FAST-TO-SLOW THINKING COMES NATURALLY TO MULTIMODAL DIFFUSION LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Diffusion language models (DLLMs) enable parallel generation by denoising multiple tokens at arbitrary masked positions, with existing studies primarily exploiting this property to accelerate inference. Beyond efficiency, the flexibility to generate different numbers of tokens at each step naturally supports different thinking speeds, yet its potential for fast-to-slow thinking remains unexplored. We exploit this inherent property of DLLMs for fast-to-slow thinking, without specially designed training or additional denoising steps compared with standard decoding. We first obtain a complete thought through fast denoising, then remask uncertain tokens for slow refinement while retaining confident predictions as context. We further extend this idea to multimodal DLLMs, where fast and slow thinking favor different visual conditions: rich visual details can distract aggressive fast denoising but benefit slow refinement. We thus introduce EdgeDrop, which suppresses regions with complex edge information during fast thinking and restores the whole image for slow refinement. Extensive experiments across four multimodal benchmarks demonstrate consistent performance improvements with comparable overall inference time. Notably, the fast stage alone achieves a $2\times$ speedup while outperforming standard decoding on SQA-IMG and MMBench, with subsequent slow refinement yielding further gains on M3CoT and V*.

1 INTRODUCTION

Recent work has explored fast and slow thinking in autoregressive models (AR), allowing models to answer quickly when little reasoning is needed and spend more time on more demanding problems Chung et al. (2025); Lou et al. (2025). As AR models generate one token at a time in order, fast and slow computation is mainly reflected by shorter or longer reasoning traces, which are learned through supervised fine-tuning on chains of thought varying in length. Switching between the two modes further relies on specially constructed reasoning data or additional mechanisms for mode selection and control Lin et al. (2025); Cheng et al. (2025). Diffusion language models (DLLMs) Sahoo et al. (2024); Gat et al. (2024); Shi et al. (2024), instead, generate multiple tokens in parallel, providing another way to control the thinking speed: the number of tokens denoised at each step (more tokens, less time, and vice versa). In this work, we exploit this inherent flexibility of DLLMs to instantiate *fast-to-slow thinking directly during inference, without specially designed training recipes or datasets*.

Recent advances in DLLMs have demonstrated promising performance on reasoning and coding tasks Nie et al. (2026); Ye et al. (2025); Khanna et al. (2025); Gong et al. (2025), with much of the progress centered on exploiting parallel generation and bidirectional attention Wu et al. (2026b;a); He et al. (2026); Chen et al. (2026b). By controlling the number of tokens denoised at each step, DLLMs naturally support different thinking speeds. Yet, how to translate this inherent property into a systematic fast-to-slow thinking mechanism remains underexplored in existing literature.

Instantiating fast-to-slow thinking in DLLMs faces two key challenges: First, **error accumulation**: fast thinking requires aggressive decoding, where more tokens are unmasked at each step. However, under the standard irreversible denoising schedule, once a token is unmasked, it remains fixed in subsequent steps. Inaccurate predictions made early can persist and affect subsequent thinking, leading to error accumulation throughout denoising Liu et al. (2024a); Xu et al. (2024); Kang et al. (2025). Recent methods correct inaccurate predictions by interleaving token remasking and

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

correction within a single denoising trajectory (correction before completion) Wang et al. (2026); Zhai et al. (2026); Ouyang et al. (2026); Peng et al. (2025). In multimodal settings, existing work further relies on external visual experts to identify insufficient visual evidence and revise model outputs Jeong et al. (2026). In contrast, we defer token revision until the fast denoising trajectory is complete, requiring no external experts or training; all revision is performed within the denoising process itself. Specifically, we first obtain a complete fast prediction and then remask the uncertain tokens for slow refinement, with the confident tokens retained as context. For example, given 64 masked tokens, we first denoise 4 tokens per step, then remask the 32 most uncertain tokens and refine them at 2 tokens per step. This fast-to-slow process requires the same 32 denoising steps as standard decoding with 2 tokens per step (16 fast+16 slow *vs.* 32 standard), without additional inference steps.

Second, **visual condition mismatch**: for multimodal DLLMs, different thinking modes must also be conditioned on visual information. We find that simply switching the denoising granularity is not sufficient, and fast and slow thinking require different visual conditions. For fast thinking, rich local details can distract the aggressive denoising process and lead to less reliable predictions. We therefore suppress regions with complex edge information, allowing the model to focus more on the big picture. Interestingly, our fast stage alone already outperforms standard decoding on SQA-IMG and MMBench (with $2\times$ speedup), where many questions require relatively direct answering. For slow thinking, it benefits from the whole image, where some visual information supports the confident text predictions from fast thinking, while additional details help reconsider the uncertain tokens (see Tab.4 for other visual conditions, *e.g.*, image cropping and noise perturbation). With further slow thinking, the gains become more pronounced on M3CoT and V* (+2.63 and +3.14), where questions require more multi-step reasoning or fine-grained visual evidence.

Our contributions are summarized as follows:

- We introduce fast-to-slow thinking for the DLLM (F2S-DLLM), exploiting its inherent denoising flexibility to vary thinking speed directly at inference time, without specially designed training recipes or datasets for learning different thinking modes.
- We extend this idea to multimodal DLLMs and study how visual information should condition different thinking modes. We show that fast and slow thinking benefit from different visual conditions, revealing a dimension beyond controlling the denoising granularity.
- We design EdgeDrop, an edge-aware visual conditioning strategy that suppresses regions with complex edge information during fast thinking, allowing the model to focus on the big picture, while the whole image is restored for slow refinement.
- Extensive experiments on four multimodal benchmarks show that F2S-DLLM consistently improves performance while maintaining comparable overall inference time. The fast stage alone achieves a $2\times$ speedup and outperforms standard decoding on SQA-IMG and MMBench, while slow refinement further improves M3CoT and V*.

2 Related Work

Diffusion Language Models. Diffusion models were first developed for continuous data generation and have achieved remarkable success in image synthesis Ramesh et al. (2022); Rombach et al. (2022); Sohl-Dickstein et al. (2015); Song & Ermon (2019); Ho et al. (2020); Sahoo et al. (2023). Subsequent studies extended diffusion processes to text generation in continuous representation spaces Gulrajani & Hashimoto (2023); Dieleman et al. (2022). To accommodate the discrete nature of text, discrete diffusion models instead define the corruption and denoising processes directly over a finite vocabulary Li et al. (2022); Austin et al. (2021); Sahoo et al. (2024); Lou et al. (2024). More recently, large-scale masked diffusion language models Nie et al. (2026); Ye et al. (2025) generate sequences by iteratively resolving masked positions from a fully masked sequence. Unlike autoregressive decoding, DLLMs support flexible generation orders and maintain a global intermediate sequence state at each denoising step. This paradigm has demonstrated competitive performance on language understanding, reasoning, and code generation tasks. However, how the intermediate state formed influences subsequent denoising and reasoning remains underexplored.

Recent studies have incorporated visual modalities into diffusion language models, extending parallel denoising to visual understanding and reasoning. Representative models include LaViDa Li et al.

(2026), LLaDA-V You et al. (2026), MMaDA Yang et al. (2026), and Dimple Yu et al. (2025). These models combine visual inputs with diffusion decoders, enabling parallel generation for multimodal reasoning and understanding. However, their decoding process still largely follows progressive unmasking, where generated tokens are difficult to revise once fixed. This motivates multimodal diffusion decoding mechanisms that can break irreversible generation and rectify early mistakes.

Fast and Slow Thinking. Human cognition is commonly characterized by two complementary modes: System 1 is fast, automatic, and intuitive, whereas System 2 is slower, effortful, and analytical (Hua & Zhang, 2022). A similar distinction appears in language-model reasoning, ranging from direct/short answer generation to deliberate multi-step reasoning (Wei et al., 2022; Wang et al., 2022; Zhou et al., 2022; Zelikman et al., 2022). Although extended reasoning can improve performance on difficult problems, it also introduces unnecessary computation for questions that can be answered directly. Models deployed across diverse real-world tasks need to allocate reasoning computation according to the difficulty and uncertainty of each input. Existing approaches typically estimate this need using heuristics like query complexity or uncertainty signals. Alternatively, they train routers to dynamically select appropriate models, decoding strategies, or reasoning paths (Ong et al., 2025; Chen et al., 2023; He et al., 2025; Schuster et al., 2022).

These approaches are primarily developed for AR models, where reasoning intensity is controlled through the length and structure of the sequential reasoning trace. Switching between fast and slow modes therefore often relies on specially constructed reasoning data, learned stopping criteria, or additional routing mechanisms. Diffusion language models (DLLMs) offer a complementary inference-time control: because they can denoise multiple masked positions in parallel, the number of tokens committed at each step directly determines the granularity of generation. Denoising more tokens per step supports rapid prediction, whereas updating fewer tokens allows computation to be concentrated on uncertain positions. However, existing DLLM decoding typically uses a fixed granularity throughout generation and does not explicitly transition between these two modes. Based on this observation, we propose F2S-DLLM, which first forms a complete prediction through fast parallel denoising and then remasks low-confidence tokens for slower, context-aware refinement. This formulation realizes fast-to-slow thinking without specialized training or additional denoising steps, while allowing the visual condition to be adapted to each thinking stage in multimodal settings.

