DRAMA: Dissecting Attention Redundancy for Accelerating Multimodal Diffusion Large Language Models

Yi-Hsin Hung[1], **Fangfu Liu**[2], **Shao-Yuan Lo**[1], **Chu-Song Chen**[1]
[1]National Taiwan University, [2]Tsinghua University
{r14922a09, sylo, chusong}@csie.ntu.edu.tw, liuff19@mails.tsinghua.edu.cn

Abstract

Multimodal diffusion large language models (dLLMs) generate text through iterative masked denoising with bidirectional attention, offering a compelling alternative to autoregressive vision-language modeling. However, each denoising step requires a full forward pass over long multimodal sequences in which visual tokens dominate the sequence length, and a fixed step budget further compounds this quadratic per-step cost. Existing inference acceleration approaches for multimodal dLLMs characterize attention redundancy implicitly, motivating designs from isolated attention visualizations rather than from measurement. We close this gap with DRAMA, a training-free acceleration framework grounded in a systematic dissection of attention redundancy in multimodal dLLMs. By probing attention tensors along spatial and temporal axes, we reveal two persistent redundancy structures: (i) attention mass consistently concentrates on a sparse subset of visual tokens, and (ii) attention distributions rapidly stabilize across denoising steps after an initial formation phase. These findings motivate two complementary mechanisms: Spatial-Redundancy-Aware Visual Token Pruning (SVTP) and Sufficiency-Guided Adaptive Stopping (SGAS). Together, DRAMA achieves an average $18.67\times$ inference speedup across six vision-language benchmarks without training, architectural modification, or attention caching, while maintaining competitive accuracy. Our results establish a strong, measurement-grounded acceleration baseline for multimodal dLLM inference. The code is available at `https://github.com/CindyHung20/DRAMA`.

1 Introduction

Diffusion large language models (dLLMs) [38, 3] generate text by iteratively denoising a fully masked sequence using bidirectional self-attention, enabling parallel token generation and global context reasoning as an alternative to left-to-right autoregressive decoding. Models such as LLaDA [38] and Gemini Diffusion [11] have demonstrated competitive performance with autoregressive counterparts. Recent extensions, including LLaDA-V [52], LaViDa [28], and MMaDA [51], further bring this paradigm to vision-language tasks by conditioning the denoising process on visual inputs.

While effective, the iterative nature of dLLMs incurs substantial computational cost. This issue is further amplified in multimodal settings. Each denoising step requires a full forward pass through all L transformer layers over the concatenated sequence $[\mathbf{v};\, \mathbf{c};\, \mathbf{x}_t]$ of visual tokens $\mathbf{v}$, text condition $\mathbf{c}$, and noisy output $\mathbf{x}_t$, where the visual tokens alone contribute 2–3k tokens per image and dominates the total sequence length N. With a fixed budget of T steps, total cost scales as $\mathcal{O}(T \cdot L \cdot N^2)$, where the visual token count N_v is the dominant term in N. Crucially, the visual prefix is never masked and remains static across all steps, yet its quadratic attention cost is repeatedly incurred at every iteration.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

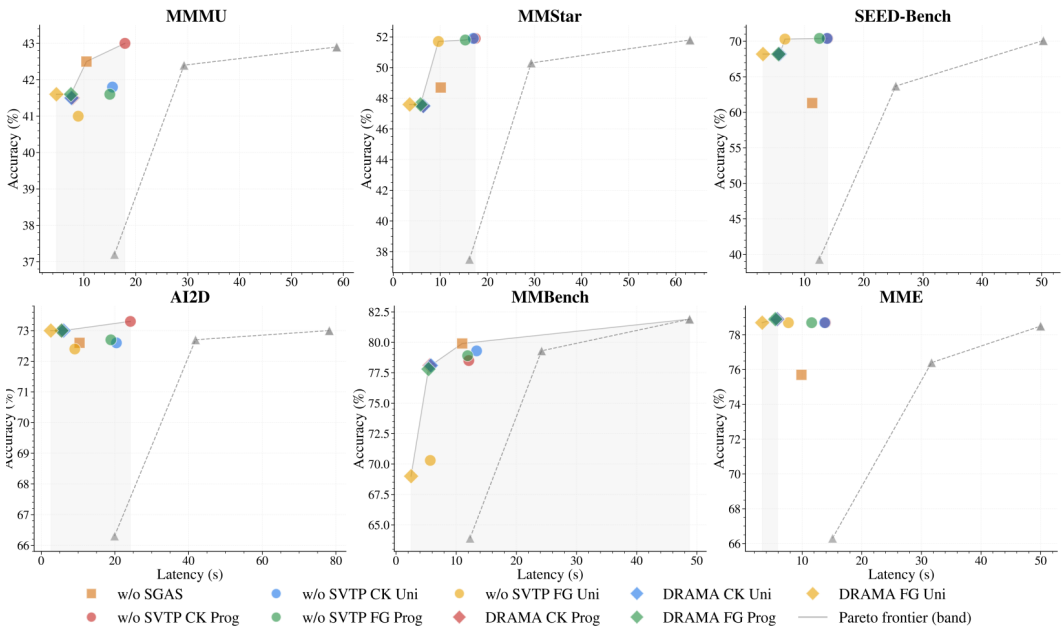


Figure 1: **Accuracy-latency Pareto frontier**. DRAMA variants (diamonds) dominate the Pareto front, achieving higher accuracy at lower latency than all fixed-budget baselines (triangles).

A concurrent line of work has begun extending acceleration techniques to multimodal dLLMs [50, 7, 48]. Their empirical speedups, together with attention visualizations, suggest the presence of exploitable redundancy in this regime. However, these studies characterize such attention redundancy either implicitly through downstream performance gains or isolated examples. Consequently, their designs are rarely accompanied by a systematic measurement of the attention dynamics they exploit.

In this work, we instead ask how redundancy itself can be systematically exploited to derive an analysis-driven, training-free acceleration framework whose budget is set by measurement rather than by tuning. To answer this, we systematically dissect attention redundancy in multimodal dLLMs along two orthogonal axes: a spatial axis examining how attention mass is distributed over visual tokens, and a temporal axis examining how attention evolves across consecutive denoising steps. Quantified across six vision-language benchmarks, our dissection consistently exposes two analysis-grounded redundancy structures, one per axis.

These findings motivate DRAMA (Dissecting attention Redundancy for Accelerating MultimodAl diffusion large language models), a training-free acceleration framework whose two complementary mechanisms each translate one identified redundancy structure into a dedicated strategy. Exploiting the spatial redundancy, Spatial-Redundancy-Aware Visual Token Pruning (SVTP) was designed to retain only dominant visual patches before denoising begins, with no per-step overhead. Sufficiency-Guided Adaptive Stopping (SGAS) exploits the inter-step convergence by terminating denoising once a sufficient decoded sequence has stabilized. By integrating these strategies, DRAMA achieves an average $18.67\times$ inference speedup across six vision-language benchmarks [22, 33, 13, 54, 8, 26] on LLaDA-V without any training, architectural modifications, or attention caching, as shown in Tab. 1 and Fig. 1. Our contributions are summarized as follows:

- We close the gap between observing and exploiting attention redundancy in multimodal dLLMs through the first systematic, two-axis dissection of redundancy structures, providing a quantitative basis for deciding what can be removed and how much.
- We propose DRAMA, a training-free framework that accelerates multimodal dLLMs via Spatial-Redundancy-Aware Visual Token Pruning and Sufficiency-Guided Adaptive Stopping, grounded in the discovered redundancy structures.
- We demonstrate that DRAMA achieves an average $18.67\times$ inference speedup across six benchmarks with competitive performance, providing a strong acceleration baseline for multimodal dLLM inference.

2 Related Work

Diffusion Large Language Models. Diffusion large language models (dLLMs) have emerged as an alternative to autoregressive LLMs, offering parallel decoding capabilities and bidirectional attention mechanisms that enable more effective bidirectional reasoning [38, 42]. Following the pioneering work of LLaDA [38], subsequent efforts including Mercury Coder [24], Gemini Diffusion [11], and Seed Diffusion [42] have demonstrated the commercial viability of this paradigm. Recent works have extended dLLMs to multimodal settings, including MMaDA [51], LLaDA-V [52], and LaViDa [28]. Despite these advances, dLLMs face several fundamental challenges compared to autoregressive models: a persistent perplexity gap [2, 42], inability to naturally generate variable-length sequences [2], and high inference latency due to iterative denoising [42]. While dLLMs theoretically offer higher parallelism than autoregressive models, their iterative nature often results in slower practical inference. Our work addresses this gap by proposing interpretable attention analysis-based acceleration strategies specifically designed for multimodal dLLMs.

Inference Acceleration for Diffusion Language Models. Existing acceleration methods for dLLMs fall into two categories. Since bidirectional attention precludes the standard AR KV cache, caching-based approaches [27, 17, 36, 21, 35, 34, 37, 47, 41, 9] reduce redundant computation by reusing intermediate states across denoising steps: dKV-Cache [35] and dLLM-Cache [34] adaptively cache key-value or feature states based on token-level dynamics, Elastic-Cache [37] triggers layer-aware cache refresh via attention-drift detection, and Fast-dLLM [46] combines block-wise approximate KV caching with confidence-aware parallel decoding. Sampling-optimization approaches alter the generation procedure itself via parallel [6, 18, 45, 12, 23, 5] or speculative [49, 1, 15, 16] decoding, distillation [10, 14], and hybrid modeling methods [46, 44, 19]. A concurrent line of work [50, 7, 48] has begun extending acceleration to the multimodal setting. While their empirical gains signal the presence of exploitable redundancy, this redundancy is characterized only implicitly, inferred from downstream speedups or illustrated on isolated visualizations, leaving open how much of it can actually be removed, and where.

3 Method

3.1 Dissecting Spatial-Axis Attention Redundancy

Multimodal dLLMs based on LLaVA-NeXT [32] partition each image into N_p patches, yielding 2–3k visual tokens that dominate the sequence length and, consequently, the attention cost at every denoising step. Since this visual prefix is static across all T steps, any spatial redundancy in attention is replicated T-fold. We dissect the attention tensors along the spatial dimension at three granularities: individual tokens, image patches, and their cross-step evolution. Each of them reveals a consistent, exploitable structure that generalizes across all evaluated benchmarks. We use MMMU [54] for illustration throughout this section, and the full per-benchmark results are in the Appendix Section E.

Token-Level Attention Mass Concentration. We first examine the finest granularity: the distribution of attention mass over individual visual key tokens, stratified by the modality of query-key interaction. For each query token, we sort its attention weights over the modality-specific key set and compute cumulative coverage at varying retention fractions (Fig. 2(c)). The two cross-modal pairs involving visual keys, namely output→image (**t2v**) and image→image (**v2v**), exhibit strikingly concentrated profiles: the top 1% of visual key tokens already capture $\sim 47\%$ of a query's attention mass, and the top 10% suffice for $\sim 78\%$ coverage. By contrast, text-key interactions (**t2t** and **v2t**) follow a markedly flatter curve, with the top 1% accounting for only $\sim 26\%$ and $\sim 33\%$, respectively. This asymmetry confirms that the extreme sparsity is specific to the visual modality: the model concentrates its attention budget on a small set of visually salient key positions while treating the remaining visual tokens as near-irrelevant context.

Patch-Level Non-Uniformity. If salient tokens were spread uniformly across patches, each patch would receive $\sim 1/N_p$ of the output→image attention, leaving no efficient patch-level pruning strategy. Instead, the distribution is highly skewed within individual samples: the most-attended patch captures $\sim 44\%$ of aggregate attention, which is $1.7\times$ the uniform baseline, and the top two patches suffice for $\sim 68\%$ coverage, while the remaining patches contribute marginally (Fig. 2(a)). Patches are ranked per-sample before averaging, so these fractions reflect genuine per-sample concentration rather than a cross-sample aggregation artefact. The dominant patch's attention share is consistent

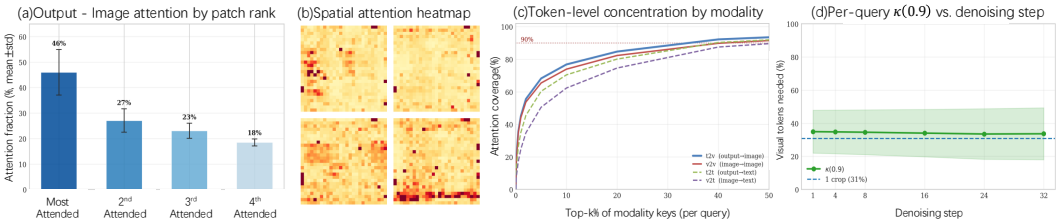


Figure 2: **Spatial redundancy analysis**. **(a)** Per-sample attention share of each image patch ranked by dominance. **(b)** Per-sample attention share distribution of the most-attended patch across the dataset. **(c)** Cumulative attention coverage by top-$k\%$ visual key tokens, stratified by query-key modality pair. **(d)** Minimum visual tokens needed for 90% attention coverage ($\kappa(0.9)$) across denoising steps

across the dataset (see Appendix Section E.1), confirming a stable structural property of the model. This cross-patch skewness implies the effective visual receptive field is far smaller than the full token set, motivating our spatial-redundancy-aware visual token pruning (Section 3.3.1).

