (e.g., `[PAD]`) tend to appear in early groups, while content-bearing and constraint-critical tokens are deferred.

D.3 ALGORITHMS

The detailed procedures for group construction and trajectory sampling are provided in Algorithm 1 and Algorithm 2, respectively.

Algorithm 1: Group Collection via Confidence-Gated Correctness

Input: Input x, target sequence $y = (y_1, \ldots, y_L)$, initial model M_{initial} with θ_{initial}, group threshold γ
Output: Ordered disjoint groups $(G_0, \ldots, G_{K-1})$ such that $\bigcup_k G_k = \{1, \ldots, L\}$
Initialize $z^{(0)} \leftarrow$ (`[MASK]`, ..., `[MASK]`);
$k \leftarrow 0$;
while $\exists\, j \in \{1, \ldots, L\} : z_j^{(k)} =$ *`[MASK]`* **do**
  $\mathcal{M}^{(k)} \leftarrow \{j \in \{1, \ldots, L\} : z_j^{(k)} =$ `[MASK]`$\}$;
  foreach $j \in \mathcal{M}^{(k)}$ **do**
    $\hat{y}_j \leftarrow \arg\max_{v \in \mathcal{V}} p_{\theta_{\text{initial}}}(v \mid x, z^{(k)}, j)$;
    $c_j \leftarrow p_{\theta_{\text{initial}}}(\hat{y}_j \mid x, z^{(k)}, j)$;
  $S^{(k)}(\gamma) \leftarrow \{j \in \mathcal{M}^{(k)} : \hat{y}_j = y_j \ \wedge\ c_j \geq \gamma\}$; `// Eq. 1`
  if $S^{(k)}(\gamma) \neq \emptyset$ **then**
    $G_k \leftarrow S^{(k)}(\gamma)$;
  else
    $j^\star \leftarrow \arg\max_{j \in \mathcal{M}^{(k)}} p_{\theta_{\text{initial}}}(y_j \mid x, z^{(k)}, j)$; `// Eq. 2`
    $G_k \leftarrow \{j^\star\}$;
  foreach $j \in G_k$ **do**
    $z_j^{(k+1)} \leftarrow y_j$; `// teacher-forced unmasking`
  $k \leftarrow k + 1$;
return $(G_0, \ldots, G_{k-1})$;

E DETAILS OF EXPERIMENTS

E.1 TRAINING SETUP

Tasks We evaluate on three representative benchmarks for discrete diffusion reasoning and planning: (i) **Sudoku** (4×4) (Zhao et al., 2025), which requires filling a partially specified grid under row/column/subgrid constraints; (ii) **Countdown** (N3–N5) (Ye et al., 2025a), which requires composing arithmetic operations over given operands to reach a target; and (iii) **Trip Planning** (TP3–TP5) (Zheng et al., 2024; Ye et al., 2025b), which requires constructing a feasible multi-day itinerary under flight connectivity and duration constraints. We follow the standard evaluation protocols provided by prior work and report task accuracy (exact-match correctness). Correctness is deterministic in all cases, and these benchmarks are standard in prior works (Ye et al., 2025b; Kang et al., 2025).

Models and training We use the same initial diffusion language model, Dream 7B Instruct (Ye et al., 2025b), for all methods and benchmarks. **SFT** ($\alpha = 0.0$) denotes standard supervised fine-tuning using only the random forward noising process. **CATS** fine-tunes the same initial model using a mixture of random and group-ordered forward processes; we denote the group selection ratio by α (e.g., **CATS** with $\alpha = 0.25$). Groups are collected once using M_{initial} with a group threshold γ,

Algorithm 2: Forward Process with Context-sensitivity Aware Trajectory Sampling

Input: Input x, target sequence $y = (y_1, \ldots, y_L)$, timestep t, # tokens to mask m_t, ordered disjoint groups $(G_0, \ldots, G_{K-1})$
Output: Noised target sequence z for timestep t
Initialize $z \leftarrow (y_1, \ldots, y_L)$;

for $k \leftarrow 0$ **to** $K-1$ **do**
  $s_k \leftarrow \sum_{i=k+1}^{K-1} |G_i|$; `// # tokens in strictly-later groups`

// Find boundary group k **s.t.** $s_k \le m_t < s_k + |G_k|$;
$k \leftarrow 0$;
while $k < K-1$ ***and*** $s_k > m_t$ **do**
  $k \leftarrow k+1$;

// Sample additional mask positions inside G_k;
$n \leftarrow m_t - s_k$;
if $n > 0$ **then**
  $P \leftarrow$ UNIFORMSAMPLE(G_k, n) ; `// sample` n `distinct indices from` G_k
else
  $P \leftarrow \emptyset$;

// Mask selected positions: all later groups + sampled subset in G_k;
foreach $i \in \left(\bigcup_{j=k+1}^{K-1} G_j\right) \cup P$ **do**
  $z_i \leftarrow$ `[MASK]`;
return z;

and are then used to sample trajectories during SFT. **GIFT** (Xu et al., 2025) is trained with the same optimization setup described above (learning rate sweep, training steps).