Self-correction and Refinement. While standard masked diffusion decoding enables parallel generation by progressively filling masked positions, it causes a revision bottleneck where decoded tokens are typically fixed in later steps. Recent work has therefore explored remasking and correction mechanisms to improve the revisability of masked diffusion models. Existing work introduces remasking to relax this constraint. Methods such as ReMDM Wang et al. (2026); Zhai et al. (2026); Ouyang et al. (2026) use internal signals such as confidence scores or probability margins to revise tokens during the denoising trajectory, while P2-Self Peng et al. (2025) separates planning and denoising within each decoding step. Training-based approaches, including PRISM Kim et al. (2025), RemeDi Huang et al. (2025), and GIDD Von Rütte et al. (2025), instead learn explicit correction behavior through specialized objectives, policies, or auxiliary modules. These methods primarily improve revisability within a single denoising trajectory, rather than using revision to explicitly transition between different decoding granularities.

Revision after completing an initial response has also been explored in adjacent settings. Speculative Correction Chen et al. (2026a) applies bidirectional diffusion refinement to complete text responses, primarily targeting quality–latency trade-offs, while Draft and Refine with visual experts Jeong et al. (2026) uses feedback from external visual experts to revise autoregressive vision-language outputs. Together, these studies show that a completed prediction can serve as the starting point for subsequent refinement, but revision is still primarily treated as a mechanism for correcting unreliable outputs. In contrast, F2S-DLLM uses remasking to transition between fast and slow inference: it generates a fast prediction via parallel denoising, then reallocates computation to refine uncertain tokens. This remasking explicitly shifts decoding from coarse generation to targeted refinement, enabling fast-to-slow inference without extra training, external models, or additional denoising steps.

3 Fast-to-Slow Thinking in Multimodal DLLMs

In this section, we present F2S-DLLM, a training-free inference framework that brings fast-to-slow thinking to multimodal diffusion language models. Standard decoding follows a single denoising

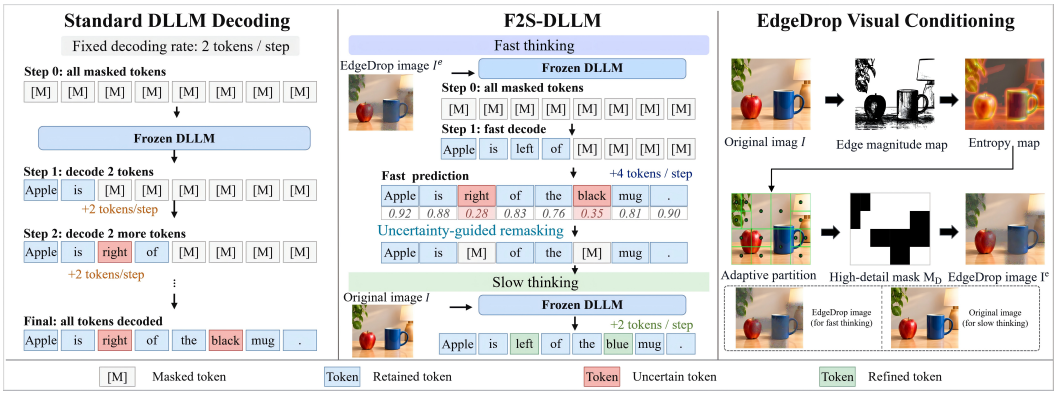


Figure 1: Overview of F2S-DLLM. The framework divides inference into fast and slow stages. In the fast stage, the model quickly generates an answer using an EdgeDrop image I^e (built from edge and entropy maps) and remasks uncertain tokens. In the slow stage, these remasked tokens are refined using the original image I for the final prediction.

granularity, progressively unmasking a fixed number of tokens at each step and keeping them unchanged once decoded. In contrast, F2S-DLLM dynamically changes the denoising granularity across two stages: *fast thinking* for forming a complete prediction, followed by *slow thinking* for selectively refining uncertain tokens. Specifically, during fast thinking, F2S-DLLM adopts a coarse denoising granularity, decoding more tokens per step (4 by default) to quickly form a complete prediction, conditioned on the visual input constructed by EdgeDrop. Once the fast prediction is complete, the model identifies and remasks low-confidence tokens while retaining confident tokens as context. It then switches to a fine denoising granularity, decoding fewer tokens per step (2 by default) to refine the uncertain predictions. Figure 1 provides an overview of the framework.

3.1 Fast-to-Slow Thinking

Given an input image I and a text prompt q, F2S-DLLM first generates a complete fast prediction conditioned on an EdgeDrop image I^e, and then corrects low-confidence positions under the guidance of the retained global context and the original image I. Starting from a fully masked sequence $x^{(0)} = [\texttt{MASK}]^L$ of length L, the model generates an initial complete response within T_f fast denoising steps. At each drafting step t, the model predicts the distribution over all remaining masked positions, and progressively unmasks a subset of tokens with the highest confidence according to a scheduling scheme. After T_f steps, a fully unmasked initial answer $x^{\text{fast}} = (x_1, \ldots, x_L)$ is obtained.

To locate uncertain tokens without introducing extra computation, we directly reuse the logits $\boldsymbol{\ell}^{(T_d)}$ from the final fast step. We compute a confidence score s_l for each predicted token and select the K lowest-confidence positions to form the remasking set $\mathcal{R}$. These tokens are reset to $[\texttt{MASK}]$, while the remaining high-confidence tokens are kept fixed as a contextual scaffold. In the subsequent T_s slow thinking steps, the model decodes only over the remasked set $\mathcal{R}$, progressively unmasking r_j tokens at each step. Because the overall response structure is already in place, the model can revise these uncertain tokens using the retained fast prediction and the full image as context. This fast-to-slow process mitigates the compounding errors common in standard decoding while preserving the same overall denoising budget.

3.2 Complex-Edge Suppression

F2S-DLLM uses different visual conditions for fast and slow thinking. During fast thinking, the model fills many tokens at each step, where intricate local details may introduce noise or lead to premature commitment. Motivated by this intuition, we introduce EdgeDrop to construct image I^e by suppressing regions with complex edge information. During slow thinking, the original image I is restored, allowing remasked uncertain tokens to be revised.

To construct the EdgeDrop image, we first identify complex-edge regions through local gradient entropy. Given an input image I defined on the spatial domain Ω, we first convert it into a normalized grayscale representation G and initialize the candidate pool as $\mathcal{P} \leftarrow \{\Omega\}$. For any candidate region

Algorithm 1 F2S-DLLM Decoding

Require: Model f_θ, sequence length L, fast steps T_f, slow steps T_s, remask budget K, EdgeDrop image I^e, original image I
Ensure: Final sequence x.
Initialize sequence $x \leftarrow [\text{MASK}]^L$
Compute fast schedule $\{b_i\}_{i=1}^{T_f}$ such that $\sum_{i=1}^{T_f} b_i = L$
Stage 1: Fast Thinking
for $i = 1$ **to** T_f **do**
  $\mathbf{P} \leftarrow \text{Softmax}(f_\theta(x, I^e))$ *// Parallel prediction*
  Select top-b_i most confident positions in $\mathcal{M}(x)$ *// Confidence ranking*
  $x_l \leftarrow \arg\max(\mathbf{P}_l)$ for selected positions *// Progressive unmasking*
end for
Cache final token confidences $\{s_l\}_{l=1}^{L}$ *// Reuse final fast step logits without extra forward pass*
Stage 2: Uncertainty-Guided Remasking
$\mathcal{R} \leftarrow \text{TopK}_{l \in \{1,\dots,L\}}(-s_l, K)$ *// Identify K lowest-confidence tokens*
$x_l \leftarrow [\text{MASK}]$ for all $l \in \mathcal{R}$ *// Remask uncertain positions*
Stage 3: Slow Thinking
Compute slow schedule $\{r_j\}_{j=1}^{T_s}$ such that $\sum_{j=1}^{T_r} r_j = K$
for $j = 1$ **to** T_s **do**
  $\mathbf{P} \leftarrow \text{Softmax}(f_\theta(x, I))$ *// Use original image and retained context*
  Select top-r_j most confident positions in $\mathcal{M}(x) \cap \mathcal{R}$ *// Rank candidate revisions*
  $x_l \leftarrow \arg\max(\mathbf{P}_l)$ for selected positions *// Fill refined tokens*
end for
return x

$\mathcal{R} \in \mathcal{P}$, we quantify its visual complexity by the entropy of its gradient magnitude distribution. Specifically, for each pixel $p \in \mathcal{R}$, we compute the spatial gradient magnitude:

$$M(p) = \sqrt{\big(\nabla_x G(p)\big)^2 + \big(\nabla_y G(p)\big)^2}, \tag{1}$$

and estimate the B-bin histogram $p_b(\mathcal{R})_{b=1}^{B}$ by normalizing the bin counts over $M(p) \mid p \in \mathcal{R}$. In formal, the local edge entropy is defined as:

$$E(\mathcal{R}) = -\sum_{b=1}^{B} p_b(\mathcal{R}) \log\big(p_b(\mathcal{R}) + \epsilon\big), \tag{2}$$

where ϵ ensures numerical stability. We then use this edge entropy to guide a recursive binary partitioning of the image. Specifically, we assign each region $\mathcal{R}$ a splitting score:

$$S(\mathcal{R}) = |\mathcal{R}| \cdot E(\mathcal{R})^\lambda, \tag{3}$$

where $|\mathcal{R}|$ denotes the region area and λ controls the contribution of edge complexity. Starting from the whole image, we select the region with the highest $S(\mathcal{R})$ and bifurcate it along its longer side (*e.g.*, vertically if width is larger than height). The resulting regions are scored again, and the highest-scoring region is selected for the next bifurcation. This process continues until N regions are obtained. In this way, binary partitioning is progressively biased toward regions with more complex edge information, producing finer partitions in complex regions and coarser partitions in simpler ones (as shown in Figure 1). Given the final partition $\mathcal{P}$, we set the edge-complexity score of each patch $\mathcal{R}_i$ as $d_i = E(\mathcal{R}_i)$. We then select patches whose scores rank in the top ρ fraction to form the high-edge set $\mathcal{D}$, keeping at least $K_{\min}$ patches for sufficient coverage. Next, we create a binary mask $M_\mathcal{D} \in \{0,1\}^{H \times W}$ that marks all pixels in $\mathcal{D}$, and construct the first-stage visual input as:

$$I^e = M_\mathcal{D} \odot \mathcal{Q}_\tau(I) + (1 - M_\mathcal{D}) \odot I, \tag{4}$$

where $\mathcal{Q}_\tau(\cdot)$ denotes the noise-perturbation operator with noise strength τ, and $M_\mathcal{D}$ is the binary mask marking the selected high-edge regions $\mathcal{D}$. The model first forms a fast prediction based on I^e, and then refines the remasked tokens using the original image I to access complete visual context (More implementation details are provided in Algorithm 2).