Step-Invariant Spatial Redundancy. A critical question for acceleration design is whether the spatial redundancy structure changes as denoising progresses. We track the per-query metric $\kappa_i(0.9)$, the minimum number of visual key tokens needed to capture 90% of an output query's attention mass, averaged across all layers and queries at each denoising step (Fig. 2(d)). Across all benchmarks, $\kappa(0.9)$ remains remarkably stable: it accounts for only $\sim$27% of the total visual token budget at every step, with a relative variation below 5% from the first step to the last. The near-flat trajectory demonstrates that the degree of spatial concentration is step-invariant. Moreover, the identity of the salient positions is also preserved: the inter-step attention divergence measured in Section 3.2 drops to near zero during the plateau phase, and Pinsker's inequality bounds each token's weight change accordingly (formal derivation in Section 3.3.2), confirming that a patch selection made before denoising remains valid across all T steps.

3.2 Dissecting Temporal-Axis Attention Redundancy

The spatial analysis dissects redundancy across tokens, identifying which visual positions can be pruned. Here, we dissect redundancy across denoising steps, examining whether the full attention distribution changes enough between consecutive steps to justify repeating the $\mathcal{O}(L \cdot N^2)$ forward pass. When it does not, successive steps recompute near-identical outputs at full cost, constituting temporal redundancy. We quantify this via the divergence of per-query attention distributions at consecutive steps across the full trajectory.

Inter-Step Distributional Divergence. Following [39], we adopt the Jensen-Shannon divergence (JSD), a symmetric, $[0, \ln 2]$-bounded variant of KL divergence, as our inter-step attention redundancy metric to measure how attention distributions change between consecutive steps. For each layer l, head h, and query position i, the attention weights at denoising step t form a probability distribution over all N key positions:

$$\mathbf{a}_{l,h,i}^{(t)} = \mathrm{Softmax}\left(\frac{\mathbf{q}_{l,h,i}^{(t)} \cdot \mathbf{K}_{l,h}^{(t)\top}}{\sqrt{d_k}}\right) \in \Delta^{N-1}, \tag{1}$$

where Δ^{N-1} denotes the probability simplex. The JSD between consecutive-step attention distributions at the same query position is

$$\mathrm{JSD}\left(\mathbf{a}^{(t)} \,\|\, \mathbf{a}^{(t-1)}\right) = \tfrac{1}{2}\,\mathrm{KL}\left(\mathbf{a}^{(t)} \,\|\, \mathbf{m}\right) + \tfrac{1}{2}\,\mathrm{KL}\left(\mathbf{a}^{(t-1)} \,\|\, \mathbf{m}\right), \tag{2}$$

where $\mathbf{m} = \frac{1}{2}(\mathbf{a}^{(t)} + \mathbf{a}^{(t-1)})$ is the midpoint distribution. Unlike [39], which compares attention across spatial positions within a single step, we compare the same (layer, head, query) triplet across two consecutive steps. We aggregate into a mean per-layer divergence and its normalized redundancy complement:

$$\bar{J}_l(t) = \frac{1}{H \cdot N} \sum_{h=1}^{H} \sum_{i=1}^{N} \mathrm{JSD}\left(\mathbf{a}_{l,h,i}^{(t)} \,\|\, \mathbf{a}_{l,h,i}^{(t-1)}\right), \qquad R_l(t) = 1 - \frac{\bar{J}_l(t)}{\ln 2}. \tag{3}$$

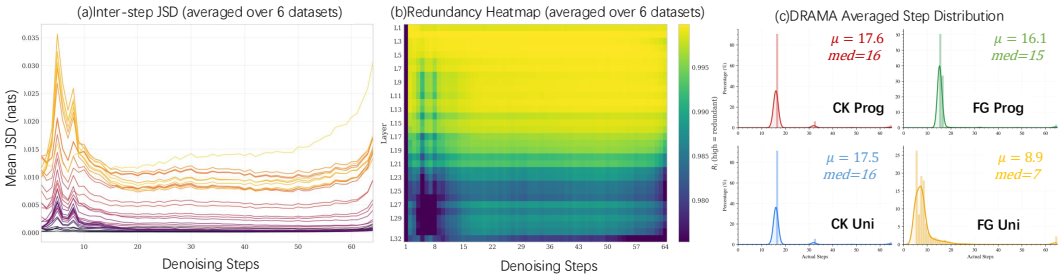


Figure 3: **Temporal redundancy analysis across the denoising trajectory** (averaged over six benchmarks). **(a)** Per-layer inter-step JSD ($\bar{J}_l$) between consecutive denoising steps; peaks highlight steps of greatest attention change. **(b)** Redundancy heatmap $R_l(t)$ across layers and steps; bright regions indicate high redundancy ($R_l \to 1$). **(c)** Distribution of actual stopping steps of our proposed DRAMA, whose concentration aligns with the plateau onset in (a).

$R_l(t) \to 1$ indicates near-identical consecutive distributions.

Phase-Structured Denoising Dynamics. Fig. 3(a) plots $\bar{J}_l(t)$ across the full denoising trajectory for each transformer layer, averaged over all evaluation benchmarks. Rather than declining monotonically, the inter-step divergence reveals a characteristic three-phase temporal structure:

(i) Structure formation ($\bar{J}_l$ peak). The JSD peaks sharply as the sequence transitions from fully masked to carrying initial content. Although only $\sim N_g/T$ tokens are revealed per step, they represent the first semantic signal in an otherwise content-agnostic attention landscape. Naturally resulting in a maximized relative information gain and inducing the largest attention redistribution in the entire trajectory.

(ii) Stable plateau ($\bar{J}_l \to 0$). Once sufficient context has been established, each subsequent unmasking becomes marginal, and the redundancy score rises to $R_l(t) > 0.99$ across all layers (Fig. 3(b), yellow region). Because $\mathbf{O}_l^{(t)} = \mathbf{A}_l^{(t)}\mathbf{V}_l^{(t)}$, near-identical attention maps ($\mathbf{A}_l^{(t)} \approx \mathbf{A}_l^{(t-1)}$) combined with only marginally updated values indicate approximately stationary outputs $\mathbf{O}_l^{(t)} \approx \mathbf{O}_l^{(t-1)}$ during this denoising phase. This plateau spans the majority of the trajectory and constitutes the primary locus of temporal redundancy: the model pays full $\mathcal{O}(L \cdot N^2)$ cost per step for diminishing informational return.

(iii) Terminal refinement ($\bar{J}_l$ secondary rise). A secondary JSD rise emerges as each remaining step must resolve a growing fraction of the remaining ambiguity: with $\sim N_g/T$ tokens unmasked per step but progressively fewer masked positions left, each unmasking carries disproportionate semantic weight, requiring fine-grained disambiguation that redistributes attention in the deeper layers.

3.3 DRAMA

3.3.1 Spatial-Redundancy-Aware Visual Token Pruning.

Given the extreme spatial concentration identified in Section 3.1, a natural strategy is to retain only the most attended visual tokens at each denoising step. However, scoring by attention from the current step requires a forward pass that has already been computed. A cross-step reuse variant that scores tokens using the previous step's attention would circumvent this, but still introduces per-step overhead across all $T \cdot L$ layer passes and faces a cold-start problem at $t = 1$. Crucially, both strategies overlook a key finding from our analysis: the identity of the dominant patches is content-dependent yet step-invariant (Section 3.1), meaning a single, well-chosen selection made before denoising can remain valid across all T steps. This motivates a fundamentally different approach: rather than relying on attention signals that are either unavailable or redundantly recomputed, we score patches by their cross-modal relevance to the input question, performing the selection at the beginning of the denoising process at zero marginal cost.

Specifically, the frozen vision encoder and the dLLM's token embedding layer project visual patches and question tokens into a shared representation space, providing a natural relevance signal without any additional computation. We score each patch i by the similarity between its mean-pooled embedding $\bar{\mathbf{p}}_i$ and the mean-pooled question representation $\bar{\mathbf{q}}$ in this space, and construct the active

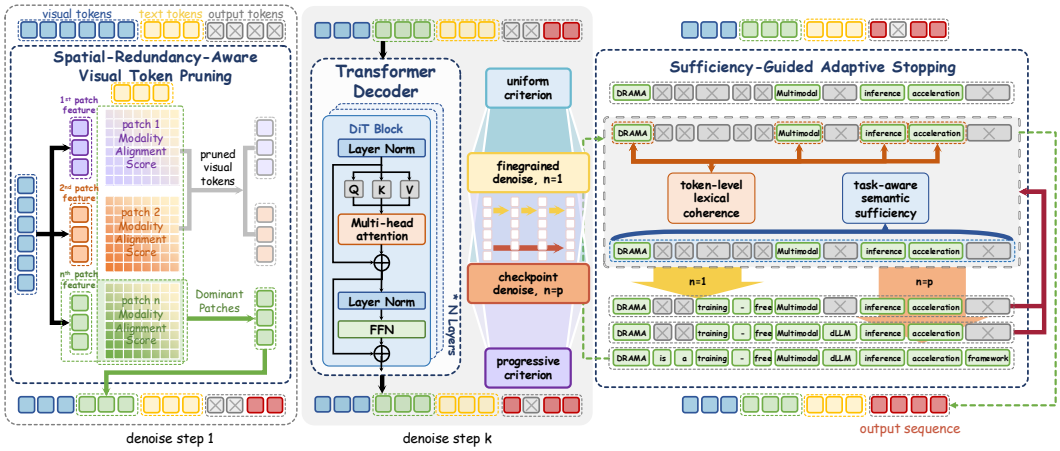


Figure 4: **Overview of DRAMA.** DRAMA applies Spatial-Redundancy-Aware Visual Token Pruning (SVTP) to select a compact patch subset once before denoising. Sufficiency-Guided Adaptive Stopping (SGAS) terminates the denoising trajectory early.

set $\mathcal{P}$ by retaining the global low-resolution overview base patch together with the highest-scoring sub-patches. At every denoising step t and layer l, attention is computed over the filtered key-value set:

$$\hat{\mathbf{A}}_l^{(t)} = \mathrm{Softmax}\left(\frac{\mathbf{Q}\,\mathbf{K}[\mathcal{P}\cup\mathcal{T}]^\top}{\sqrt{d_k}}\right)\mathbf{V}[\mathcal{P}\cup\mathcal{T}], \quad \text{where} \quad \mathcal{P}=\{0\}\cup\left\{i \mid s_i \geq s_{(|\mathcal{P}|)}\right\}, \tag{4}$$

with $s_i = \mathrm{sim}(\bar{\mathbf{p}}_i, \bar{\mathbf{q}})$ and $\mathcal{T} = \mathcal{I}_c \cup \mathcal{I}_g$ denoting all text token indices.

This reduces the effective key-value length from N to $|\mathcal{P}|\cdot M + |\mathcal{T}|$; since visual tokens constitute over 90% of the sequence ($N_v \gg N_c + N_g$), even moderate pruning yields a substantial per-step speedup that compounds across all T steps. The entire scoring procedure involves only N_p inner products in $\mathbb{R}^D$ and is performed once before the first denoising step, introducing no per-step overhead or attention caching.

3.3.2 Sufficiency-Guided Adaptive Stopping.

The plateau established in Section 3.2 shows that, beyond the structure-formation phase, consecutive denoising steps produce near-identical attention distributions across all layers. A Pinsker-type bound on the resulting layer-output drift (full derivation in App. D) then propagates this stationarity through the residual stream and the output projection to the decoded token sequence, allowing us to elevate the decoded text to a token-space sufficiency surrogate. This surrogate supplants any runtime monitoring of $\bar{J}_l(t)$, which would otherwise re-introduce the per-layer attention caching we seek to eliminate, with a stopping rule defined in decoded sequence space.

Stop predicate. We frame the stopping rule as the first-hit time of a conjunction of two failure-mode-orthogonal predicates,

$$\begin{aligned} t^\star = \ & \min\left\{\, t \geq T_{\min} : t \in \mathcal{E},\ \mathrm{Suff}(\mathbf{x}_t) \wedge \mathrm{Coh}(\mathbf{x}_t) \,\right\}, \\ \text{where} \ \ \mathrm{Suff}(\mathbf{x}_t) := \ & \mathbb{1}\left[\textsc{Parse}(\mathbf{x}_t) \in \mathcal{V}_{task}\right],\ \ \mathrm{Coh}(\mathbf{x}_t) := \mathbb{1}\left[\boldsymbol{r}(\mathbf{x}_t) \in \mathcal{R}_{\boldsymbol{\theta}(t)}\right], \end{aligned} \tag{5}$$

with $\boldsymbol{r}(\mathbf{x}_t) = (r^{\mathrm{div}}, r^{\mathrm{rep}}, r^{\mathrm{con}})$ the diversity, peak-frequency, and consecutive-repetition statistics of $\mathbf{x}_t$, $\mathcal{R}_{\boldsymbol{\theta}(t)}$ their admissible region under the threshold schedule $\boldsymbol{\theta}(t)$, $\mathcal{V}_{task}$ the set of structurally valid task answers, and $\mathcal{E}$ the granularity-controlled evaluation set; the procedure returns $\mathbf{x}_{t^\star}$, defaulting to the full-budget output $\mathbf{x}_T$ when the predicate never triggers, so the worst-case behaviour gracefully recovers the unaccelerated baseline. The two predicates discharge complementary failure modes: Suff alone admits parseable answers embedded in degenerate near-mask sequences, while Coh alone admits fluent but task-irrelevant continuations, so their conjunction rules out both pathologies by construction, requiring neither retraining nor an auxiliary verifier. The concrete forms of $\textsc{Parse}$, $\boldsymbol{r}$, and the per-variant schedules $\boldsymbol{\theta}(t)$ are detailed in App. B.3 due to space.