Optimization For each setting, we train for 1000 steps and consider learning rates in $\{1e{-}5,\ 2e{-}5,\ 5e{-}5\}$. We evaluate every 200 steps and report the best-performing checkpoint for each method and benchmark. Table 2 summarizes the common training hyperparameters.

Table 2: Common training hyperparameters used across all experiments.

Hyperparameter	**Value**
Effective batch size	64
Batch size per GPU	8
Number of GPUs	8
LoRA rank	32
LoRA α	32
LoRA target modules	`q, k, v, o, up, down, gate`
β (Adam)	0.9, 0.95
Weight decay	0.01
Warmup ratio	0.1
Learning rate scheduler	Constant

Computing infrastructure All experiments are conducted on a single node equipped with eight NVIDIA A100 80GB GPUs. Each 1000-step training run takes approximately one hour, and evaluating five checkpoints requires a similar duration. Group construction takes 30–60 minutes per dataset. Including all hyperparameter sweeps, the total compute budget across 100+ runs is approximately 800 A100 GPU-hours.

Software and licenses We build on the official Dream codebase (Ye et al., 2025b) with HuggingFace Transformers (Wolf et al., 2020) and PEFT (Mangrulkar et al., 2022) for LoRA fine-tuning, and

Table 3: Evaluation results with $b = 2, 4$. For CATS, $\gamma = 0.0$ and $\alpha = 0.5$ are used.

	b	Sudoku	Countdown	Trip Plan.	Average
SFT	2	86.2	**30.9**	51.5	56.2
CATS	2	**88.4**	27.3	**57.3**	**57.4**
SFT	4	78.8	**28.8**	48.3	52.0
CATS	4	**79.2**	26.3	**55.0**	**53.5**

evaluate with a customized fork of lm-eval (Gao et al., 2024). Dream 7B Instruct and all datasets (*Sudoku* (Zhao et al., 2025), *Countdown* (Ye et al., 2025a), *Trip Planning* (Ye et al., 2025b)) are released under the Apache-2.0 license.

Group threshold When constructing groups, we mark a masked position as *solvable* only if the initial model's *top-1* prediction matches the ground truth and its confidence is at least $\gamma \in [0, 1]$; otherwise, it is treated as unsolved. We sweep $\gamma \in \{0.0, 0.3, 0.5, 0.7, 0.9\}$ to study how γ affects downstream training.

E.2 Evaluation setup

Generation length For each benchmark, we choose the generation length to exceed the maximum ground-truth completion length while minimizing padding for shorter samples. Concretely, we set the generation length to 32 for *Sudoku*, 64 for *Countdown N3/4/5*, 128 for *Trip Planning N3*, and 192 for *Trip Planning N4/5*.

Decoding strategy At each decoding step, the b most confident masked tokens are unmasked simultaneously. Our default setting is b=1 (one-by-one decoding), which keeps evaluation deterministic and keeps the decoding cost identical across methods. We additionally evaluate b=2, 4 to measure robustness under parallel decoding (Table 3).

Few-shot evaluation For few-shot evaluation on initial models, we use 8-shot prompting for *Sudoku* and all *Countdown* tasks, and 3-shot prompting for all *Trip Planning* tasks.

F Parallel Decoding Results

We evaluate parallel decoding with $b = 2, 4$ tokens unmasked per step. The gains from CATS are largely preserved under parallel decoding, with consistent improvements on *Sudoku* and *Trip Planning*.

G Complete Evaluation Results

We report the evaluation results for all sub-tasks in *Sudoku*, *Countdown*, and *Trip Planning*, across all γ and α values. **Bold** and underline indicate the best and the second-best results, respectively.

Table 4: Evaluation results on *Sudoku*, *Countdown*, and *Trip Planning* with $\gamma = 0.0$.

	α	Sudoku	CD3	CD4	CD5	TP3	TP4	TP5
Few Shots	N/A	1.0	43.6	8.4	0.9	55.5	23.5	12.0
SFT	0.0	<u>89.0</u>	65.5	**15.2**	**2.5**	80.0	47.5	27.5
GIFT	N/A	84.8	68.2	7.7	1.5	74.0	37.5	20.5
CATS	0.25	88.4	69.8	12.3	<u>2.3</u>	**83.5**	47.5	**28.5**
CATS	0.5	**89.4**	71.9	11.6	1.9	<u>82.5</u>	**54.0**	<u>28.0</u>
CATS	0.75	<u>89.0</u>	<u>73.6</u>	<u>14.0</u>	<u>2.3</u>	78.0	<u>49.5</u>	25.0
CATS	1.0	88.6	**76.0**	11.2	0.6	78.0	48.0	21.5

Table 5: Evaluation results on *Sudoku*, *Countdown*, and *Trip Planning* with $\gamma = 0.3$.