Table 1: Main benchmark results. F2S-DLLM *(w/o ED)* improves standard decoding in most settings, while EdgeDrop further improves or maintains competitive performance.

Method	Visual Reasoning		General VQA		Micro Avg.
	M^3CoT	V^*	SQA	MMB	
LaViDa					
Standard	34.86	45.55	62.22	65.05	56.08
F2S-DLLM *w/o ED*	36.32 (+1.46)	47.12 (+1.57)	66.14 (+3.92)	69.72 (+4.67)	59.67 (+3.59)
F2S-DLLM	**37.49** (+2.63)	**48.69** (+3.14)	**66.73** (+4.51)	**69.72** (+4.67)	**60.15** (+4.07)
MMaDA					
Standard	**26.62**	32.46	46.16	39.45	37.47
F2S-DLLM *w/o ED*	24.81 (-1.81)	33.51 (+1.05)	46.26 (+0.10)	40.19 (+0.74)	37.40 (-0.07)
F2S-DLLM	26.10 (-0.52)	**36.65** (+4.19)	**47.25** (+1.09)	**41.03** (+1.58)	**38.36** (+0.89)
LLaDA-V					
Standard	47.80	41.36	67.67	74.52	65.25
F2S-DLLM *w/o ED*	**49.40** (+1.60)	39.27 (-2.09)	**68.91** (+1.24)	**75.10** (+0.58)	**66.19** (+0.94)
F2S-DLLM	49.01 (+1.21)	**43.98** (+2.62)	68.77 (+1.10)	74.91 (+0.39)	66.06 (+0.81)

4 Experiment

4.1 Experimental Setup

Benchmarks. We validate our method on four complementary benchmarks: M^3CoT Chen et al. (2024) for multi-step chain-of-thought tasks, SQA Lu et al. (2022) for knowledge-grounded scientific reasoning, MMB Liu et al. (2024b) for broad multimodal perception and reasoning, and V^* Wu & Xie (2024) for high-resolution, fine-grained visual search.

Models and Baselines. We evaluate our framework across three representative multimodal diffusion models: LaViDa (`LaViDa-llada-reason`), MMaDA (`MMaDA-8B-MixCoT`), and LLaDA-V. These architectures span distinct model scales and training recipes, offering a diverse testbed. For each backbone, we benchmark three decoding configurations: standard decoding, F2S-DLLM without EdgeDrop(*w/o ED*) and F2S-DLLM.

Decoding Settings. Across all experiments, we set the max sequence length to 64 tokens. Standard decoding uses 32 unmasking steps (unmasking 2 tokens per step). Under the same total budget of 32 steps, F2S-DLLM allocates T_d=16 steps to fast thinking (4 tokens per step) and T_r=16 steps to slow thinking. After fast thinking, we remask the K=32 tokens with the lowest confidence and regenerate them over the slow thinking steps. All evaluations use greedy decoding to ensure reproducibility.

4.2 Main Results

The results in Table 1 show that F2S-DLLM generally improves standard decoding under the same denoising-step budget by revising low-confidence tokens during slow refinement. Moreover, EdgeDrop shows that visual conditioning during fast thinking can affect subsequent slow thinking performance, although its gains vary across benchmarks and backbones.

Specifically, F2S-DLLM *(w/o ED)* generally improves performance across architectures. In particular, it increases the micro-average accuracy of LaViDa and LLaDA-V by 3.59 and 0.94, respectively. Meanwhile, its performance remains comparable to standard decoding on MMaDA. Results across stages also show how fast and slow thinking complement each other. The fast stage is already effective on SQA and MMB, where many questions can be answered through relatively direct visual reasoning. In contrast, M3CoT and V^* benefit more clearly from subsequent slow thinking, as they require multi-step reasoning or careful inspection of fine-grained visual evidence. The limited improvement on MMaDA without EdgeDrop may come from its shorter and more direct responses. These brief outputs leave less textual context to reconsider uncertain tokens. Overall, these results show that slow thinking is particularly useful for fast predictions rich in global context. These initial predictions provide context to correct mistakes needing further reasoning or visual verification.

EdgeDrop further demonstrates that fast and slow thinking benefit from different visual conditions. On LaViDA and MMaDA, the complete F2S-DLLM improves the micro-average score over its variant without EdgeDrop from 59.67 to 60.15 and from 37.40 to 38.36, respectively. The effect is most clear on V^*, where EdgeDrop improves performance across all three backbones. However, EdgeDrop does not always improve the score, as shown by LLaDA-V (66.06 vs. 66.19). One reason is that different backbones compress visual tokens at different scales. Because EdgeDrop works at the pixel level, heavy spatial pooling can spread out its local effect across larger visual tokens and weaken its impact. These results indicate that visual conditioning is not merely an auxiliary enhancement, but an important component for coordinating fast prediction and slow thinking.

4.3 Ablation Experiment

Effect of Fixed Remask Size. To assess the effect of the slow-thinking revision budget in F2S-DLLM, we ablate the remask size K, which denotes the number of lowest-confidence tokens selected for slow thinking. We vary K from 0 (fast-only prediction) to 32, where K=32 is the default setting evaluated under the same denoising budget as standard decoding.

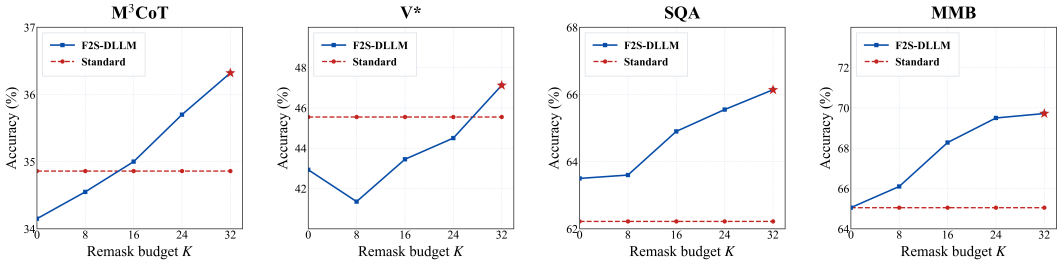


Figure 2: Effect of the slow-thinking budget. Performance generally improves as more uncertain tokens are reconsidered, highlighting the importance of sufficient slow-thinking capacity.

Figure 2 illustrates the impact of the remasking budget K across benchmarks. While overall accuracy scales with K and peaks at K=32, task sensitivity to this budget varies markedly by reasoning complexity. On benchmarks requiring direct visual recognition or factual matching (MMB and SQA), the initial prediction alone is often sufficient, making performance robust to small values of K. In contrast, tasks that require precise visual localization (V^*) or multi-step reasoning (M3CoT) often make mistakes in the initial prediction, where a small K leaves the model without enough room to correct reasoning errors. Consequently, a larger budget such as K=32 provides the model with enough capacity to effectively correct errors across complex tasks.

Effect of Token Selection Strategy. To study how token selection affects slow thinking stage, we compare three strategies for selecting tokens from the complete fast prediction: lowest-confidence, random, and highest-confidence selection. For a fair comparison, all variants start from the same fast prediction and use the same remask budget and slow thinking schedule.

Table 2 compares the three token selection strategies across all benchmarks. Selecting the lowest-confidence tokens consistently achieves the highest final accuracy. Meanwhile, random and high-confidence selection produce limited or inconsistent improvements. These results demonstrate that the performance gains of F2S-DLLM originate from targeted computation on uncertain predictions during the slow-thinking stage, rather than merely relying on additional denoising steps.

Table 2: Comparison of different token selection strategies for slow-thinking stage.

Selection	M^{3}CoT	V*	SQA	MMB
Low conf.	**36.32**	**47.12**	**66.14**	**69.72**
Random	34.73	41.88	63.36	65.63
High conf.	34.60	42.93	63.56	65.60

Effect of Fast-thinking Visual Context. We investigate how the spatial location of visual suppression during fast decoding affects final performance. Specifically, we compare perturbing our default high-edge set $\mathcal{D}$,low-edge set $\mathcal{D}_{\text{low}}$, and a uniform random baseline $\mathcal{D}_{\text{rand}}$ during fast thinking.

Table 3: Effect of the fast-stage perturbation location. Suppressing high-edge regions yields the strongest final performance, showing that edge-aware visual conditioning better supports subsequent slow thinking stage.