Design space. Eq. (5) exposes two orthogonal design axes whose Cartesian product instantiates the 2×2 design space ablated in Section 4.3. Evaluation granularity controls when the predicate is queried along the trajectory: a *checkpoint* schedule evaluates only on a sparse set $\mathcal{C}$, amortising decoding overhead over a coarse partition, whereas a *fine-grained* schedule evaluates at every step beyond $T_{\min}$ to admit the earliest viable stop. Threshold scheduling controls how strictly the predicates are evaluated: a *progressive* schedule tightens $\boldsymbol{\theta}(t)$ early and relaxes it later, encoding the structure-formation prior surfaced in Section 3.2, whereas a *uniform* schedule applies fixed thresholds throughout, encoding the absence of such a prior. The two axes are independent by construction, with the first governing temporal density and the second semantic strictness, so the four resulting variants form a 2×2 instantiation whose two axes correspond to the two redundancy structures measured in Section 3.2 and 3.1. The full procedure (Algorithm 1, deferred to App. B.3) and the empirical stopping distribution under each variant (Fig. 5) concentrate near the plateau onset predicted in Section 3.2, closing the loop between dissection and design.

4 Experiments

4.1 Experimental Setup

Datasets and Baselines. We evaluate on six vision-language benchmarks spanning diverse task formats: AI2D [22], MMBench [33], MME [13], MMMU [54], MM-Star [8], and SEED-Bench [26]. For SEED-Bench, we evaluate on a randomly sampled subset of 2,000 samples on LLaDA-V and 1,000 samples on LaViDa, following LLaDA-V's own protocol; all other benchmarks are evaluated in full. We compare against LLaDA-V [52] with fixed denoising budgets of $T \in \{16, 32, 64\}$ steps, representing the natural baseline of uniform step reduction. The 64-step configuration serves as the unaccelerated reference throughout. We further provide comparisons of LaViDa on five of the evaluation benchmarks, in Appendix Table 6.

Implementation Details. We adopt LLaDA-V [52], a 7B-parameter multimodal dLLM built on $L=32$ transformer layers with $H=32$ attention heads and a frozen SigLIP2 [43] vision encoder, as our evaluation backbone. Inference is performed in bfloat16 on a single NVIDIA RTX PRO 6000 GPU (96 GB) with a batch size of 1. Generation uses a single block of $N_g=128$ tokens with deterministic decoding (temperature 0.0) and low-confidence remasking [38]. For SGAS, the maximum denoising budget is $T = 64$ with $T_{\min} = 1$; checkpoint mode evaluates at steps $\mathcal{C} = \{16, 32, 48, 64\}$, and fine-grained mode evaluates at every step beyond $T_{\min}$. The full study consumes 747.99 GPU-hours in total (including baselines, ablations, and preliminary runs), of which 82.18 GPU-hours are spent on the four reported DRAMA variants.

4.2 Main Results

Accuracy-Speed Trade-Off. We report accuracy, wall-clock latency, throughput, and speedup in Table 1 across all six benchmarks for three fixed-budget baselines, four full DRAMA variants, and five ablation configurations (detailed in Section 4.3). DRAMA CK Prog and CK Uni achieve $\sim9.5\times$ average speedup at 64.5% accuracy (-1.9 pp vs. the 64-step reference), simultaneously surpassing Baseline 32 in both speed ($\sim5\times$ faster) and accuracy ($+0.4$ pp). FG Prog offers a middle ground at $10.00\times$ and 64.5% (-1.9 pp), comparable to CK variants but with finer-grained stopping control. At the aggressive end, FG Uni pushes the speedup to $18.67\times$ at 63.0% (-3.4 pp). All DRAMA variants substantially outperform Baseline 16 (51.8%, $3.80\times$) in both quality and speed, confirming that analysis-guided acceleration yields strictly better Pareto trade-offs than uniform step reduction. Notably, acceleration does not always come at an accuracy cost: on MME, every DRAMA variant surpasses the unaccelerated baseline (78.9% vs. 78.5%), and on AI2D, all variants match it, suggesting that pruning redundant visual tokens can alleviate attention dilution when the model's full visual budget exceeds the task's information requirement. The accuracy-latency Pareto frontier (Fig. 1) further illustrates this: DRAMA variants consistently dominate the front across all benchmarks, with no fixed-budget baseline achieving a comparable speed-accuracy combination.

Throughput. In absolute throughput, the baseline's 2.6 tokens/s is impractical for interactive use. DRAMA CK variants raise this to 22.3 tokens/s, while FG Uni reaches 51.9 tokens/s on average and peaks at 60.2 tokens/s on AI2D, bringing multimodal dLLM inference into a regime comparable to autoregressive models of similar scale. Even the most conservative variant, CK Prog, delivers a

Table 1: Results on six benchmarks. Metrics: Acc (%, ↑), Latency (s/sample, ↓), TPS (tokens/s, ↑), Speedup vs. Baseline 64 steps (↑). w/o SGAS uses 32 denoising steps; its Speedup is normalized by $\div 2$ to discount the inherent gain from halving the step count. Best 2nd best (Baselines and DRAMA); Best 2nd best (Ablation). †Evaluated on a subset of the full benchmark.

Metric	Baseline			Ablation					DRAMA (Ours)			
	64 steps	32 steps	16 steps	w/o SGAS	w/o SVTP FG Prog	w/o SVTP FG Uni	w/o SVTP CK Prog	w/o SVTP CK Uni	FG Prog	FG Uni	CK Prog	CK Uni
AI2D												
Acc (%)	**73.0**	72.7	66.3	72.6	72.7	72.4	**73.3**	72.6	73.0	**73.0**	73.0	73.0
Latency (s)	78.2	41.9	19.9	10.4	18.9	**9.1**	24.2	20.4	5.5	**2.6**	5.7	6.1
TPS (tok/s)	2.4	4.2	9.2	12.2	9.8	**23.4**	7.8	9.0	23.9	**60.2**	23.5	21.7
Speedup	1.00×	1.87×	3.93×	3.76×	4.14×	**8.59×**	3.23×	3.83×	14.22×	**30.08×**	13.72×	12.82×
MMBench												
Acc (%)	**81.9**	79.3	63.9	79.9	78.9†	70.3	78.5	79.3	77.8	69.0	78.1	78.1
Latency (s)	48.8	24.2	12.3	11.0	11.9	**5.7**	12.1	13.4	5.4	**2.5**	5.6	5.8
TPS (tok/s)	2.8	5.7	11.3	11.6	11.7	**25.3**	11.5	10.4	23.6	**53.7**	22.9	21.8
Speedup	1.00×	2.02×	3.97×	2.22×	4.10×	**8.56×**	4.03×	3.64×	9.04×	**19.52×**	8.71×	8.41×
MME												
Acc (%)	78.5	76.4	66.3	75.7	**78.7**	78.7	**78.7**	**78.7**	**78.9**	78.7	**78.9**	**78.9**
Latency (s)	50.0	31.7	15.1	9.9	11.6	**7.7**	13.9	13.8	5.5	**3.3**	5.9	5.8
TPS (tok/s)	2.8	4.3	9.3	12.8	12.1	**21.4**	10.6	10.7	23.9	**46.0**	22.9	23.2
Speedup	1.00×	1.58×	3.31×	2.53×	4.31×	**6.49×**	3.60×	3.62×	9.09×	**15.15×**	8.47×	8.62×
MMMU												
Acc (%)	**42.9**	42.4	37.2	42.5	41.6	41.0	**43.0**	41.8	41.6	41.6	41.5	41.5
Latency (s)	58.7	29.3	15.9	10.5	15.0	**8.9**	17.9	15.5	7.5	**4.7**	7.8	7.6
TPS (tok/s)	2.5	5.1	9.6	12.1	10.3	**21.0**	9.0	10.0	20.8	**46.3**	19.9	20.4
Speedup	1.00×	2.00×	3.69×	2.80×	3.91×	**6.60×**	3.28×	3.79×	7.83×	**12.49×**	7.53×	7.72×
MM-Star												
Acc (%)	**51.8**	50.3	37.5	48.7	51.8	51.7	**51.9**	**51.9**	47.6	47.6	47.5	47.5
Latency (s)	63.0	29.3	16.2	10.1	15.3	**9.6**	17.4	17.0	5.8	**3.5**	6.3	6.4
TPS (tok/s)	2.5	5.3	9.9	12.6	11.3	**25.2**	10.2	10.4	23.5	**49.7**	22.2	21.8
Speedup	1.00×	2.15×	3.89×	3.12×	4.12×	**6.56×**	3.62×	3.71×	10.86×	**18.00×**	10.00×	9.84×
SEED-Bench												
Acc (%)	**70.1**†	63.7†	39.3†	61.3†	**70.4**†	70.3†	**70.4**†	**70.4**†	68.2†	68.2†	68.2†	68.2†
Latency (s)	50.3	25.4	12.5	11.3	12.5	**6.7**	13.9	13.8	5.6	**3.0**	5.7	5.8
TPS (tok/s)	2.7	5.3	10.8	11.3	11.2	**26.6**	10.2	10.3	23.8	**55.5**	23.3	23.1
Speedup	1.00×	1.98×	4.02×	2.23×	4.02×	**7.51×**	3.62×	3.64×	8.98×	**16.77×**	8.82×	8.67×
Average												
Acc (%)	**66.4**	64.1	51.8	63.5	65.7	64.1	**66.0**	65.8	64.5	63.0	64.5	64.5
Latency (s)	58.2	30.3	15.3	10.5	14.2	**8.0**	16.6	15.7	5.9	**3.3**	6.2	6.3
TPS (tok/s)	2.6	5.0	10.0	12.1	11.1	**23.8**	9.9	10.1	23.3	**51.9**	22.5	22.0
Speedup	1.00×	1.93×	3.80×	2.78×	4.10×	**7.39×**	3.56×	3.71×	10.00×	**18.67×**	9.54×	9.35×

$\sim 9\times$ throughput increase, confirming that the redundancy structures identified in Section 3.1 and Section 3.2 translate into substantial real-world speed gains.

Per-Benchmark Variation. Acceleration gains vary with task characteristics. On AI2D, FG Uni achieves $30.08\times$ speedup while matching the 64-step baseline accuracy (73.0%), as diagram-centric queries concentrate attention on a few salient patches. On MME, CK variants simultaneously improve accuracy (78.9% vs. 78.5%) and deliver up to $8.47\times$ speedup, reflecting rapid attention convergence for perceptual tasks. On MMMU, CK variants retain 41.5% (-1.4 pp) at $\sim 7.5\times$ speedup; on MM-Star, 47.5% (-4.3 pp) at $\sim 10\times$; MMBench follows a similar pattern at 78.1% (-3.8 pp, $\sim 8.5\times$). We further disentangle the contribution of each component in Section 4.3, where SGAS alone matches accuracy on most of the benchmarks, confirming that it serves as a stable backbone that absorbs the accuracy cost of aggressive visual token pruning.

4.3 Ablation Study

We further ablate (i) the individual contributions of SVTP and SGAS as orthogonal cost-axis reductions; and (ii) the interaction between SGAS's evaluation-granularity and threshold-scheduling axes that defines the 2×2 design space introduced in Section 3.3.2. We defer (iii) generalisation to a second backbone (LaViDa); and (iv) head-to-head comparisons with two concurrent multimodal dLLM acceleration methods to Appendix C.

Effect of SVTP. At a fixed budget of $T = 32$ steps, SVTP alone yields $2.78\times$ normalised speedup at 63.5% accuracy, reducing per-sample time from 30.3 s to 10.5 s (see the ablation columns of

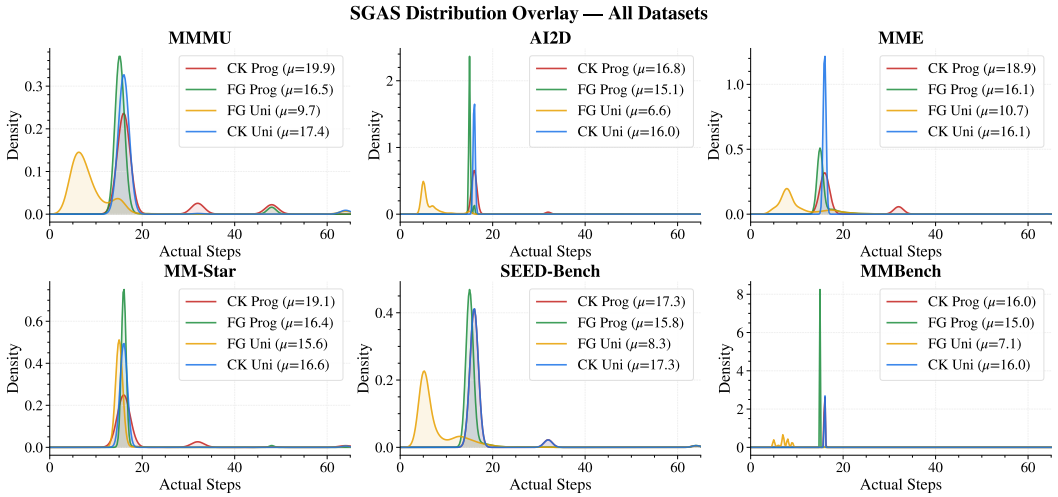


Figure 5: **Distribution of actual denoising steps** under SGAS across six benchmarks. Each subplot shows the histogram and KDE for one SGAS variant. FG Uni concentrates most samples at very early steps ($\mu \approx 8$–12), while CK variants cluster around the first eligible checkpoint (step 16). Progressive scheduling shifts the distribution rightward, reflecting its stricter early-step requirements.