	α	Sudoku	CD3	CD4	CD5	TP3	TP4	TP5
Few Shots	N/A	1.0	43.6	8.4	0.9	55.5	23.5	12.0
SFT	0.0	89.0	65.5	**15.2**	2.5	80.0	**47.5**	**27.5**
GIFT	N/A	84.8	68.2	7.7	1.5	74.0	37.5	20.5
CATS	0.25	**91.0**	71.0	13.0	2.3	**82.0**	47.0	26.5
CATS	0.5	88.8	73.7	12.4	**2.6**	81.0	45.5	**27.5**
CATS	0.75	90.0	**76.0**	11.4	1.6	78.0	41.5	23.5
CATS	1.0	88.6	67.6	9.2	1.0	70.5	31.0	20.0

Table 6: Evaluation results on *Sudoku*, *Countdown*, and *Trip Planning* with $\gamma = 0.5$.

	α	Sudoku	CD3	CD4	CD5	TP3	TP4	TP5
Few Shots	N/A	1.0	43.6	8.4	0.9	55.5	23.5	12.0
SFT	0.0	89.0	65.5	**15.2**	**2.5**	80.0	47.5	27.5
GIFT	N/A	84.8	68.2	7.7	1.5	74.0	37.5	20.5
CATS	0.25	**89.8**	71.7	13.8	2.2	80.5	**49.0**	27.5
CATS	0.5	**89.8**	74.2	12.3	2.3	**82.0**	43.0	**29.0**
CATS	0.75	89.6	**75.5**	14.2	1.9	79.5	45.0	25.0
CATS	1.0	88.6	63.9	6.8	2.1	75.5	37.0	20.0

Table 7: Evaluation results on *Sudoku*, *Countdown*, and *Trip Planning* with $\gamma = 0.7$.

	α	Sudoku	CD3	CD4	CD5	TP3	TP4	TP5
Few Shots	N/A	1.0	43.6	8.4	0.9	55.5	23.5	12.0
SFT	0.0	89.0	65.5	**15.2**	2.5	80.0	47.5	27.5
GIFT	N/A	84.8	68.2	7.7	1.5	74.0	37.5	20.5
CATS	0.25	**89.6**	70.5	13.3	2.5	**80.5**	**49.5**	**29.5**
CATS	0.5	88.8	**77.5**	12.0	**2.7**	77.5	45.5	27.0
CATS	0.75	**89.6**	76.2	14.2	2.4	76.5	42.5	24.5
CATS	1.0	89.0	71.4	10.1	1.7	76.0	36.5	19.0

Table 8: Evaluation results on *Sudoku*, *Countdown*, and *Trip Planning* with $\gamma = 0.9$.

	α	Sudoku	CD3	CD4	CD5	TP3	TP4	TP5
Few Shots	N/A	1.0	43.6	8.4	0.9	55.5	23.5	12.0
SFT	0.0	89.0	65.5	**15.2**	2.5	**80.0**	47.5	27.5
GIFT	N/A	84.8	68.2	7.7	1.5	74.0	37.5	20.5
CATS	0.25	89.2	71.9	12.9	2.5	79.5	**49.0**	**28.5**
CATS	0.5	89.2	**74.9**	14.9	**2.7**	76.5	43.5	24.5
CATS	0.75	**89.6**	74.1	12.3	2.6	76.5	44.0	24.0
CATS	1.0	88.0	59.7	9.3	2.2	74.0	36.0	22.0

H BENCHMARK EXAMPLES

Table 9: A sample question and response pair for *Sudoku*

Question	Fill the positions where the values are 0 in a 4x4 grid with digits 1-4 so that each column, each row, and each of the four 2x2 subgrids that compose the grid contains all of the digits from 1 to 4. Puzzle: 0321003004002100
Response	4321123434122143

Table 10: A sample question and response pair for *Countdown N3*

Question	Given 4 numbers, use +-*/ to operate over the first 3 numbers to achieve the last number. Numbers: 44,2,54,64
Response	2*54=108,108-44=64

Table 11: A sample question and response pair for *Trip Planning N3*

Question	You plan to visit 3 European cities for 14 days in total. You only take direct flights to commute between cities. You would like to visit Florence for 6 days. You want to meet a friend in Florence between day 9 and day 14. You would like to visit Barcelona for 5 days. You would like to visit Helsinki for 5 days. Here are the cities that have direct flights: Barcelona and Florence, Helsinki and Barcelona. Find a trip plan of visiting the cities for 14 days by taking direct flights to commute between them.
Response	Here is the trip plan for visiting the 3 European cities for 14 days: **Day 1-5:** Arriving in Helsinki and visit Helsinki for 5 days. **Day 5:** Fly from Helsinki to Barcelona. **Day 5-9:** Visit Barcelona for 5 days. **Day 9:** Fly from Barcelona to Florence. **Day 9-14:** Visit Florence for 6 days.