Noise Location	M^{3}CoT			V*			SQA			MMB		
	Draft	Final	Δ	Draft	Final	Δ	Draft	Final	Δ	Draft	Final	Δ
Low-edge	33.34	36.10	+2.76	42.93	46.07	+3.14	63.01	65.94	+2.93	64.80	69.48	+4.69
Random-region	33.82	36.50	+2.68	41.36	46.07	+4.71	62.92	66.14	+3.22	64.75	69.69	+5.04
High-edge	33.65	**37.49**	**+3.84**	43.46	**48.69**	**+5.24**	63.36	**66.73**	**+3.37**	64.80	**69.72**	+4.92

Table 3 compares the three fast-stage perturbation locations across all benchmarks. Suppressing high-edge regions consistently produces the highest final accuracy, reaching 48.69 on the vision-centric V^* benchmark. Importantly, the effectiveness of slow thinking stage is not determined solely by the accuracy of the initial fast prediction. On M3CoT, low-edge and high-edge perturbations yield similar fast-stage accuracies (33.34 vs. 33.65), but their final accuracies differ substantially after slow thinking (36.10 vs. 37.49). A similar pattern appears on V^*, where random-region perturbation produces a lower fast-stage accuracy but still fails to match the final performance of high-detail suppression. These results show that EdgeDrop does more than introduce generic visual corruption: selectively suppressing complex visual regions during fast thinking provides a more effective starting point for slow thinking once the original image is restored.

Effect of EdgeDrop Strength. We evaluate the impact of suppression intensity by varying the perturbation strength $\tau \in \{0, 50, 100, 150, 200\}$ during the fast decoding stage.

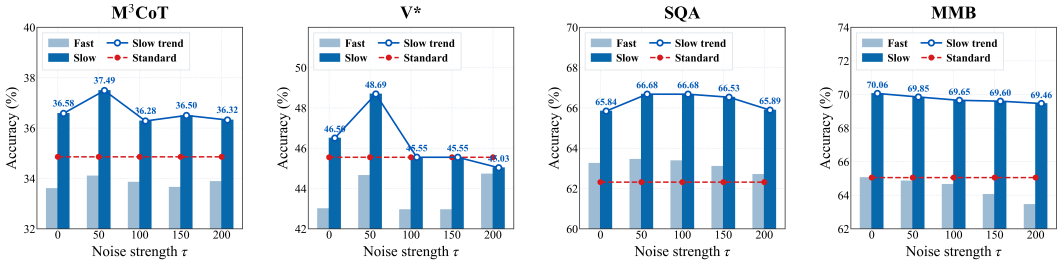


Figure 3: Effect of EdgeDrop strength on fast-stage and final accuracy. Slow thinking generally improves the corresponding fast prediction, while the benefit over standard decoding varies across benchmarks and suppression strengths.

Figure 3 evaluates the sensitivity of F2S-DLLM to the EdgeDrop strength. Slow thinking improves the corresponding fast-stage prediction at every evaluated strength, although the best value of τ remains task-dependent. Final accuracy generally peaks at low-to-moderate strengths ($\tau \in [50, 100]$), while more sensitive benchmarks favor weaker suppression and degrade as τ increases. These results show that stronger EdgeDrop does not monotonically improve performance.

Effect of Slow-thinking Visual Context. We study which visual context best supports the slow-thinking stage. Specifically, we compare three variants using different visual inputs:(1) Original image I; (2) Cropped region I_{crop}: a bounding crop tightly enclosing the high-complexity patches $\mathcal{D}$ identified by EdgeDrop; and (3) Spotlight image I_{spot}:which retains the selected regions while perturbing the remaining background context. Detailed visualizations are provided in Section I.

Table 4: Effect of different visual contexts on slow thinking stage performance.

Context	M^{3}CoT	V*	SQA	MMB
Original (I)	**37.49**	**48.69**	**66.73**	**69.72**
Crop (I_{crop})	37.23	47.12	66.58	68.35
Spotlight (I_{spot})	37.14	46.60	66.19	69.05

As shown in Table 4, restoring the original image during slow thinking achieves the best accuracy, reaching an average of 59.98. In contrast, both Crop and Spotlight lower performance. The largest

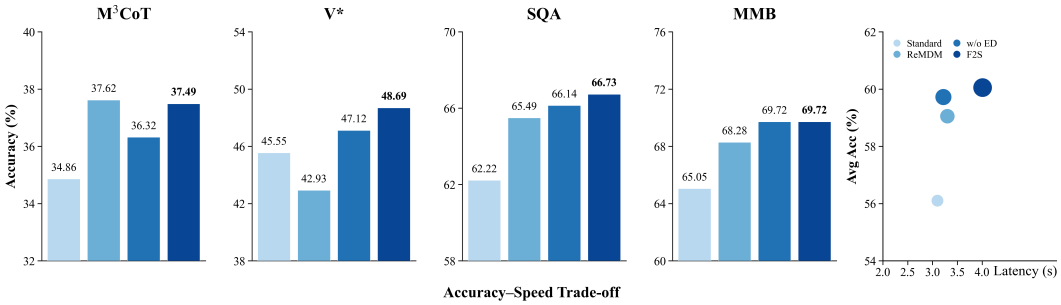


Figure 4: Comparison with in-trajectory remasking and efficiency profiling. F2S-DLLM(w/o ED) outperforms ReMDM on three of four benchmarks and achieves higher average accuracy under the same budget, while EdgeDrop trades additional visual-processing latency for the best overall score.

drop occurs on the V^* benchmark. These results show that slow thinking stage requires complete visual context. While cropped regions contain the key visual features, they lack the background context required to correct uncertain predictions.

Comparison with In-Trajectory Remasking and Efficiency Profiling. To compare our fast-to-slow granularity decoding with correction performed during generation, we evaluate standard decoding, ReMDM, F2S-DLLM*(w/o ED)*, and the complete F2S-DLLM under the same budget of 32 diffusion steps (NFEs). ReMDM progressively remasks tokens within a single denoising trajectory, whereas F2S-DLLM first completes a fast prediction and then reallocates computation to low-confidence tokens during slow thinking stage using the retained global context. We report both benchmark accuracy and wall-clock latency in Figure 4.

Under the same NFE budget, F2S-DLLM*(w/o ED)* achieves a higher micro-average accuracy than ReMDM (59.67 vs. 59.07) while also reducing latency (3.261 s vs. 3.373 s). It outperforms ReMDM on V^*, SQA, and MMB, while ReMDM remains slightly stronger on M3CoT. Compared with standard decoding, fast-to-slow scheduling introduces only 0.008 s of additional latency, indicating that its improvement does not rely on additional denoising iterations. The F2S-DLLM further increases the micro-average accuracy to 59.98 with additional visual preprocessing and separate visual encoding, increasing latency by 0.32 s. Overall, these results show that separating fast thinking from conditioned slow thinking provides a stronger trade-off than interleaving remasking throughout the denoising trajectory.

5 Conclusion

This paper introduces F2S-DLLM, a training-free framework that instantiates fast-to-slow thinking in multimodal DLLMs through flexible denoising granularity. It first forms a complete prediction through aggressive parallel denoising, then remasks low-confidence tokens for further refinement, without additional denoising steps compared with standard decoding. We further introduce EdgeDrop to adapt the visual condition across the two stages, suppressing high-edge-complexity regions during fast thinking and restoring the full image for slow refinement. Experiments across multiple backbones and benchmarks consistently improve over standard decoding, with the slow denoising stage being particularly effective for multi-step reasoning and fine-grained visual understanding.

Limitations and Future Work. We hope this work motivates further exploration of the flexibility of DLLMs, moving beyond parallel decoding for efficiency toward using it to better align multimodal information and augment reasoning. Our current work instantiates fast-to-slow thinking directly at inference time without additional training. While we specify how visual information is conditioned on the fast and slow trajectories, an important future direction is to allow the model to learn this interaction autonomously from data. In particular, different denoising trajectories could provide a useful structure for training multimodal DLLMs, allowing fast and slow reasoning to selectively interact with different visual information. This could enable better alignment between vision and language throughout the denoising process and further improve multimodal reasoning.

AI USE STATEMENT

In this work, we used generative AI tools for language editing, AI-assisted coding, and scholarly navigation, such as identifying related work. We did not use generative AI tools for research ideation or to draft any figures or sections of the paper. We reviewed all AI-assisted work, including checking that every LLM-polished passage preserved our intended meaning and verifying AI-assisted code for correctness before using it in our experiments. We take responsibility for the final content of this work, including any text or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

We are not aware of any ethical concerns or conflicts of interest associated with this work.

REPRODUCIBILITY STATEMENT

We aim to make our experimental setup fully reproducible. The paper and appendix specify the model backbones, decoding schedules, fast and slow denoising granularities, remasking ratios, EdgeDrop construction, and all evaluation settings. All datasets used in our experiments are publicly available, and we use the official benchmark splits and accuracy-based evaluation protocols. We also report the inference budget and implementation details needed to reproduce the reported results. Code and evaluation scripts will be released upon publication.

REFERENCES

Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne Van Den Berg. Structured denoising diffusion models in discrete state-spaces. *Advances in neural information processing systems*, 34:17981–17993, 2021.

Brian K. Chen, Chong Wu, and Kenji Kawaguchi. Speculative correction: Draft-then-refine decoding for diffusion language models. *arXiv preprint arXiv:2608.02625*, 2026a.

Lingjiao Chen, Matei Zaharia, and James Zou. Frugalgpt: How to use large language models while reducing cost and improving performance. *arXiv preprint arXiv:2305.05176*, 2023.

Qiguang Chen, Libo Qin, Jin Zhang, Zhi Chen, Xiao Xu, and Wanxiang Che. M3cot: A novel benchmark for multi-domain multi-step multi-modal chain-of-thought. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8199–8221, 2024.

Yuhang Chen, Xianfeng Wu, Jinhao Duan, Mingfu Liang, Xiaohan Wei, Yunchen Pu, Fei Tian, Chonglin Sun, Parish Aggarwal, Frank Shyu, et al. Bifocal diffusion language models: Asymmetric bidirectional context for parallel generation. *arXiv preprint arXiv:2606.27732*, 2026b.

Jeffrey Cheng et al. Controlling thinking speed in reasoning models. In *Advances in Neural Information Processing Systems*, 2025.

Stephen Chung, Wenyu Du, and Jie Fu. Thinker: Learning to think fast and slow. In *Advances in Neural Information Processing Systems*, 2025.