Table 1). The gain comes entirely from shortening the key-value sequence, consistent with the spatial concentration in Section 3.1, and accuracy remains within -2.4 to $+0.6$ pp of the 32 steps baseline on all benchmarks. When combined with SGAS, SVTP contributes a consistent $\sim 2.5\times$ multiplicative factor regardless of the stopping variant. This confirms that spatial pruning and temporal stopping target orthogonal cost axes (N and T in $\mathcal{O}(T \cdot L \cdot N^2)$) and compose independently.

Effect of SGAS. Without SVTP, all four SGAS configurations achieve 3.56–$7.39\times$ speedup while staying within 2.3 pp of the 64 steps baseline on average (see the ablation columns of Table 1). The most conservative variant, CK Prog, matches or exceeds the 64-step baseline's accuracy on five of six benchmarks at $3.56\times$ speedup, indicating that the fixed budget over-denoises on most tasks. At the aggressive end, FG Uni reaches $7.39\times$ while matching Baseline 32 accuracy. This confirms that the temporal redundancy identified in Section 3.2 translates directly into our SGAS component without dedicated retraining.

Stopping Strategy Interaction. We discover that evaluation granularity (FG vs. CK) and threshold schedule (Prog vs. Uni) interact strongly. Under uniform thresholds, FG nearly doubles CK's speedup, aligning with per-step evaluation's more exit opportunities. Under progressive thresholds, this gap largely vanishes: the stricter early-step requirements absorb FG's frequent evaluations, keeping accuracy close to the baseline. Symmetrically, switching from Prog to Uni barely affects CK ($3.56\times vs. 3.71\times$) but dramatically shifts FG ($4.1\times vs. 7.4\times$). We visualize DRAMA's stopping-step distributions (Fig. 5, per-benchmark breakdowns are shown in Supplementary Materials), which corroborate our JSD redundancy analysis: FG Uni concentrates exits at early steps, CK variants cluster around the first checkpoint, and progressive scheduling shifts the distribution rightward. This shows that progressive scheduling acts as a safeguard during the structure-formation phase (Section 3.2), while still allowing the system to exploit plateau-phase redundancy for substantial acceleration.

5 Conclusion

We present DRAMA, a training-free inference acceleration framework for multimodal dLLMs that closes the gap between merely observing attention redundancy and systematically exploiting it. Driven by a measured dissection of where redundancy lies, DRAMA delivers an average $18.67\times$ inference speedup on LLaDA-V across six vision-language benchmarks with competitive accuracy, attained without any training, architectural modification, or attention caching. Beyond its empirical gains, DRAMA reframes acceleration in this emerging paradigm as a problem of identifying and exploiting structural redundancy, offering an analysis-driven perspective which informs broader efforts toward measurement-grounded acceleration of multimodal diffusion large language models.

Acknowledgments and Disclosure of Funding

This work was supported by the National Science and Technology Council (NSTC) of Taiwan under Grant 115-2634-F-002-012 and by the NTU AI Center of Research Excellence (AI-CoRE), with additional computational support provided through H100 GPUs donated by Wistron Corporation. Shao-Yuan Lo is honored to be supported by the Yushan Young Fellow Program by the Ministry of Education (MOE) of Taiwan.

References

[1] Sudhanshu Agrawal, Risheek Garrepalli, Raghavv Goel, Mingu Lee, Christopher Lott, and Fatih Porikli. Spiffy: Multiplying diffusion llm acceleration via lossless speculative decoding. *arXiv preprint arXiv:2509.18085*, 2025.

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

[3] 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.

[4] Zechen Bai, Pichao Wang, Tianjun Xiao, Tong He, Zongbo Han, Zheng Zhang, and Mike Zheng Shou. Hallucination of multimodal large language models: A survey. *arXiv preprint arXiv:2404.18930*, 2024.

[5] Parikshit Bansal and Sujay Sanghavi. Enabling approximate joint sampling in diffusion lms. *arXiv preprint arXiv:2509.22738*, 2025.

[6] Wenrui Bao, Zhiben Chen, Dan Xu, and Yuzhang Shang. Learning to parallel: Accelerating diffusion large language models via learnable parallel decoding. *arXiv preprint arXiv:2509.25188*, 2025.

[7] Shuochen Chang, Xiaofeng Zhang, Qingyang Liu, and Li Niu. D3tom: Decider-guided dynamic token merging for accelerating diffusion mllms. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 19961–19969, 2026.

[8] Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Jiaqi Wang, Yu Qiao, Dahua Lin, et al. Are we on the right way for evaluating large vision-language models? *Advances in Neural Information Processing Systems*, 37:27056–27087, 2024.

[9] Xinhua Chen, Sitao Huang, Cong Guo, Chiyue Wei, Yintao He, Jianyi Zhang, Hai Li, Yiran Chen, et al. Dpad: Efficient diffusion language models with suffix dropout. *arXiv preprint arXiv:2508.14148*, 2025.

[10] Zigeng Chen, Gongfan Fang, Xinyin Ma, Ruonan Yu, and Xinchao Wang. dparallel: Learnable parallel decoding for dllms. *arXiv preprint arXiv:2509.26488*, 2025.

[11] Google DeepMind. Gemini diffusion. *https://blog.google/technology/google-deepmind/gemini-diffusion/. https://blog.google/technology/google-deepmind/gemini-diffusion/*, 2025. Accessed: 2024-07-24.

[12] Yihong Dong, Zhaoyu Ma, Xue Jiang, Zhiyuan Fan, Jiaru Qian, Yongmin Li, Jianha Xiao, Zhi Jin, Rongyu Cao, Binhua Li, et al. Saber: An efficient sampling with adaptive acceleration and backtracking enhanced remasking for diffusion language model. *arXiv preprint arXiv:2510.18165*, 2025.

[13] Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, et al. Mme: A comprehensive evaluation benchmark for multimodal large language models. *arXiv preprint arXiv:2306.13394*, 2023.

[14] Feiyang Fu, Tongxian Guo, and Zhaoqiang Liu. Learnable sampler distillation for discrete diffusion models. *arXiv preprint arXiv:2509.19962*, 2025.

[15] Yifeng Gao, Ziang Ji, Yuxuan Wang, Biqing Qi, Hanlin Xu, and Linfeng Zhang. Self speculative decoding for diffusion large language models. *arXiv preprint arXiv:2510.04147*, 2025.

[16] Feng Hong, Geng Yu, Yushi Ye, Haicheng Huang, Huangjie Zheng, Ya Zhang, Yanfeng Wang, and Jiangchao Yao. Wide-in, narrow-out: Revokable decoding for efficient and effective dllms. *arXiv preprint arXiv:2507.18578*, 2025.

[17] Zhanqiu Hu, Jian Meng, Yash Akhauri, Mohamed S Abdelfattah, Jae-sun Seo, Zhiru Zhang, and Udit Gupta. Flashdlm: Accelerating diffusion language model inference via efficient kv caching and guided diffusion. *arXiv preprint arXiv:2505.21467*, 2025.

[18] Pengcheng Huang, Shuhao Liu, Zhenghao Liu, Yukun Yan, Shuo Wang, Zulong Chen, and Tong Xiao. Pc-sampler: Position-aware calibration of decoding bias in masked diffusion models. *arXiv preprint arXiv:2508.13021*, 2025.

[19] Daniel Israel, Tian Jin, Ellie Cheng, Guy Van den Broeck, Aditya Grover, Suvinay Subramanian, and Michael Carbin. Planned diffusion. *arXiv preprint arXiv:2510.18087*, 2025.

[20] Huiqiang Jiang, Yucheng Li, Chengruidong Zhang, Qianhui Wu, Xufang Luo, Surin Ahn, Zhenhua Han, Amir H Abdi, Dongsheng Li, Chin-Yew Lin, et al. Minference 1.0: Accelerating pre-filling for long-context llms via dynamic sparse attention. *Advances in Neural Information Processing Systems*, 37:52481–52515, 2024.

[21] Yuchu Jiang, Yue Cai, Xiangzhong Luo, Jiale Fu, Jiarui Wang, Chonghan Liu, and Xu Yang. d2 cache: Accelerating diffusion-based llms via dual adaptive caching. *arXiv preprint arXiv:2509.23094*, 2025.

[22] Aniruddha Kembhavi, Mike Salvato, Eric Kolve, Minjoon Seo, Hannaneh Hajishirzi, and Ali Farhadi. A diagram is worth a dozen images, 2016.

[23] Fanheng Kong, Jingyuan Zhang, Yahui Liu, Zirui Wu, Yu Tian, Wei Bi, and Guorui Zhou. Accelerating diffusion language models inference via local determinism propagation.

[24] Inception Labs, Samar Khanna, Siddhant Kharbanda, Shufan Li, Harshit Varma, Eric Wang, Sawyer Birnbaum, Ziyang Luo, Yanis Miraoui, Akash Palrecha, et al. Mercury: Ultra-fast language models based on diffusion. *arXiv preprint arXiv:2506.17298*, 2025.

[25] Xunhao Lai, Jianqiao Lu, Yao Luo, Yiyuan Ma, and Xun Zhou. Flexprefill: A context-aware sparse attention mechanism for efficient long-sequence inference. *arXiv preprint arXiv:2502.20766*, 2025.

[26] Bohao Li, Rui Wang, Guangzhi Wang, Yuying Ge, Yixiao Ge, and Ying Shan. Seed-bench: Benchmarking multimodal llms with generative comprehension. *arXiv preprint arXiv:2307.16125*, 2023.

[27] Pengxiang Li, Yefan Zhou, Dilxat Muhtar, Lu Yin, Shilin Yan, Li Shen, Soroush Vosoughi, and Shiwei Liu. Diffusion language models know the answer before decoding. *arXiv preprint arXiv:2508.19982*, 2025.

[28] 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. *arXiv preprint arXiv:2505.16839*, 2025.

[29] Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pages 292–305, 2023.

[30] Yucheng Li, Huiqiang Jiang, Chengruidong Zhang, Qianhui Wu, Xufang Luo, Surin Ahn, Amir H Abdi, Dongsheng Li, Jianfeng Gao, Yuqing Yang, et al. Mminference: Accelerating pre-filling for long-context vlms via modality-aware permutation sparse attention. *arXiv preprint arXiv:2504.16083*, 2025.

[31] Hanchao Liu, Wenyuan Xue, Yifei Chen, Dapeng Chen, Xiutian Zhao, Ke Wang, Liping Hou, Rongjun Li, and Wei Peng. A survey on hallucination in large vision-language models. *arXiv preprint arXiv:2402.00253*, 2024.

[32] Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. Llava-next: Improved reasoning, ocr, and world knowledge, January 2024. URL `https://llava-vl.github.io/blog/2024-01-30-llava-next/`.

[33] 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*, pages 216–233. Springer, 2024.

[34] Zhiyuan Liu, Yicun Yang, Yaojie Zhang, Junjie Chen, Chang Zou, Qingyuan Wei, Shaobo Wang, and Linfeng Zhang. dllm-cache: Accelerating diffusion large language models with adaptive caching. *arXiv preprint arXiv:2506.06295*, 2025.

[35] Xinyin Ma, Runpeng Yu, Gongfan Fang, and Xinchao Wang. dkv-cache: The cache for diffusion language models. *arXiv preprint arXiv:2505.15781*, 2025.

[36] Yuxin Ma, Lun Du, Lanning Wei, Kun Chen, Qian Xu, Kangyu Wang, Guofeng Feng, Guoshan Lu, Lin Liu, Xiaojing Qi, et al. dinfer: An efficient inference framework for diffusion language models. *arXiv preprint arXiv:2510.08666*, 2025.

[37] Quan Nguyen-Tri, Mukul Ranjan, and Zhiqiang Shen. Attention is all you need for kv cache in diffusion llms. *arXiv preprint arXiv:2510.14973*, 2025.

[38] 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. *arXiv preprint arXiv:2502.09992*, 2025.

[39] Yifan Pu, Zhuofan Xia, Jiayi Guo, Dongchen Han, Qixiu Li, Duo Li, Yuhui Yuan, Ji Li, Yizeng Han, Shiji Song, et al. Efficient diffusion transformer with step-wise dynamic attention mediators. In *European Conference on Computer Vision*, pages 424–441. Springer, 2024.