Sander Dieleman, Laurent Sartran, Arman Roshannai, Nikolay Savinov, Yaroslav Ganin, Pierre H Richemond, Arnaud Doucet, Robin Strudel, Chris Dyer, Conor Durkan, et al. Continuous diffusion for categorical data. *arXiv preprint arXiv:2211.15089*, 2022.

Itai Gat, Tal Remez, Neta Shaul, Felix Kreuk, Ricky T. Q. Chen, Gabriel Synnaeve, Yossi Adi, and Yaron Lipman. Discrete flow matching. *ArXiv*, abs/2407.15595, 2024. URL https://api.semanticscholar.org/CorpusID:271328803.

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*, abs/2506.20639, 2025. URL https://api.semanticscholar.org/CorpusID:280012040.

Ishaan Gulrajani and Tatsunori B Hashimoto. Likelihood-based diffusion language models. *Advances in Neural Information Processing Systems*, 36:16693–16715, 2023.

Yang He, Xiao Ding, Bibo Cai, Yufei Zhang, Kai Xiong, Zhouhao Sun, Bing Qin, and Ting Liu. Self-route: Automatic mode switching via capability estimation for efficient reasoning. *arXiv preprint arXiv:2505.20664*, 2025.

Zhihao He, Tieyuan Chen, Kangyu Wang, Ziran Qin, Yang Shao, Chaofan Gan, Shijie Li, Zuxuan Wu, and Weiyao Lin. Vidlada: Bidirectional diffusion large language models for efficient video understanding. *arXiv preprint arXiv:2601.17868*, 2026.

Jonathan Ho, Ajay Jain, and P. Abbeel. Denoising diffusion probabilistic models. *ArXiv*, abs/2006.11239, 2020. URL https://api.semanticscholar.org/CorpusID:219955663.

Wenyue Hua and Yongfeng Zhang. System 1+ system 2= better world: Neural-symbolic chain of logic reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pp. 601–612, 2022.

Zemin Huang, Yuhang Wang, Zhiyang Chen, and Guo-Jun Qi. Don't settle too early: Self-reflective remasking for diffusion language models. *arXiv preprint arXiv:2509.23653*, 2025.

Sungheon Jeong, Ryozo Masukawa, Jihong Park, Sanggeon Yun, Wenjun Huang, Hanning Chen, Mahdi Imani, and Mohsen Imani. Draft and refine with visual experts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026.

Wonjun Kang, Kevin Galim, Seunghyuk Oh, Minjae Lee, Yuchen Zeng, Shuibai Zhang, Coleman Hooper, Yuezhou Hu, Hyung Il Koo, Nam Ik Cho, and Kangwook Lee. Parallelbench: Understanding the trade-offs of parallel decoding in diffusion llms. *ArXiv*, abs/2510.04767, 2025. URL https://api.semanticscholar.org/CorpusID:281843176.

Samar Khanna, Siddhant Kharbanda, Shufan Li, Harshit Varma, Eric Wang, Sawyer Birnbaum, Ziyang Luo, Yanis Miraoui, Akash Palrecha, Stefano Ermon, Aditya Grover, and Volodymyr Kuleshov. Mercury: Ultra-fast language models based on diffusion. *ArXiv*, abs/2506.17298, 2025. URL https://api.semanticscholar.org/CorpusID:280000358.

Jaeyeon Kim, Seunggeun Kim, Taekyun Lee, David Z Pan, Hyeji Kim, Sham Kakade, and Sitan Chen. Fine-tuning masked diffusion for provable self-correction, 2025a. *URL https://arxiv.org/abs/2510.01384*, 2025.

Shufan Li, Konstantinos Kallidromitis, Hritik Bansal, Akash Gokul, Yusuke Kato, Kazuki Kozuka, Jason Kuen, Zhe Lin, Kai-Wei Chang, and Aditya Grover. Lavida: A large diffusion language model for multimodal understanding. *Advances in Neural Information Processing Systems*, 38: 105101–105134, 2026.

Xiang Li, John Thickstun, Ishaan Gulrajani, Percy S Liang, and Tatsunori B Hashimoto. Diffusion-lm improves controllable text generation. *Advances in neural information processing systems*, 35: 4328–4343, 2022.

Chenyu Lin, Cheng Chi, Jinlin Wu, Sharon Li, and Kaiyang Zhou. Learning to think fast and slow for visual language models. *arXiv preprint arXiv:2511.16670*, 2025.

Anji Liu, Oliver Broadrick, Mathias Niepert, and Guy Van den Broeck. Discrete copula diffusion. *ArXiv*, abs/2410.01949, 2024a. URL https://api.semanticscholar.org/CorpusID:273098329.

Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, et al. Mmbench: Is your multi-modal model an all-around player? In *European conference on computer vision*, pp. 216–233. Springer, 2024b.

Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion modeling by estimating the ratios of the data distribution. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 32819–32848, 2024.

Chenwei Lou, Zewei Sun, Xinnian Liang, Meng Qu, Wei Shen, Wenqi Wang, Yuntao Li, Qingping Yang, and Shuangzhi Wu. AdaCoT: Pareto-optimal adaptive chain-of-thought triggering via reinforcement learning. *arXiv preprint arXiv:2505.11896*, 2025.

Pan Lu, Swaroop Mishra, Tanglin Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. *Advances in neural information processing systems*, 35:2507–2521, 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. *Advances in Neural Information Processing Systems*, 38:50608–50646, 2026.

Isaac Ong, Amjad Almahairi, Vincent Wu, Wei-Lin Chiang, Tianhao Wu, Joseph E Gonzalez, Mohammed Kadous, and Ion Stoica. Routellm: Learning to route llms from preference data. In *International Conference on Learning Representations*, volume 2025, pp. 34433–34448, 2025.

Yidong Ouyang, Panwen Hu, Zhengyan Wan, Zhe Wang, Liyan Xie, Dmitriy Bespalov, Ying Nian Wu, Guang Cheng, Hongyuan Zha, and Qiang Sun. Training-free self-correction for multimodal masked diffusion models. *arXiv preprint arXiv:2602.02927*, 2026.

Fred Zhangzhi Peng, Zachary Bezemek, Sawan Patel, Jarrid Rector-Brooks, Sherwood Yao, Avishek Joey Bose, Alexander Tong, and Pranam Chatterjee. Path planning for masked diffusion model sampling. *arXiv preprint arXiv:2502.03540*, 2025.

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1(2):3, 2022.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.

Subham S Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin T Chiu, Alexander Rush, and Volodymyr Kuleshov. Simple and effective masked diffusion language models. *Advances in Neural Information Processing Systems*, 37:130136–130184, 2024.

Subham Sekhar Sahoo, Aaron Gokaslan, Christopher De Sa, and Volodymyr Kuleshov. Diffusion models with learned adaptive noise. *ArXiv*, abs/2312.13236, 2023. URL `https://api.semanticscholar.org/CorpusID:266374515`.

Tal Schuster, Adam Fisch, Jai Gupta, Mostafa Dehghani, Dara Bahri, Vinh Tran, Yi Tay, and Donald Metzler. Confident adaptive language modeling. *Advances in Neural Information Processing Systems*, 35:17456–17472, 2022.

Jiaxin Shi, Kehang Han, Zhe Wang, Arnaud Doucet, and Michalis K. Titsias. Simplified and generalized masked diffusion for discrete data. *ArXiv*, abs/2406.04329, 2024. URL `https://api.semanticscholar.org/CorpusID:270286107`.

Jascha Narain Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. *ArXiv*, abs/1503.03585, 2015. URL `https://api.semanticscholar.org/CorpusID:14888175`.

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In *Neural Information Processing Systems*, 2019. URL `https://api.semanticscholar.org/CorpusID:196470871`.

Dimitri Von Rütte, Janis Fluri, Yuhui Ding, Antonio Orvieto, Bernhard Schölkopf, and Thomas Hofmann. Generalized interpolating discrete diffusion. *arXiv preprint arXiv:2503.04482*, 2025.

Guanghan Wang, Yair Schiff, Subham Sahoo, and Volodymyr Kuleshov. Remasking discrete diffusion models with inference-time scaling. *Advances in Neural Information Processing Systems*, 38:147282–147339, 2026.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*, 2022.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

Chengyue Wu, Hao Zhang, Shuchen Xue, Shizhe Diao, Yonggan Fu, Zhijian Liu, Pavlo Molchanov, Ping Luo, Song Han, and Enze Xie. Fast-dllm v2: Efficient block-diffusion llm. In *International Conference on Learning Representations*, volume 2026, pp. 128353–128370, 2026a.

Chengyue Wu, Hao Zhang, Shuchen Xue, Zhijian Liu, Shizhe Diao, Ligeng Zhu, Ping Luo, Song Han, and Enze Xie. Fast-dllm: Training-free acceleration of diffusion llm by enabling kv cache and parallel decoding. In *International Conference on Learning Representations*, volume 2026, pp. 57027–57051, 2026b.

Penghao Wu and Saining Xie. V?: Guided visual search as a core mechanism in multimodal llms. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13084–13094, 2024.

Minkai Xu, Tomas Geffner, Karsten Kreis, Weili Nie, Yilun Xu, Jure Leskovec, Stefano Ermon, and Arash Vahdat. Energy-based diffusion language models for text generation. *ArXiv*, abs/2410.21357, 2024. URL https://api.semanticscholar.org/CorpusID:273661614.

Ling Yang, Ye Tian, Bowen Li, Xinchen Zhang, Ke Shen, Yunhai Tong, and Mengdi Wang. Mmada: Multimodal large diffusion language models. *Advances in Neural Information Processing Systems*, 38:138867–138907, 2026.

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*, 2025.

Zebin You, Shen Nie, Xiaolu Zhang, JUN ZHOU, Zhiwu Lu, Ji-Rong Wen, and Chongxuan Li. Llada-v: Large language diffusion models with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10093–10105, 2026.

Runpeng Yu, Xinyin Ma, and Xinchao Wang. Dimple: Discrete diffusion multimodal large language model with parallel decoding. *arXiv preprint arXiv:2505.16990*, 2025.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. Star: Bootstrapping reasoning with reasoning. *Advances in Neural Information Processing Systems*, 35:15476–15488, 2022.

Kevin Zhai, Sabbir Mollah, Zhenyi Wang, and Mubarak Shah. Core: Context-robust remasking for diffusion language models. *arXiv preprint arXiv:2602.04096*, 2026.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, et al. Least-to-most prompting enables complex reasoning in large language models. *arXiv preprint arXiv:2205.10625*, 2022.

Fast-to-Slow Thinking Comes Naturally to Multimodal Diffusion Language Models

— Appendix —

A PRELIMINARIES

Diffusion models define a forward corruption process and learn a reverse denoising process to recover the original data. For discrete text, the process is defined over token sequences from a vocabulary. Let $x^{1:L} \in \mathcal{V}^L$ denote a sequence of length L, where each token $x^l \in \mathcal{V}$ is represented as a one-hot vector over a vocabulary of size V. In masked diffusion models, the forward process randomly replaces tokens with a special mask token according to a noise schedule. The reverse process starts from a highly masked sequence and iteratively predicts the masked positions to recover the complete sequence.

Forward Process. In masked diffusion models, the forward process randomly corrupts a discrete token sequence according to a time-dependent noise schedule. Given a clean sequence $x^{1:L}$, the marginal distribution of the corrupted sequence $z_t^{1:L}$ at time $t \in [0, 1]$ is defined as

$$q(z_t^{1:L} \mid x^{1:L}) = \prod_{l=1}^{L} \mathrm{Cat}\left(z_t^l; \alpha_t x^l + (1 - \alpha_t)m\right),$$

where m denotes the one-hot vector of the special mask token $[M]$, and $\alpha_t \in [0, 1]$ is a monotonically decreasing noise schedule. Thus, each token remains unchanged with probability α_t and is replaced by $[M]$ with probability $1 - \alpha_t$.

Unlike general discrete diffusion processes, which may transition tokens among different vocabulary categories, masked diffusion only corrupts tokens into an absorbing mask state. Once a token is masked, it remains masked in subsequent forward steps.

Reverse Process. The reverse process aims to recover a less corrupted sequence from a more corrupted one. Given two time steps $s < t$, the model learns a denoising transition $p_\theta(z_s^{1:L} \mid z_t^{1:L})$ to approximate the true posterior $q(z_s^{1:L} \mid x^{1:L}, z_t^{1:L})$. For each position l, the posterior has two cases. If z_t^l is not the mask token, its earlier state must remain unchanged. If z_t^l is the mask token, then z_s^l may either remain masked or recover to the original token x^l:

$$q(z_s^l \mid x^l, z_t^l) = \begin{cases} \mathrm{Cat}(z_s^l; z_t^l), & z_t^l \neq m, \\ \mathrm{Cat}\left(z_s^l; \frac{\alpha_s - \alpha_t}{1 - \alpha_t} x^l + \frac{1 - \alpha_s}{1 - \alpha_t} m\right), & z_t^l = m. \end{cases}$$

During generation, the clean token x^l is unknown. Therefore, masked diffusion models usually replace x^l with the model-predicted vocabulary distribution $x_\theta^l(z_t^{1:L})$ to parameterize the reverse denoising transition. The model gradually predicts the masked positions and moves the sequence toward the data distribution.

Training Objective. Masked diffusion models train a parameterized neural network p_θ to approximate the true reverse denoising process. Given a corrupted sequence $z_t^{1:L}$, the model predicts a vocabulary distribution $p_\theta^l(z_t^{1:L})$ for each position l. This predicted distribution is used to replace the unknown clean token in the true posterior. The model is trained by minimizing the KL divergence between the true reverse conditional distribution and the parameterized reverse distribution.

In the continuous-time limit, the MDM training objective can be written as a weighted log-likelihood over masked positions:

$$\mathcal{L}_{\mathrm{MDM}}(\theta) = -\mathrm{E}_{x,t,z_t}\left[\sum_{l=1}^{L} \mathbf{1}[z_t^l = m]\, w(t)\, \log \left\langle x_\theta^l(z_t^{1:L}), x^l \right\rangle\right],$$

where $x \sim q_{\mathrm{data}}$, $t \sim \mathcal{U}(0, 1)$, and $z_t^{1:L} \sim q(z_t^{1:L} \mid x^{1:L})$. The weight $w(t) = \dot{\alpha}_t/(1 - \alpha_t)$ is induced by the noise schedule. Since x^l is a one-hot vector, the inner product gives the probability assigned to the ground-truth token.

Algorithm 2 EdgeDrop Image Construction

Require: Image I, partition budget N, complexity weight λ, quantile threshold ρ, min budget $K_{\min}$, noise strength τ.
Ensure: EdgeDrop image I^e, high-edge binary mask $M_{\mathcal{D}}$.

$G \leftarrow \mathrm{GrayNorm}(I)$ *// Convert to normalized grayscale*
Initialize patch pool $\mathcal{P} \leftarrow \{\Omega\}$ with image canvas Ω
STAGE 1: EDGE-GUIDED RECURSIVE PARTITION
while $|\mathcal{P}| < N$ **do**
  for all $\mathcal{R} \in \mathcal{P}$ **do**
    Compute gradient magnitude $M(p) = \sqrt{(\nabla_x G(p))^2 + (\nabla_y G(p))^2},\ \forall p \in \mathcal{R}$
    Compute edge entropy $E(\mathcal{R}) = -\sum_{b=1}^{B} p_b(\mathcal{R}) \log(p_b(\mathcal{R}) + \epsilon)$
    Compute area-weighted complexity $S(\mathcal{R}) \leftarrow |\mathcal{R}| \cdot E(\mathcal{R})^{\lambda}$
  end for
  $\mathcal{R}^* \leftarrow \arg\max_{\mathcal{R} \in \mathcal{P}} S(\mathcal{R})$ *// Locate patch with highest complexity*
  $\{\mathcal{R}_1, \mathcal{R}_2\} \leftarrow \mathrm{SplitLongerSide}(\mathcal{R}^*)$ *// Bisect $\mathcal{R}^*$ along longer spatial dimension*
  $\mathcal{P} \leftarrow (\mathcal{P} \setminus \{\mathcal{R}^*\}) \cup \{\mathcal{R}_1, \mathcal{R}_2\}$ *// Update partition pool*
end while
STAGE 2: HIGH-EDGE REGION MINING
for all $\mathcal{R}_i \in \mathcal{P}$ **do**
  Assign edge score $d_i \leftarrow E(\mathcal{R}_i)$ *// Measure localized visual sharpness*
end for
Determine cutoff $\eta \leftarrow \mathrm{Quantile}(\{d_i\}_{i=1}^{|\mathcal{P}|}, \rho)$
$\mathcal{D} \leftarrow \{\mathcal{R}_i \in \mathcal{P} \mid d_i > \eta\}$ *// Filter candidate patches exceeding quantile*
if $|\mathcal{D}| < K_{\min}$ **then**
  $\mathcal{D} \leftarrow \mathrm{TopK}_{\mathcal{R}_i \in \mathcal{P}}(d_i, K_{\min})$ *// Enforce minimum coverage budget*
end if
STAGE 3: EDGEDROP BLENDING
Construct binary mask $M_{\mathcal{D}}(p) \leftarrow \mathbb{I}\big(p \in \bigcup_{\mathcal{R}_i \in \mathcal{D}} \mathcal{R}_i\big)$
Generate edge-suppressed image $\widetilde{I}_\tau \leftarrow \mathcal{Q}_\tau(I)$ *// Apply EdgeDrop with strength τ*
Blend fast-stage image $I^e \leftarrow M_{\mathcal{D}} \odot \widetilde{I}_\tau + (1 - M_{\mathcal{D}}) \odot I$ *// Suppress high-edge areas only*
return $I^e, M_{\mathcal{D}}$

Sampling Process. During generation, MDMs start from a fully masked sequence $z_1^{1:L} = m^{1:L}$ and apply the learned reverse process over a discrete time schedule $1 = t_K > t_{K-1} > \cdots > t_0 = 0$. At each step, the model predicts a vocabulary distribution for all masked positions in parallel and samples a less corrupted sequence $z_{t_{k-1}}^{1:L}$ from the current sequence $z_{t_k}^{1:L}$. As the process approaches t_0, more masked positions are filled with concrete tokens, and the final output is obtained when no mask tokens remain.

This sampling procedure can be viewed as iterative unmasking. Unlike autoregressive decoding, it updates multiple positions in parallel. However, once a position is filled with a token, standard MDM sampling typically keeps it fixed in later steps. This property limits the ability of standard MDM sampling to revisit earlier predictions.

B EDGEDROP FOR FAST VISUAL CONDITIONING

Algorithm 2 details the construction of the visual condition used during fast thinking. The algorithm first partitions the image into candidate regions, identifies visually complex regions according to edge entropy, and applies perturbations only to these regions while preserving the remaining visual content. This yields an edge-aware visual condition that suppresses distracting local details during fast denoising while retaining the global structure of the image.

Table 5: Summary of the multimodal benchmarks used for evaluation.