[40] Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards vqa models that can read. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8309–8318. IEEE, 2019.

[41] Yuerong Song, Xiaoran Liu, Ruixiao Li, Zhigeng Liu, Zengfeng Huang, Qipeng Guo, Ziwei He, and Xipeng Qiu. Sparse-dllm: Accelerating diffusion llms with dynamic cache eviction. *arXiv preprint arXiv:2508.02558*, 2025.

[42] 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.

[43] Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, et al. Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786*, 2025.

[44] Xu Wang, Chenkai Xu, Yijie Jin, Jiachun Jin, Hao Zhang, and Zhijie Deng. Diffusion llms can do faster-than-ar inference via discrete diffusion forcing. *arXiv preprint arXiv:2508.09192*, 2025.

[45] Qingyan Wei, Yaojie Zhang, Zhiyuan Liu, Dongrui Liu, and Linfeng Zhang. Accelerating diffusion large language models with slowfast sampling: The three golden principles. *arXiv preprint arXiv:2506.10848*, 2025.

[46] 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. *arXiv preprint arXiv:2509.26328*, 2025.

[47] 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. *arXiv preprint arXiv:2505.22618*, 2025.

[48] Chengyue Wu, Shiyi Lan, Yonggan Fu, Sensen Gao, Jin Wang, Jincheng Yu, Jose M Alvarez, Pavlo Molchanov, Ping Luo, Song Han, et al. Fast-dvlm: Efficient block-diffusion vlm via direct conversion from autoregressive vlm. *arXiv preprint arXiv:2604.06832*, 2026.

[49] Shutong Wu and Jiawei Zhang. Free draft-and-verification: Toward lossless parallel decoding for diffusion large language models. *arXiv preprint arXiv:2510.00294*, 2025.

[50] Jingqi Xu, Jingxi Lu, Chenghao Li, Sreetama Sarkar, Souvik Kundu, and Peter A Beerel. Redvtp: Training-free acceleration of diffusion vision-language models inference via masked token-guided visual token pruning. *arXiv preprint arXiv:2511.12428*, 2025.

[51] Ling Yang, Ye Tian, Bowen Li, Xinchen Zhang, Ke Shen, Yunhai Tong, and Mengdi Wang. Mmada: Multimodal large diffusion language models. *arXiv preprint arXiv:2505.15809*, 2025.

[52] Zebin You, Shen Nie, Xiaolu Zhang, Jun Hu, Jun Zhou, Zhiwu Lu, Ji-Rong Wen, and Chongxuan Li. Llada-v: Large language diffusion models with visual instruction tuning. *arXiv preprint arXiv:2505.16933*, 2025.

[53] Zhihang Yuan, Hanling Zhang, Lu Pu, Xuefei Ning, Linfeng Zhang, Tianchen Zhao, Shengen Yan, Guohao Dai, and Yu Wang. Ditfastattn: Attention compression for diffusion transformer models. *Advances in Neural Information Processing Systems*, 37:1196–1219, 2024.

[54] Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, et al. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9556–9567, 2024.

[55] Hanling Zhang, Rundong Su, Zhihang Yuan, Pengtao Chen, Mingzhu Shen, Yibo Fan, Shengen Yan, Guohao Dai, and Yu Wang. Ditfastattnv2: Head-wise attention compression for multi-modality diffusion transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16399–16409, 2025.

A Limitations and Potential Negative Social Impact

A.1 Limitations

Our work accelerates inference for multimodal dLLMs without modifying the underlying model weights or training data. As such, it is not intended to introduce new capabilities beyond those already present in the base model. Rather than improving the model's intrinsic reasoning or multimodal understanding ability, our method focuses on making existing capabilities more efficient at inference time. This limitation is shared by prior training-free acceleration methods [35, 34, 47, 37, 50, 7], whose primary goal is to reduce latency and computation while preserving model quality as much as possible, instead of advancing the underlying performance frontier.

A.2 Potential Negative Social Impact

By substantially lowering the computational cost of deploying multimodal dLLMs, DRAMA may also reduce the barrier for malicious use, including the large-scale generation of misleading or deceptive visual question answering outputs [31, 4]. Such misuse could amplify the spread of incorrect visual interpretations or plausible-sounding but unreliable multimodal responses. This concern is not specific to our method, but is shared broadly by inference acceleration techniques that improve the accessibility and scalability of foundation models.

B Additional Implementation Details

B.1 Datasets and Evaluation Splits

We evaluate on six standard vision-language benchmarks. For AI2D [22], MMBench [33], MME [13], MMMU [54], and MM-Star [8], we use the full evaluation split released by each benchmark. For SEED-Bench [26], we evaluate on a fixed random subset of 2,000 samples (seed fixed for reproducibility) to manage computational cost; the same subset is used across all baselines, ablations, and DRAMA variants to ensure fair comparison. Table 2 summarizes the splits and sample counts.

Table 2: Evaluation splits used in our experiments.

Benchmark	Split	#Samples
AI2D [22]	test (full)	3,088
MMBench [33]	dev-en (full)	4,329
MME [13]	full	2,374
MMMU [54]	validation (full)	1,050
MM-Star [8]	full	1,500
SEED-Bench [26]	random subset (seed-fixed)	2,000

B.2 Prompt Template

Following LLaDA-V [52], we use a multi-turn chat format with a system message ("*You are a helpful language and vision assistant...*"), a user turn containing the image token, question, and a chain-of-thought thinking instruction, and an empty assistant turn for the model to complete. All experiments use a fixed generation length of $N_g = 128$ tokens with deterministic decoding (temperature 0.0) and low-confidence remasking. Table 3 shows an example of a prompt template used for our evaluation benchmarks.

B.3 SGAS Configuration Details

We summarize the four SGAS variants evaluated in the main paper in Table 4. Under progressive scheduling, the stopping criterion is strict at early evaluation steps, requiring both semantic sufficiency and high lexical coherence, and progressively relaxed at later steps, reflecting the diminishing marginal return of additional denoising iterations identified in the temporal analysis of the main paper. Under uniform scheduling, fixed thresholds are applied throughout all evaluation steps. Checkpoint mode

Table 3: **An example of prompt template**

Component	Content
Image token	`<image>`
Question	`Question: {question}`

Options	`Options:`
	`A. {option_a}`
	`B. {option_b}`
	`...`

Thinking instruction	*You should first think about the reasoning process in the mind and then provide the user with the answer. The reasoning process is enclosed within* `<think> </think>` *tags, i.e.* `<think>` *reasoning process here* `</think>` `<answer>` *answer here:* Please select the correct option by responding with only the letter (A, B, C, D, or E) `</answer>`

Table 4: SGAS configuration variants. All variants share $T = 64$, $T_{\min} = 1$.

Variant	Granularity	Schedule	Evaluation Steps
CK Prog	Checkpoint	Progressive	$\mathcal{C} = \{16, 32, 48, 64\}$
CK Uni	Checkpoint	Uniform	$\mathcal{C} = \{16, 32, 48, 64\}$
FG Prog	Fine-grained	Progressive	Every step $t > T_{\min}$
FG Uni	Fine-grained	Uniform	Every step $t > T_{\min}$

only evaluates at the sparse set $\mathcal{C}$, while fine-grained mode evaluates at every step beyond $T_{\min}$, providing more exit opportunities at the cost of slightly more frequent criterion checks.

B.3.1 Stopping Decision Procedure

We formalize the full SGAS procedure in Algorithm 1. After each denoising step, if $t \geq T_{\min}$ and t is admitted by the granularity schedule $\mathcal{E}$, the decoded sequence is evaluated against two complementary checks. The *semantic sufficiency* check parses the decoded text under the task schema. The *lexical coherence* check verifies three statistics of $\mathbf{x}_t$ against the threshold vector $\boldsymbol{\theta}_t = (\theta_t^{\text{div}}, \theta_t^{\text{rep}}, \theta_t^{\text{con}})$:

- Vocabulary diversity r^{div}: ratio of unique tokens to total non-mask tokens.
- Max single-token frequency r^{rep}: fraction of the most repeated token.
- Consecutive-repetition ratio r^{con}: fraction of adjacent token pairs that are identical.

The coherence gate passes when $r^{\text{div}} \geq \theta_t^{\text{div}}$, $r^{\text{rep}} \leq \theta_t^{\text{rep}}$, and $r^{\text{con}} \leq \theta_t^{\text{con}}$ simultaneously. If both checks pass, denoising terminates and $\mathbf{x}_t$ is returned; otherwise the procedure proceeds to step $t+1$. If no step in the trajectory triggers the predicate, $\mathbf{x}_T$ is returned, recovering the unaccelerated baseline as a worst case.

B.3.2 Threshold Parameters

Table 5 lists the threshold vectors used by each variant. Checkpoint variants use a set of per-checkpoint thresholds; fine-grained uniform uses a single fixed vector; and fine-grained progressive smoothly interpolates thresholds as a function of denoising progress $p = t/T$.

B.4 Compatibility with Block Diffusion

Block diffusion [2] is a semi-autoregressive generation paradigm for dLLMs in which the N_g generation tokens are partitioned into B blocks of size $b = N_g/B$, with each block denoised sequentially over its own trajectory of T_b steps; it is a generation strategy rather than an acceleration mechanism. Our experiments adopt the default single-block setting ($B = 1$) of LLaDA-V [52], but DRAMA extends to the multi-block regime without modification. The visual prefix is shared across all blocks, so SVTP performs its cross-modal scoring once before the first block and reuses the selected patch set $\mathcal{P}$ for every subsequent block. SGAS evaluates its stop predicate inside each block

Algorithm 1 Sufficiency-Guided Adaptive Stopping

Require: Denoising budget T, minimum steps $T_{\min}$, evaluation set $\mathcal{E}$ ▷ $\mathcal{C}$ or $\{T_{\min}+1, \ldots, T\}$
Require: Threshold schedule $\theta(\cdot) = (\theta^{\text{div}}, \theta^{\text{rep}}, \theta^{\text{con}})$ ▷ progressive or uniform

$\mathbf{x}_0 \leftarrow [\text{MASK}]^{N_g}$
for $t = 1, \ldots, T$ **do**
  $\mathbf{x}_t \leftarrow \text{DENOISE}(\mathbf{x}_{t-1})$ ▷ Predict & unmask $\sim N_g/T$ tokens
  if $t < T_{\min}$ **or** $t \notin \mathcal{E}$ **then**
    continue
  end if
  $(\theta_t^{\text{div}}, \theta_t^{\text{rep}}, \theta_t^{\text{con}}) \leftarrow \theta(t)$
  text $\leftarrow \text{DECODE}(\mathbf{x}_t)$
  sufficient $\leftarrow \text{SEMANTICSUFFICIENCYCHECK}(\textit{text}, \textit{task})$
  $r^{\text{div}} \leftarrow \text{VOCABULARYDIVERSITY}(\mathbf{x}_t)$
  $r^{\text{rep}} \leftarrow \text{MAXTOKENFREQUENCY}(\mathbf{x}_t)$
  $r^{\text{con}} \leftarrow \text{CONSECUTIVEREPETITIONRATIO}(\mathbf{x}_t)$
  coherent $\leftarrow (r^{\text{div}} \geq \theta_t^{\text{div}}) \wedge (r^{\text{rep}} \leq \theta_t^{\text{rep}}) \wedge (r^{\text{con}} \leq \theta_t^{\text{con}})$
  if *sufficient* **and** *coherent* **then**
    return $\mathbf{x}_t$ ▷ Early termination
  end if
end for
return $\mathbf{x}_T$ ▷ Fallback: full budget

Table 5: Lexical coherence thresholds for all SGAS variants ($T = 64$).

Variant	Step / Phase	$\theta^{\text{div}} \uparrow$	$\theta^{\text{rep}} \downarrow$	$\theta^{\text{con}} \downarrow$
CK Uni	$t = 16$	0.50	0.22	0.09
	$t = 32$	0.50	0.22	0.09
	$t = 48$	0.50	0.22	0.09
	$t = 64$	0.50	0.22	0.09
CK Prog	$t = 16$	0.50	0.23	0.09
	$t = 32$	0.55	0.20	0.08
	$t = 48$	0.50	0.23	0.09
	$t = 64$	0.45	0.25	0.10
FG Uni	All steps	0.50	0.22	0.09
FG Prog	$p \leq 0.5$ (tighten)	$0.45 \rightarrow 0.55$	$0.25 \rightarrow 0.20$	$0.10 \rightarrow 0.08$
	$p > 0.5$ (relax)	$0.55 \rightarrow 0.45$	$0.20 \rightarrow 0.25$	$0.08 \rightarrow 0.10$

independently, using the same threshold schedule $\boldsymbol{\theta}(t)$, so each block terminates at its own first-hit step $t_b^\star$ without coupling across blocks. The aggregate skipped-step count is then $\sum_{b=1}^{B}(T_b - t_b^\star)$.

B.5 Hyperparameter selection.

All settings reported in this paper are fixed once and shared across every benchmark and both backbones; we perform no per-benchmark tuning and no search. SVTP retains a single patch, $|\mathcal{P}| = 1$, in all experiments. The SGAS threshold vectors of Table 5 were set from the output statistics of 100 LLaDA-V samples on MMMU, and the resulting values were then applied unchanged to the remaining five benchmarks and to LaViDa. Accuracy on the benchmarks we report was therefore never consulted when choosing these values.