Dataset	Evaluation Subset	#Samples	Task Type
M^{3}CoT	Evaluation split	2,318	Multi-step multimodal reasoning
V*Bench	Evaluation split	191	Fine-grained visual reasoning
ScienceQA	Image-based subset (SQA-IMG)	2,017	Multimodal science QA
MMBench	English development split	4,329	General multimodal QA

C Experimental Setup

Dataset. We evaluate F2S-DLLM on four multimodal reasoning benchmarks covering complementary aspects of visual understanding. All tasks are formulated as multiple-choice question answering. Given an image, a question, and a set of candidate answers, the model generates a textual response, from which we extract the predicted option and compute accuracy against the benchmark annotations. We follow the original evaluation protocols and use the same evaluation split for all decoding methods within each benchmark. The datasets are summarized below.

The counts in Table 5 refer to the subsets used for evaluation in this work, rather than the full sizes of the original benchmark releases. All decoding methods are evaluated on identical test examples within each benchmark. The micro average in the main results is computed over all evaluated instances across the four benchmarks, effectively weighting each benchmark by its sample count.

M^{3}CoT Chen et al. (2024) evaluates multi-domain, multi-step multimodal reasoning across tasks such as science, mathematics, and commonsense reasoning. We use it to assess the effectiveness of slow refinement on problems requiring extended reasoning.

ScienceQA Lu et al. (2022) is a multiple-choice science reasoning benchmark with textual and visual contexts. We evaluate on its image-based subset (SQA-IMG) to test multimodal reasoning that combines scientific knowledge with visual evidence.

MMBench Liu et al. (2024b) covers a broad range of multimodal perception and reasoning abilities, including recognition, attribute and relation understanding, and logical reasoning. We use it to evaluate general multimodal performance across diverse question types.

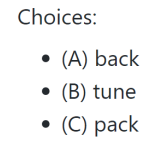
V*Bench Wu & Xie (2024) focuses on high-resolution and fine-grained visual reasoning. We use it to examine whether slow refinement benefits from detailed visual evidence that may be unnecessary during fast thinking.

Together, these benchmarks cover complementary aspects of multimodal reasoning, ranging from general visual understanding to multi-step and fine-grained reasoning. We report accuracy on each benchmark and use the instance-weighted micro average as the aggregate metric.

Decoding Protocol. Unless otherwise specified, the maximum generation length is set to 64 tokens, and all methods employ greedy decoding. The standard decoding baseline unmasks 2 tokens per step over 32 steps. F2S-DLLM operates under the identical total budget of 32 denoising steps, partitioned equally between the fast and slow stages (16 steps each). During the fast stage, the model unmasks 4 tokens per step to form an initial candidate sequence. Subsequently, the K=32 tokens with the lowest prediction confidence are remasked and refined at a rate of 2 tokens per step over the remaining 16 steps. Consequently, F2S-DLLM redistributes the fixed computational budget without incurring extra latency. In isolation, the fast stage requires only 16 steps, achieving a $2\times$ speedup over standard decoding.

EdgeDrop Configuration. EdgeDrop is applied exclusively to the visual input during fast thinking. Specifically, each image is recursively partitioned into N=16 candidate regions guided by area-weighted edge entropy with complexity weight $\lambda = 2.5$. We then select regions falling above the $\rho = 0.5$ entropy quantile, ensuring a minimum retention of $R_{\min} = 4$ regions. Additive perturbation with strength $\tau = 50$ is applied to these selected high-complexity areas, while the remaining image regions remain unaltered. During the subsequent slow refinement stage, the visual input is fully restored to the original pristine image, enabling the model to reconsider uncertain tokens under

Table 6: Qualitative comparison between standard decoding and F2S-DLLM. F2S-DLLM addresses common failure modes in standard decoding, effectively resolving errors in visual grounding, answer selection, and logical consistency.

Question	Image	Standard Decoding	F2S-DLLM
Which property do these three objects have in common? A. Opaque; B. Scratchy; C. Transparent.		To determine the common property among the three objects: 's analyze each one: - Theaser is opaque. - The crayons are opaque. - The rubber ball is opaque. The rubber ball is opaque. Therefore, the correct answer is: C **Answer: C** (*incorrect*)	To determine the common property among the three objects, let's analyze each option: A. Opaque: All three objects are opaque. Theaser is opaque, the rubber crayon is opaque, and the rubber ball is opaque. Therefore, the correct answer is: A **Answer: A** (*correct*)
When does the terrier group competition end? A. 3:25 PM; B. 12:55 PM; C. 11:35 AM; D. 9:40 AM.		To determine when the terrier group competition ends, we need to look at the "END" column associated with the terrier group row. According to the schedule, the terrier group ends at 11:35 A.M. Therefore, the correct answer is: A **Answer: A** (*incorrect*)	To determine when the terrier group competition ends, we need to look at the "END" column for the terrier group. According to the schedule, the terrier group ends at 11:35 A.M. Therefore, the correct answer is: C **Answer: C** (*correct*)
Which word does not rhyme? A. Back; B. Tune; C. None of the options.		To determine which word does not rhyme, let's examine the ending sounds of each word: - Back: /bæk/ - Tune: /aIn/ - Pack: /æk/ The word that does not rhyme is "back". A **Answer: A** (*incorrect*)	To determine which word does not rhyme, let's examine the ending sounds of each word: - back: /æk/ - tune: /In/ - pack: /æk/ Therefore, the word that does not rhyme is: B **Answer: B** (*correct*)
Which image is brighter? A. The first image; B. The second image.		To determine which image is brighter, we need to compare the overall brightness of both images. 1. The first image has a darker background and fewer colors. 2. The second image has a lighter background and more colors. Therefore, the second image is brighter. B **Answer: B** (*incorrect*)	To determine which image is brighter, we need to consider the overall luminance of each. The first image shows the dog in the light, which is clearly visible, while the second image shows the dog in the dark, which is much darker. Therefore, the first image is brighter. A **Answer: A** (*correct*)
Which option correctly describes the relationship in the image? A. The sink contains the cat; B. The cat is beside the microwave; C. The cat is at the edge of the sink; D. The book is beside the cat.		To determine the correct answer, let's analyze the image: 1. The image shows a cat sitting inside a sink. 2. The cat is not completely inside the sink but rather at the edge of the sink. Given this analysis, the correct description is: C **Answer: C** (*incorrect*)	To determine the correct answer, let's analyze the image: 1. The image clearly shows a white cat. 2. The cat is positioned inside the sink. 3. The sink is partially containing the cat. Given these observations, the correct description is: A **Answer: A** (*correct*)

complete visual context. Unless stated otherwise, this parameter configuration remains fixed across all models and benchmarks.

D Qualitative Analysis on Correcting Standard Decoding Errors

Table 6 presents five representative cases where F2S-DLLM successfully yields the correct answer while standard decoding makes errors. These examples span object properties, visual text understanding, phonological reasoning, perceptual comparison, and spatial relationships.

The first three examples show that standard decoding can identify the relevant evidence but still commit to an inconsistent answer token. The remaining examples demonstrate corrections to perceptual and spatial judgments. Together, these cases show that F2S-DLLM can improve both response consistency and the use of visual evidence.

E Instance-Level Effects of Slow Thinking

To investigate how slow thinking alters fast predictions, we conduct an instance-level transition analysis. We categorize each test case into two primary outcomes: a *Fix*, where an incorrect fast prediction is corrected by slow thinking, and a *Regress*, where an initially correct prediction becomes corrupted. We then define *Net Fix* as Fix − Regress to measure overall error-correction efficacy.

Table 7 shows that slow thinking consistently corrects more initial errors than it introduces across all visual conditioning strategies. More importantly, the visual context provided during fast thinking

Table 7: Instance-level prediction transitions from fast to slow thinking under various visual conditions during fast thinking. Each cell reports *Fix / Regress (Net)*, where *Fix* indicates an incorrect fast prediction corrected by slow thinking, *Regress* denotes the reverse, and *Net* is Fix − Regress.

Visual Condition	M^3CoT	MMB	SQA	V^*	Total
Original Image	126 / 75 (+51)	258 / 56 (+202)	75 / 22 (+53)	16 / 8 (+8)	475 / 161 (+314)
Random Noise	141 / 79 (+62)	290 / 72 (**+218**)	89 / 24 (+65)	19 / 10 (+9)	539 / 185 (+354)
Low-Detail Noise	141 / 77 (+64)	291 / 88 (+203)	87 / 28 (+59)	16 / 10 (+6)	535 / 203 (+332)
EdgeDrop	141 / 52 (**+89**)	282 / 69 (+213)	96 / 28 (**+68**)	19 / 9 (**+10**)	538 / 158 (**+380**)

directly shapes subsequent deliberation. EdgeDrop achieves the highest overall Net Fix, primarily by suppressing regressions while preserving a comparable number of successful fixes. This indicates that an appropriate visual condition in the fast phase not only guides the initial prediction, but also establishes a more favorable starting state for subsequent slow thinking.

F Remasked Token Category Analysis

We further analyze the coarse lexical categories of tokens selected for slow thinking under the standard visual condition and EdgeDrop. The categories are defined using lexical proxies, and the reported percentages denote the proportion of remasked tokens assigned to each category.

Table 8: Coarse lexical distribution of tokens remasked for slow thinking under different fast thinking visual conditions.

Token Category	Standard	F2S-DLLM
Object-word	2.3%	2.4%
Relation-word	4.7%	4.7%
Number	1.3%	1.6%
Reasoning connector	4.8%	4.7%
Answer token	2.6%	2.6%
Function word	20.5%	21.9%

Table 8 demonstrates that EdgeDrop does not substantially alter the coarse-grained lexical composition of tokens selected for slow thinking. The proportions of object words, relation words, reasoning connectors, and answer tokens remain virtually unchanged, alongside a modest rise in function words. This indicates that the advantage of EdgeDrop cannot be attributed to simply shifting uncertainty across broad lexical categories; rather, EdgeDrop modifies which specific predictions become uncertain during fast thinking, without altering the overall distribution of token types passed to the slow phase.