C Detailed Ablation Study

C.1 Model Architecture Ablation

We further evaluate DRAMA on LaViDa [28], a multimodal dLLM that already employs 2×2 visual-token pooling and therefore exposes a substantially smaller acceleration headroom than LLaDA-V (LaViDa baseline averages 9.76 s vs. LLaDA-V's 58.2 s under the same think-mode regime). All

Table 6: Benchmark results on LaViDa (128 denoising steps, think mode). Sample counts are given in the benchmark headers. Latency is wall-clock seconds per sample and TPS is computed from the fixed generation length of 128 tokens under the measured latency. Metrics: Acc (%, ↑), Latency (s/sample, ↓), TPS (tokens/s, ↑), Speedup vs. the 128-step baseline (↑), Acc Retention (↑).

Metric	Baseline	Ablation					DRAMA (Ours)			
	128 steps	w/o SGAS	w/o SVTP FG Prog	w/o SVTP FG Uni	w/o SVTP CK Prog	w/o SVTP CK Uni	FG Prog	FG Uni	CK Prog	CK Uni
AI2D ($n = 3088$)										
Acc (%)	**69.88**	69.04	68.36	68.72	68.49	68.75	67.58	67.81	67.81	67.45
Latency (s)	9.98	7.40	7.04	7.07	7.49	7.50	5.30	**5.23**	5.50	5.54
TPS (tok/s)	12.82	17.29	18.17	18.11	17.08	17.06	24.13	**24.49**	23.27	23.11
Speedup	1.00×	1.35×	1.42×	1.41×	1.33×	1.33×	1.88×	**1.91**×	1.82×	1.80×
Acc Retention	**1.00**	0.99	0.98	0.98	0.98	0.98	0.97	0.97	0.97	0.97
MME ($n = 2374$)										
Acc (%)	68.91	**69.46**	68.32	68.62	66.51	68.62	68.53	69.17	62.22	69.17
Latency (s)	9.82	7.15	7.06	7.24	7.04	7.72	**5.34**	5.40	5.39	5.70
TPS (tok/s)	13.04	17.89	18.12	17.67	18.19	16.58	**23.97**	23.72	23.76	22.44
Speedup	1.00×	1.37×	1.39×	1.35×	1.40×	1.27×	**1.84**×	1.82×	1.82×	1.72×
Acc Retention	1.00	**1.01**	0.99	1.00	0.97	1.00	0.99	1.00	0.90	1.00
MMMU ($n = 900$)										
Acc (%)	**42.44**	41.78	41.67	42.11	42.00	**42.44**	41.11	41.22	41.44	42.00
Latency (s)	9.91	7.68	7.06	7.19	7.47	7.60	**5.52**	5.65	5.77	5.84
TPS (tok/s)	12.91	16.66	18.13	17.79	17.14	16.84	**23.19**	22.66	22.18	21.90
Speedup	1.00×	1.29×	1.40×	1.38×	1.33×	1.30×	**1.80**×	1.75×	1.72×	1.70×
Acc Retention	**1.00**	0.98	0.98	0.99	0.99	**1.00**	0.97	0.97	0.98	0.99
MM-Star ($n = 1500$)										
Acc (%)	**43.00**	40.73	41.33	42.47	41.33	42.20	39.13	40.00	39.00	40.00
Latency (s)	9.19	7.31	6.50	6.58	6.87	6.99	**5.23**	5.36	5.63	5.57
TPS (tok/s)	13.93	17.52	19.68	19.45	18.64	18.32	**24.46**	23.89	22.74	22.97
Speedup	1.00×	1.26×	1.41×	1.40×	1.34×	1.32×	**1.76**×	1.72×	1.63×	1.65×
Acc Retention	**1.00**	0.95	0.96	0.99	0.96	0.98	0.91	0.93	0.91	0.93
SEED-Bench ($n = 1000$)										
Acc (%)	**63.50**	61.00	61.40	63.40	61.40	63.40	59.10	61.00	59.10	61.00
Latency (s)	9.90	7.34	7.09	7.21	7.52	7.67	**5.19**	5.24	5.55	5.61
TPS (tok/s)	12.93	17.45	18.06	17.75	17.02	16.68	**24.67**	24.45	23.06	22.82
Speedup	1.00×	1.35×	1.40×	1.37×	1.32×	1.29×	**1.91**×	1.89×	1.78×	1.77×
Acc Retention	**1.00**	0.96	0.97	1.00	0.97	1.00	0.93	0.96	0.93	0.96
Average (over the five benchmarks above)										
Acc (%)	**57.55**	56.40	56.22	57.06	55.95	57.08	55.09	55.84	53.91	55.92
Latency (s)	9.76	7.38	6.95	7.06	7.28	7.50	**5.32**	5.38	5.57	5.65
TPS (tok/s)	13.11	17.35	18.42	18.14	17.59	17.08	**24.08**	23.81	22.99	22.65
Speedup	1.00×	1.32×	1.40×	1.38×	1.34×	1.30×	**1.84**×	1.82×	1.75×	1.73×
Acc Retention	**1.00**	0.98	0.98	0.99	0.97	0.99	0.96	0.97	0.94	0.97

LaViDa results are evaluated on the full (AI2D 3088; MME 2374; MM-Star 1500; MMMU 900) benchmark and a 1,000-sample SEED-Bench subset . Despite the compressed margin, DRAMA delivers consistent 1.73–1.84× wall-clock speedup on average across five benchmarks (Table 6), with accuracy preserved to 97.2% overall by CK Uni and 97.0% by FG Uni, and with MME improving over the LaViDa baseline of 68.91, reaching 69.46 under SVTP alone and 69.17 under DRAMA FG Uni and CK Uni, indicating that pruning redundant visual tokens can mitigate attention dilution even when the backbone has already partially compressed its visual context. The pattern of variant trade-offs largely mirrors LLaDA-V: FG Uni and FG Prog dominate the speedup axis, while CK Uni retains the most accuracy, so the original split between the two schedules still holds under full-set evaluation.

Table 7: Per-benchmark comparison with RedVTP on LLaDA-V, both reproduced by us. RedVTP runs at its most aggressive operating point (r=0.25) under the same 32-step think-mode regime as DRAMA.

Benchmark	Method	Setting	Performance	Latency (s)	Speedup	Acc. Retention
AI2D	LLaDA-V (32-step)	500 samples	73.00	49.194	1.00×	100.0%
	+ RedVTP (r=0.25)	500 samples	71.60	38.319	1.28×	98.1%
	LLaDA-V (32-step)	full set (3,088)	72.73	41.94	1.00×	100.0%
	+ **DRAMA (ours)**	full set (3,088)	**73.02**	**2.57**	**16.32×**	**100.4%**
MM-Star	LLaDA-V (32-step)	500 samples	50.00	34.663	1.00×	100.0%
	+ RedVTP (r=0.25)	500 samples	42.40	28.718	1.21×	84.8%
	LLaDA-V (32-step)	full set (1,500)	50.27	29.29	1.00×	100.0%
	+ **DRAMA (ours)**	full set (1,500)	47.60	**3.52**	**8.32×**	**94.7%**
Average	+ RedVTP (r=0.25)	500 samples	—	—	1.25×	91.5%
	+ **DRAMA (ours)**	full set	—	—	**12.32×**	**97.6%**

Table 8: Same-backbone wall-clock speedup comparison with RedVTP on LaViDa. RedVTP reports only $r = 75\%$ on LaViDa, since LaViDa's built-in 2×2 visual-token pooling already retains 25% of the original tokens. RedVTP LaViDa numbers are taken from Xu et al. [50]

; DRAMA runs under 128-step think mode. The AI2D and MME columns are measured on every sample (3,088 and 2,374); the MMBench column is measured on a 100-sample subset against a baseline on the same subset. Best , 2nd best .

Method	AI2D	MME	MMBench
RedVTP regime (LaViDa)			
LaViDa (baseline)	1.00×	1.00×	1.00×
+ RedVTP ($r = 75\%$)	1.08×	1.06×	1.06×
DRAMA regime (LaViDa, 128-step think mode)			
LaViDa (baseline)	1.00×	1.00×	1.00×
DRAMA CK Uni	1.80×	1.72×	1.56×
DRAMA CK Prog	1.82×	1.82×	1.58×
DRAMA FG Prog	1.88×	**1.84**×	1.69×
DRAMA FG Uni	**1.91**×	1.82×	1.68×

C.2 Comparison with Concurrent Acceleration Methods

C.2.1 Comparison with RedVTP [50]

RedVTP [50] accelerates multimodal dLLMs by pruning visual tokens, and is the closest prior method on our backbone. We re-run it ourselves at its most aggressive operating point (r=0.25) under our own 32-step think-mode setting, against a baseline with the same batch and the same 500 samples. Table 7 pairs each method with its own baseline on LLaDA-V. RedVTP buys 1.21–1.28× while retaining 84.8–98.1% of baseline accuracy, averaging 1.25× at 91.5%. DRAMA reaches 8.32–16.32× at 94.7–100.4% retention, averaging 12.32× at 97.6%. Pruning the visual prefix removes work that is paid once per sample, so its gain is bounded by the share of the forward pass that the prefix occupies; the step axis is not bounded in the same way, and the two axes compose. The same ordering holds on LaViDa (Table 8), where RedVTP reports only r=75% because the 2×2 visual-token pooling already built into LaViDa absorbs most of its pruning headroom. Its wall-clock gain there is a marginal 1.06–1.08×, against 1.56–1.91× for DRAMA under a heavier reasoning regime. That the ordering survives a backbone whose visual prefix is already compressed supports the reading that the temporal redundancy identified in Section 3.2 is a separate, multiplicative source of acceleration rather than a restatement of spatial redundancy.

C.2.2 Comparison with D^{3}ToM [7]

D3ToM [7] accelerates LaViDa by merging visual tokens under decider-token guidance. It reports wall-clock time only on LLaVA-Bench in a short-generation setting (T=32, $|O|$=64), so its published speedups do not transfer to think mode. We therefore re-run D3ToM-t at its own default operating

Table 9: Comparison with D3ToM on LaViDa. All rows are re-run by us under 128-step think mode, D3ToM included, so that accuracy and speedup are measured under one regime against one baseline; retention is accuracy divided by that baseline. Rows use the full set for AI2D, MME, MMMU, MMstar, and a 1000-sample subset for SEED-Bench. Best, 2nd best.

	AI2D		MMMU		MM-Star		SEED-Bench		Avg.
Method	Acc (Ret.)	Sp.	Acc (Ret.)	Sp.	Acc (Ret.)	Sp.	Acc (Ret.)	Sp.	Sp.
LaViDa (baseline)	69.88 (1.00)	1.00×	42.44 (1.00)	1.00×	43.00 (1.00)	1.00×	63.50 (1.00)	1.00×	1.00×
D3ToM-t ($keep$=0.25, L3)	66.80 (0.96)	1.20×	39.89 (0.94)	1.15×	38.80 (0.81)	1.21×	53.00 (0.83)	1.23×	1.20×
DRAMA CK Uni	67.45 (0.97)	1.80×	**42.00 (0.99)**	1.70×	**40.00 (0.93)**	1.65×	**61.00 (0.96)**	1.77×	1.73×
DRAMA CK Prog	**67.81 (0.97)**	1.82×	41.44 (0.98)	1.72×	39.00 (0.91)	1.63×	59.10 (0.93)	1.78×	1.74×
DRAMA FG Prog	67.58 (0.97)	1.88×	41.11 (0.97)	**1.80×**	39.13 (0.91)	**1.76×**	59.10 (0.93)	**1.91×**	**1.84×**
DRAMA FG Uni	**67.81 (0.97)**	**1.91×**	41.22 (0.97)	1.75×	**40.00 (0.93)**	1.72×	**61.00 (0.96)**	1.89×	1.82×

Table 10: Held-out evaluation on POPE and TextVQA, both on LLaDA-V. Neither benchmark was used when setting any threshold. Speedup is wall-clock against the 64-step baseline of the same benchmark. POPE reports F1 on 500 samples; TextVQA reports accuracy on 100 samples. "SGAS only" is DRAMA with SVTP disabled.

	POPE (F1)		**TextVQA** (Acc)	
Setting	Score	Speedup	Score	Speedup
Baseline (64 steps)	88.30	1.00×	64.60	1.00×
Baseline (32 steps)	86.37	1.90×	60.70	2.00×
Baseline (16 steps)	87.24	3.77×	–	–
SGAS only (FG Uni)	88.30	3.81×	–	–
DRAMA FG Prog	85.35	10.69×	51.60	7.91×
DRAMA FG Uni	85.35	11.23×	52.50	5.87×

point (25% of visual tokens retained, merged at layer 3) under our 128-step think mode and against our own baseline (Table 9). It gains only 1.15–$1.23\times$ against DRAMA's 1.63–$1.91\times$, and retains less accuracy on every benchmark, falling to 0.81 and 0.83 on MM-Star and SEED-Bench where DRAMA holds 0.91–0.96. On AI2D and MMMU, the two benchmarks D3ToM also evaluates, our reproduction reaches 0.96 and 0.94 against the 0.94–0.96 and 0.97–0.98 they report.

C.3 Generalization to Additional Benchmarks

Neither TextVQA [40] nor POPE [29] was used when setting any threshold, so both serve as held-out checks on whether the redundancy structures of Section 3.1 3.2 transfer beyond the benchmarks we report in the main paper. They probe opposite ends of the same question. POPE asks whether a named object is present, which depends on the visual prefix surviving pruning; TextVQA asks open-ended questions about dense text inside the image, which depends on both the prefix and a sufficient number of denoising steps. Table 10 reports both.

On POPE, SGAS alone is lossless: all four schedules reproduce the 64-step F1 of 88.30 while reaching up to $3.81\times$, confirming that the temporal stabilization measured in Section **??** holds on a task the thresholds were never tuned for. Adding SVTP costs 2.95 F1 and buys $11.23\times$ speedup. That the loss appears only once the visual prefix is pruned is consistent with the nature of the task: object-existence judgements read directly off the prefix, so they are the first to suffer when it is compressed.

TextVQA is where DRAMA is weakest: accuracy falls from 64.60 at 64 steps to 52.50 for FG Uni at $5.87\times$, and halving the step budget alone already accounts for 3.90 of those points. As stated in Appendix A.1, DRAMA adds no mechanism to recover quality lost to a reduced budget, and on open-ended text-dense inputs that cost is simply larger than on the six main benchmarks.

D Derivation of the Output-Stationarity Bound

This appendix supplies the derivation of the Pinsker-type output-stationarity bound invoked in Section 3.3.2, showing that the per-step layer-output drift vanishes as a direct corollary of the JSD collapse identified in Section 3.2.

D.1 Setup and Decomposition

At denoising step t and transformer layer l, let $\mathbf{A}_l^{(t)} \in \mathbb{R}^{H\times N\times N}$ be the attention tensor, $\mathbf{V}_l^{(t)} \in \mathbb{R}^{H\times N\times d_v}$ the value tensor, and $\mathbf{O}_l^{(t)} = \mathbf{A}_l^{(t)}\mathbf{V}_l^{(t)}$ the layer output. Adding and subtracting $\mathbf{A}_l^{(t)}\mathbf{V}_l^{(t-1)}$ yields the canonical decomposition

$$\Delta\mathbf{O}_l^{(t)} \;\equiv\; \mathbf{O}_l^{(t)} - \mathbf{O}_l^{(t-1)} \;=\; \underbrace{\Delta\mathbf{A}_l^{(t)}\mathbf{V}_l^{(t-1)}}_{\text{attention drift}} \;+\; \underbrace{\mathbf{A}_l^{(t)}\Delta\mathbf{V}_l^{(t)}}_{\text{value update}}, \tag{6}$$

with $\Delta\mathbf{A}_l^{(t)} = \mathbf{A}_l^{(t)} - \mathbf{A}_l^{(t-1)}$ and $\Delta\mathbf{V}_l^{(t)} = \mathbf{V}_l^{(t)} - \mathbf{V}_l^{(t-1)}$. The triangle inequality, combined with submultiplicativity in the Frobenius/operator-norm pair, gives

$$\big\|\Delta\mathbf{O}_l^{(t)}\big\|_F \;\le\; \big\|\Delta\mathbf{A}_l^{(t)}\big\|_F \big\|\mathbf{V}_l^{(t-1)}\big\|_{\mathrm{op}} \;+\; \big\|\mathbf{A}_l^{(t)}\big\|_{\mathrm{op}} \big\|\Delta\mathbf{V}_l^{(t)}\big\|_F. \tag{7}$$

The two summands are bounded separately in the next two subsections.

D.2 Bounding the Attention-Drift Term via JSD

For each (head h, query i), the per-row attention vectors $\mathbf{a}_{l,h,i}^{(t)}, \mathbf{a}_{l,h,i}^{(t-1)} \in \Delta^{N-1}$ are probability distributions over key positions. They satisfy the Lin (1991) inequality, the JSD analogue of Pinsker's bound,

$$\mathrm{JSD}\big(\mathbf{a}_{l,h,i}^{(t)} \,\|\, \mathbf{a}_{l,h,i}^{(t-1)}\big) \;\ge\; \tfrac{1}{2}\,\mathrm{TV}\big(\mathbf{a}_{l,h,i}^{(t)}, \mathbf{a}_{l,h,i}^{(t-1)}\big)^2. \tag{8}$$

Combined with $\|\mathbf{a}_{l,h,i}^{(t)} - \mathbf{a}_{l,h,i}^{(t-1)}\|_2 \le \|\cdot\|_1 = 2\,\mathrm{TV}(\cdot)$ and aggregation over all rows,

$$\big\|\Delta\mathbf{A}_l^{(t)}\big\|_F^2 \;=\; \sum_{h,i} \big\|\mathbf{a}_{l,h,i}^{(t)} - \mathbf{a}_{l,h,i}^{(t-1)}\big\|_2^2 \;\le\; 8\,HN\,\bar{J}_l(t), \tag{9}$$

where $\bar{J}_l(t)$ is the mean per-layer divergence defined in Eq. (3). Substituting Eq. (9) into the first summand of Eq. (7) recovers the headline form used in the main text,

$$\big\|\Delta\mathbf{A}_l^{(t)}\mathbf{V}_l^{(t-1)}\big\|_F \;\le\; C_l\,\sqrt{2\,\bar{J}_l(t)}, \qquad C_l \;:=\; 2\sqrt{HN}\cdot\big\|\mathbf{V}_l^{(t-1)}\big\|_{\mathrm{op}}. \tag{10}$$

Standard layer-norm and weight-decay regularisation keep $\|\mathbf{V}_l^{(t-1)}\|_{\mathrm{op}}$ in a bounded operator-norm regime across steps, so C_l is a step-independent constant for the purposes of the argument.

D.3 Bounding the Value-Update Term via Layer-wise Propagation

The value tensor at layer l is computed from the previous layer's output through a Lipschitz block f_l (linear projection composed with layer norm and, in pre-norm transformers, residual addition): $\mathbf{V}_l^{(t)} = f_l(\mathbf{O}_{l-1}^{(t)})$ with Lipschitz constant $L_l < \infty$. Hence

$$\big\|\Delta\mathbf{V}_l^{(t)}\big\|_F \;\le\; L_l\,\big\|\Delta\mathbf{O}_{l-1}^{(t)}\big\|_F. \tag{11}$$

Inserting Eqs. (10) and (11) into Eq. (7) produces the layer-wise recurrence

$$\big\|\Delta\mathbf{O}_l^{(t)}\big\|_F \;\le\; C_l\sqrt{2\,\bar{J}_l(t)} \;+\; \rho_l\,\big\|\Delta\mathbf{O}_{l-1}^{(t)}\big\|_F, \qquad \rho_l \;:=\; \big\|\mathbf{A}_l^{(t)}\big\|_{\mathrm{op}} L_l. \tag{12}$$

Unrolling along the layer stack with the boundary condition $\Delta\mathbf{O}_0^{(t)} = \mathbf{0}$ at the input embedding (whose only step-to-step change comes from the $\sim N_g/T$ newly unmasked tokens, which the converged attention assigns negligible aggregate weight) gives

$$\big\|\Delta\mathbf{O}_l^{(t)}\big\|_F \;\le\; \sum_{k=1}^{l}\Big(\prod_{j=k+1}^{l}\rho_j\Big)\,C_k\sqrt{2\,\bar{J}_k(t)}. \tag{13}$$

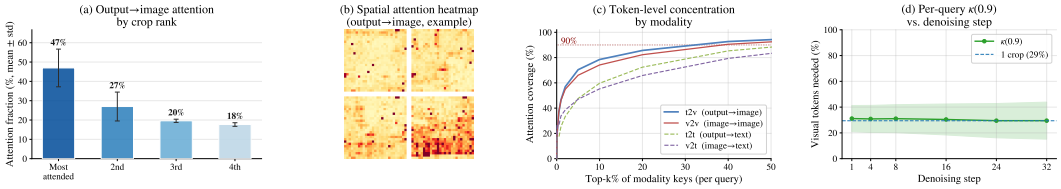


Figure 6: **Spatial redundancy analysis – AI2D.** The diagram-centric nature of AI2D leads to highly concentrated attention on a single informative patch.

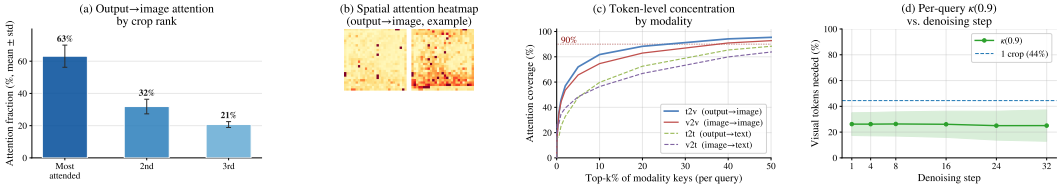


Figure 7: **Spatial redundancy analysis – MMBench.** Despite diverse image content, the top 2–3 patches consistently capture the majority of output→image attention.

The temporal dissection in Section 3.2 establishes that $\bar{J}_k(t) \to 0$ *simultaneously across all* $k \in [L]$ once past the structure-formation phase (the bright plateau region of Fig. 3(b)). Every term in the sum is therefore suppressed in lockstep, and $\|\Delta \mathbf{O}_l^{(t)}\|_F \to 0$ uniformly in l. The simultaneity is essential: it converts what would otherwise be a worst-case accumulating bound into one in which all contributing factors vanish concurrently.

D.4 From Layer Output to Decoded Sequence

The unembedding head is a fixed linear map $\mathbf{W}_U$ followed by softmax, so the per-token logit drift obeys $\|\Delta \mathbf{z}^{(t)}\|_F \le \|\mathbf{W}_U\|_{\mathrm{op}} \|\Delta \mathbf{O}_L^{(t)}\|_F \to 0$. Combined with the dLLM low-confidence remasking schedule, which unmasks only $\sim N_g/T$ positions per step, the argmax-decoded sequence inherits the same stationarity and tends to stabilize before the nominal budget T is exhausted. This justifies the promotion of the decoded text to a sufficiency surrogate in Section 3.3.2, replacing any runtime monitoring of $\bar{J}_l(t)$ that would otherwise require the per-layer attention caching this work seeks to eliminate.

E Additional Redundancy Analysis and Results

E.1 Dissecting Spatial Redundancy: Analysis per Benchmark

We visualize spatial redundancy with MMMU [54] for illustration in the main paper. Below, we provide the same four-panel spatial analysis for all six benchmarks individually. Each figure reports: (a) per-rank crop attention fraction, (b) a representative spatial attention heatmap, (c) token-level concentration CDF stratified by modality pair, and (d) the per-query visual token budget $\kappa(0.9)$ across denoising steps.

The key findings from the main paper generalize consistently: attention mass concentrates on 1–2 dominant patches across all benchmarks, the t2v concentration curve is the steepest among all modality pairs, and $\kappa(0.9)$ remains step-invariant. Benchmarks with simpler visual content (e.g., AI2D [22] diagrams, MME [13] perceptual images) exhibit slightly higher concentration, while those with richer scenes (e.g., SEEDBench [26], MMBench [33]) show moderately broader spatial distributions—yet even in the latter case, the most-attended patch captures $\ge 40\%$ of output→image attention.

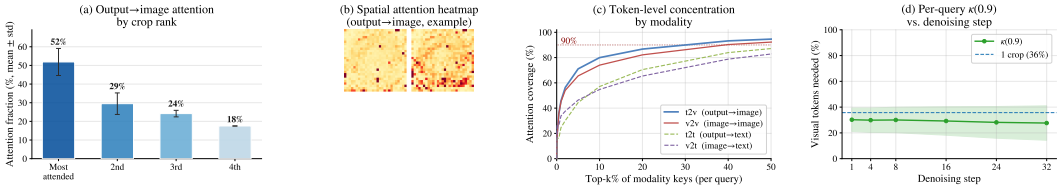


Figure 8: **Spatial redundancy analysis – MME.** Perceptual tasks in MME yield concentrated attention patterns, with $\kappa(0.9)$ remaining stable across all denoising steps.

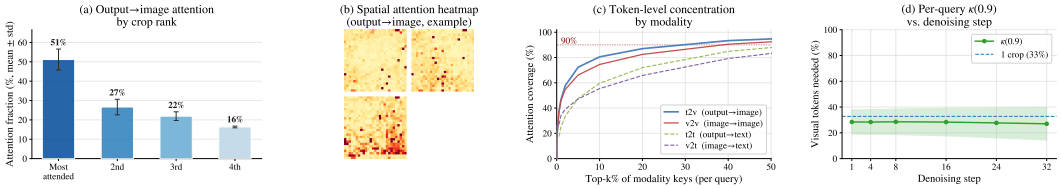


Figure 9: **Spatial redundancy analysis – MM-Star.** Knowledge-intensive queries in MM-Star show similar spatial concentration to MMMU.

E.2 Dissecting Temporal Redundancy: Analysis per Benchmark

We present the temporal analysis averaged over all benchmarks in the main paper. Here we provide per-benchmark breakdowns, each showing (a) the per-layer inter-step JSD $\bar{J}_l(t)$ across all 64 denoising steps, and (b) the layer×step redundancy heatmap $R_l(t)$.