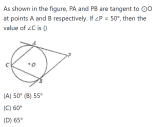
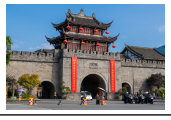
G Qualitative Error-Correction Analysis

Table 9 presents representative cases where slow thinking successfully rectifies errors introduced during the fast phase. Specifically, we showcase instances where standard decoding fails while F2S-DLLM arrives at the correct answer. In each example, the model initially drafts a complete response, after which low-confidence tokens are selectively remasked and refined under the original visual input during slow deliberation.

H Slow thinking gain versus fast thinking Accuracy

Figure 5 demonstrates that the corrective gains of slow thinking are not strictly tied to the initial accuracy achieved during the fast stage. Under varying visual conditions, settings with comparable initial performance produce markedly disparate refinement gains upon deliberation. This reveals that the visual condition during the fast phase dictates not merely immediate prediction quality, but also the refinability of uncertain tokens in the slow phase. EdgeDrop therefore functions beyond simply boosting draft-stage outputs; it establishes a more favorable starting point for downstream selective refinement.

Table 9: Qualitative examples illustrating the fast-to-slow thinking transition. The *Fast Thinking* column shows the initial drafts generated in the fast phase, while the *Slow Thinking* column presents the refined outputs after updating remasked low-confidence tokens. Red and green texts highlight initial errors and their corresponding corrections, respectively.

Question	Image	Fast Thinking	Slow Thinking
Is option D in the image correct for all questions? Choices: A. Not sure; B. False; C. True.		To determine the correct answer, let's analyze the options step by step: Option D states '65°", which is incorrect. Since the value of C is indeed 65°, option D is correct. Therefore, the correct answer is: B	To determine the correct answer, let's analyze the options step by step: Option D states '65°", which is correct. Since the value of C is indeed 65°, option D is correct. Therefore, the correct answer is: C
A tow truck moves a car. What is the direction of the pull? Choices: A. away from the tow truck; B. toward the tow truck.		To determine the direction of the pull exerted by the tow truck, the car is being moved in the opposite direction as the force applied from the truck to the car. This indicates that the force is pulling the car towards the tow truck. Therefore, the correct answer is: A	To determine the direction of the force applied by the tow truck, consider the direction of the motion. The tow truck is towing the car, meaning that the force pulls the car toward the tow truck. Therefore, the correct answer is: B
Which vehicle is closer to the camera, the black vehicle or the silver vehicle? Choices: A. black vehicle; B. silver vehicle.		To determine which vehicle is closer to the camera: the black vehicle is further away, while the silver vehicle is closer. Therefore, the correct answer is: B	To determine which vehicle is closer to the camera: the black vehicle is closer to the foreground, while the silver vehicle is further in the background. Therefore, the correct answer is: A
Which property matches this ceramic plate? Choices: A. scratchy; B. fragile.		The object in the image is a ceramic plate, which is typically fragile and prone to chipping. Given this information, the most appropriate property is fragile. Therefore, the correct answer is: A	The object in the image is a ceramic plate, which can be fragile if not handled carefully. Given this information, the most appropriate property is fragile. Therefore, the correct answer is: B
Is the orange luggage on the left or right side of the purple umbrella? Choices: A. left; B. right.		The purple umbrella is to the right of the orange luggage. Therefore, the orange luggage is on the left side of the purple umbrella. The correct answer is: A	The purple umbrella is to the left of the orange luggage. Therefore, the orange luggage is on the right side of the purple umbrella. The correct answer is: B
Does the kid have curly hair or straight hair? Choices: A. curly hair; B. straight hair.		The child in the image appears to have straight hair. Therefore, the correct answer is: B	The child in the image appears to have curly hair. Therefore, the correct answer is: A
Is the yellow umbrella on the left or right side of the pink umbrella? Choices: A. left; B. right.		The yellow umbrella is positioned to the right of the pink umbrella. Therefore, the correct answer is: B	The yellow umbrella is positioned to the left of the pink umbrella. Therefore, the correct answer is: A

I Visualization of Fast-Stage Visual Conditions

We provide qualitative visualizations of the visual conditions evaluated during the fast stage. For each example, we contrast the original image against Crop, Spotlight, and EdgeDrop constructed from identical candidate patches. These comparisons illustrate how each strategy selectively perturbs local details while preserving varied degrees of global context, offering intuitive rationale for the quantitative gaps observed earlier.

J Effect of Decoding Granularity

We investigate how the decoding granularity in F2S-DLLM influences final generation quality. Let b_f and b_s denote the number of tokens unmasked per step during the fast and slow stages, respectively. For a target sequence length of 64 tokens where 32 tokens are remasked for slow refinement,

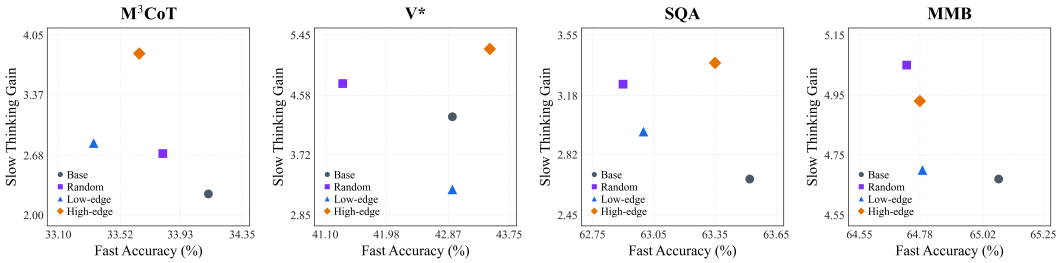


Figure 5: Fast-stage accuracy versus slow-thinking gain across visual conditions. Even with comparable initial accuracy, different visual conditions yield noticeably disparate gains, showing that refinement effectiveness is not governed by draft accuracy alone.



Figure 6: Visual comparison of different fast-stage visual conditions. Crop, Spotlight, and EdgeDrop are constructed from the same selected regions. For EdgeDrop, the noise timestep is set to 400.

the total decoding budget T is defined as:

$$T = \frac{64}{b_f} + \frac{32}{b_s}. \tag{5}$$

We evaluate varying granularity configurations (b_f, b_s) while holding all other inference parameters constant. For a fair comparison, each schedule is evaluated against the standard decoding baseline under the identical step budget T.

As detailed in Table 10, the decoding granularity displays an inverted U-shaped relationship with final accuracy. An overly aggressive schedule ($16/8$, completing in only 8 steps) degrades performance across most tasks, showing that unmasking excessive tokens in parallel deprives the model of essential intermediate context. Moderating the granularity to $8/4$ rapidly recovers accuracy, surpassing the budget-matched standard baseline (+0.92% Micro Avg.) and outperforming even the 32-step standard decoding baseline with half the compute.

The default $4/2$ schedule ($T = 32$) yields optimal performance, leading across three benchmarks and achieving the highest overall average of 60.15% (+4.07% over the baseline). This demonstrates an ideal trade-off between parallel drafting speed and iterative refinement. Further halving the step granularity to $2/1$ ($T = 64$) provides diminishing returns and slight degradation. These findings confirm that while F2S-DLLM relies on adequate bidirectional context, performance does not scale monotonically with hyper-fine decoding steps.

Table 10: Comparison across decoding granularities and step budgets. F2S-DLLM consistently outperforms the baseline under matched budgets when $T \geq 16$, peaking at $T = 32$ (4/2 schedule).

Steps	Method	Schedule	M^3CoT	V^*	SQA	MMB	Micro Avg.
8	Baseline	–	30.80	43.98	63.76	62.42	54.05
	F2S-DLLM	16/8	28.77 (-2.03)	43.98 (+0.00)	63.11 (-0.65)	59.55 (-2.87)	51.97 (-2.08)
16	Baseline	–	33.82	42.41	63.66	64.96	56.02
	F2S-DLLM	8/4	34.08 (+0.26)	**49.74** (+7.33)	65.94 (+2.28)	65.30 (+0.34)	56.94 (+0.92)
32	Baseline	–	34.86	45.55	62.22	65.05	56.08
	F2S-DLLM	4/2	**37.49** (+2.63)	48.69 (+3.14)	**66.73** (+4.51)	**69.72** (+4.67)	**60.15** (+4.07)
64	Baseline	–	33.78	41.88	61.13	64.56	55.23
	F2S-DLLM	2/1	35.81 (+2.03)	49.21 (+7.33)	65.00 (+3.87)	67.50 (+2.94)	58.24 (+3.01)

K LIMITATIONS

While F2S-DLLM shows the promise of inference-time fast-to-slow thinking, it also highlights exciting opportunities for further exploration. We outline two primary aspects as follows:

In this work, EdgeDrop regulates visual complexity through spatial edge entropy. This heuristic is training-free, model-agnostic, and imposes negligible inference overhead. However, spatial edge density serves as a general proxy and does not directly incorporate the specific semantics of the input question. An important next step is to explore task-driven visual conditioning. For example, internal signals such as token-level uncertainty could dynamically guide which regions to suppress or emphasize. Learning this conditioning end-to-end could yield more adaptive visual representations tailored to each thinking speed.

Our empirical findings reveal that visual conditions during fast drafting strongly influence the refinability of tokens in slow deliberation. While our behavioral results confirm this decoupling, the internal dynamics within the diffusion trajectory warrant deeper mechanistic investigation. Future work could examine how representations evolve across diffusion steps, trace information flow via cross-attention layers, and conduct layer-wise causal interventions during the fast-to-slow transition. Such analyses will deepen our understanding of how discrete diffusion models coordinate multimodal evidence across varying thinking velocities.