The three-phase temporal structure identified in the main paper—structure formation, stable plateau, and terminal refinement—is clearly visible in every benchmark. Across all datasets, the JSD peaks within the first 2–5 steps, then drops by over two orders of magnitude into a prolonged plateau where $R_l(t) > 0.99$, before a secondary rise in the final 5–10 steps. The heatmaps further reveal that deeper layers (L25–L32) exhibit the lowest redundancy scores throughout the trajectory, consistent with these layers performing the fine-grained disambiguation required for final token selection.

Notably, the plateau onset and duration are remarkably consistent across benchmarks despite their diverse task characteristics (diagram understanding in AI2D [22], perceptual reasoning in MME [13], multi-discipline knowledge in MMMU [54]), confirming that the temporal redundancy is a structural property of the denoising dynamics rather than a task-specific artefact. This consistency is what enables a single set of SGAS hyperparameters to work effectively across all benchmarks without per-task tuning.

E.3 Pareto Frontier

We present the accuracy–latency Pareto frontier averaged across all six benchmarks in Figure 13. The averaged view confirms that DRAMA variants consistently dominate the frontier: no fixed-budget baseline achieves a comparable combination of speed and accuracy. Among DRAMA variants, CK Prog and CK Uni occupy the high-accuracy region ($\sim$64.5%, $\sim$6 s), while FG Uni pushes to the extreme-speed end ($\sim$63.0%, $\sim$3.3 s), demonstrating the flexibility of the framework.

E.4 Step Distribution Analysis per Benchmark

We provide the combined step distribution overview in the main paper. Below we show per-benchmark distributions with full statistics for each of the four SGAS configurations under both the w/o SVTP ablation (top row) and the full DRAMA setting (bottom row).

Each subplot displays a histogram of actual stopping steps overlaid with a kernel density estimate (KDE), together with summary statistics: mean (μ), median, standard deviation (σ), computational

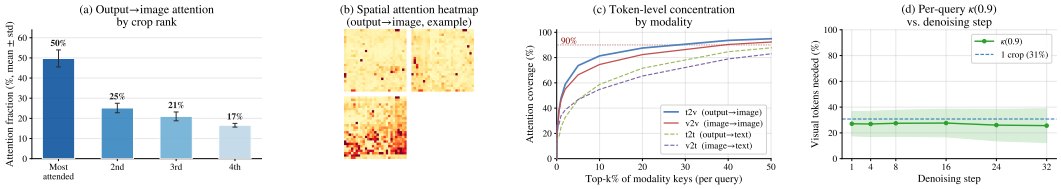


Figure 10: **Spatial redundancy analysis – SEED-Bench.** SEED-Bench exhibits slightly broader spatial distributions due to its diverse scene understanding tasks, yet the dominant patch still captures $\geq 40\%$ of attention.

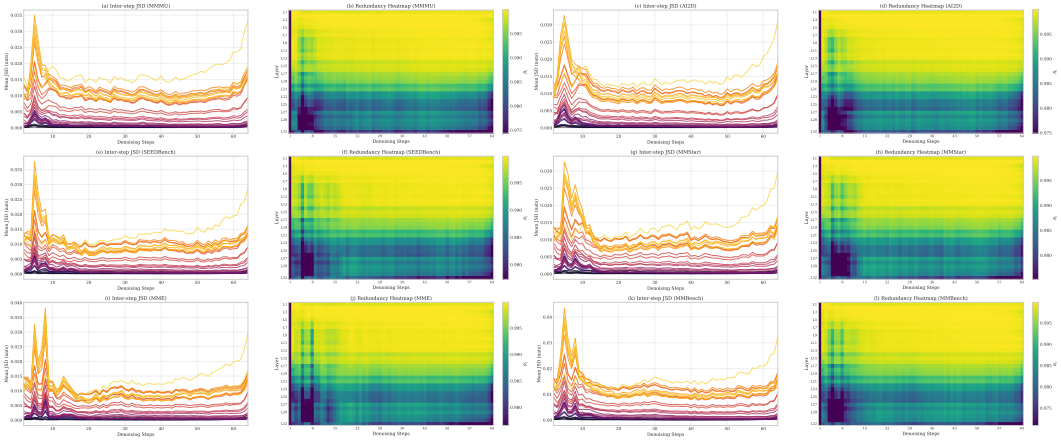


Figure 11: **Temporal redundancy analysis for all benchmarks.** (a) Per-layer inter-step JSD across denoising steps. (b) Layer×step redundancy heatmap.

saving relative to the 64-step baseline, and sample count (N). Several patterns are consistent across benchmarks:

- **DRAMA** (bottom rows) shows similar stopping patterns to the w/o SVTP ablation (top rows), confirming that SVTP and SGAS operate on orthogonal axes (spatial vs. temporal) and compose independently.
- **w/o SVTP FG Uni** concentrates exits at the earliest steps (typically around 7−−11 steps, with a later mean of 15.6 on MM-Star), reflecting its willingness to terminate as soon as both criteria are met, resulting in the highest speedup but occasionally at the cost of accuracy on harder benchmarks.
- **w/o SVTP CK variants** cluster around the first eligible checkpoint (step 16), as the sparse evaluation schedule prevents earlier termination regardless of output convergence.
- **w/o SVTP Prog Variants** consistently shifts the distribution rightward compared to uniform scheduling, reflecting its stricter early-step requirements that act as a safeguard during the structure-formation phase.

The alignment between the stopping step distributions and the JSD plateau onset provides empirical validation that SGAS terminates denoising precisely when the temporal redundancy analysis predicts the output has stabilized.

F Additional Related Work

F.1 Attention-Based Acceleration in Vision and Language Models

Attention-based acceleration has been explored in autoregressive vision-language models (VLMs) and generative diffusion transformers (DiTs), typically following an analyze-then-accelerate paradigm:

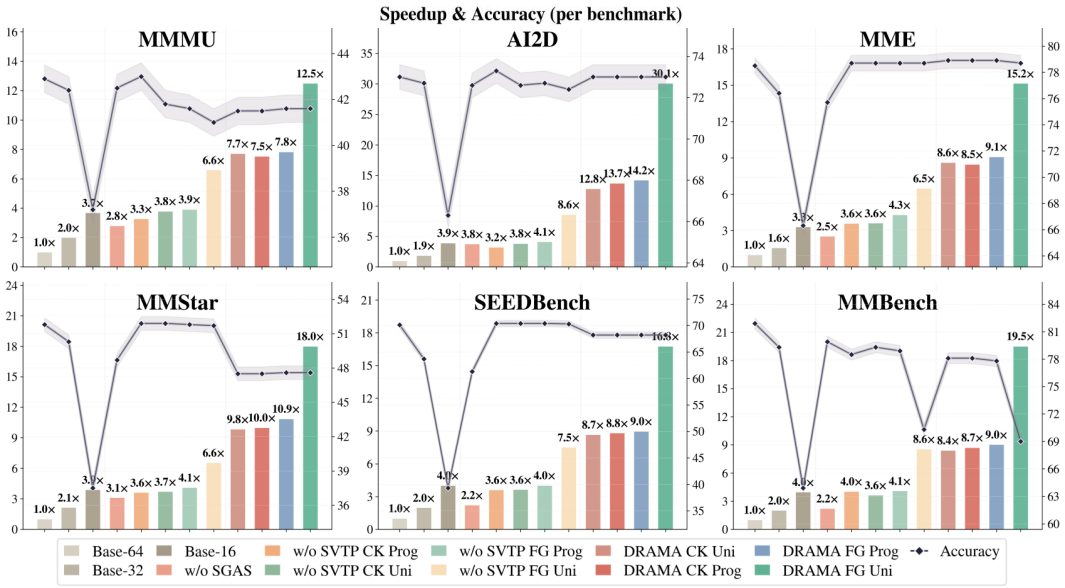


Figure 12: DRAMA achieves an average $18.67\times$ inference speedup over fixed-budget baselines across six benchmarks (MMMU [54], AI2D [22], MME [13], MM-Star [8], SEED-Bench [26], and MMBench [33]) while maintaining competitive accuracy.

first identifying structured attention patterns, then designing targeted strategies to exploit them. In autoregressive VLMs, methods such as FlexPrefill [25] and MInference [20, 30] observe that attention is sparse and concentrated on a few key tokens, motivating attention sparsification and token pruning. In generative DiTs, DiTFastAttn [53, 55] identifies temporal similarity in attention maps across denoising steps and caches redundant computations accordingly. Despite the success of this paradigm in both domains, no systematic redundancy analysis-grounded acceleration or attention-based acceleration framework exists for multimodal dLLMs. Our work addresses this gap by dissecting attention redundancy across two orthogonal axes, spatial and temporal, and deriving dimension-specific acceleration strategies from the empirical findings.

G Preliminary

G.1 Diffusion Large Language Models.

Diffusion large language models (dLLMs) [38, 3] define a forward noising process and learn a reverse denoising process for generation. Multimodal dLLMs [52, 28, 51] extend this framework by conditioning on visual input: a frozen vision encoder maps the input image to N_v visual tokens $\mathbf{v}$, which, together with N_c instruction tokens $\mathbf{c}$, form a fixed prefix exempt from masking. The forward process acts exclusively on the N_g generation tokens $\mathbf{x}_0$, independently masking each with probability $t \in [0, 1]$:

$$q_{t|0}(\mathbf{x}_t \mid \mathbf{x}_0) = \prod_{i=1}^{N_g} \mathrm{Cat}\Big(x_t^i;\ (1-t)\,\delta_{x_0^i} + t\,\delta_{[\mathrm{MASK}]}\Big)\,. \tag{14}$$

The model $p_\theta(x_0^i \mid \mathbf{x}_t, \mathbf{v}, \mathbf{c})$ is trained to predict the clean token at each masked position. At inference, generation proceeds from a fully masked sequence through T discrete steps; at each step, the model predicts all masked positions and selects $\sim N_g/T$ to unmask based on prediction confidence, with previously revealed tokens preserved. Each step processes the concatenated sequence $[\mathbf{v}; \mathbf{c}; \mathbf{x}_t]$ of length $N = N_v + N_c + N_g$ through L layers of bidirectional self-attention with H heads, yielding attention tensors $\{\mathbf{A}_l^{(t)} \in \mathbb{R}^{H\times N\times N}\}_{l=1}^{L}$. The total inference cost thus scales as $\mathcal{O}(T \cdot L \cdot N^2)$, where N_v dominates the sequence length. Since $\mathbf{v}$ is fixed across all T steps, its quadratic attention cost is replicated at every step without diminishing return.

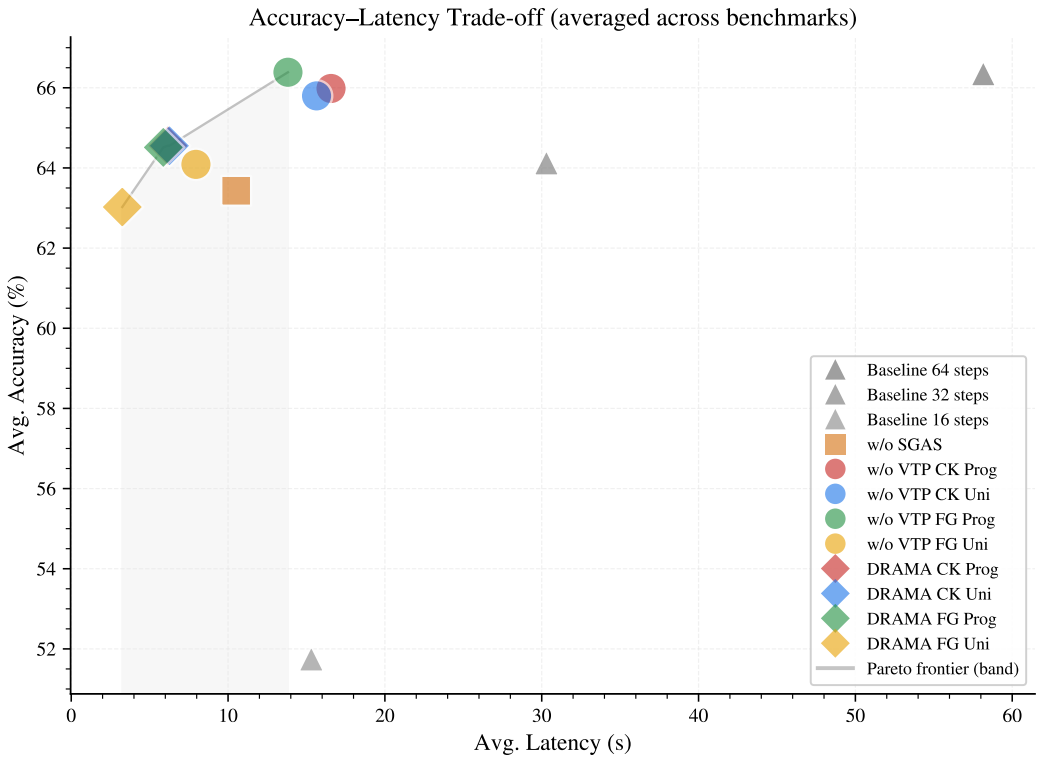


Figure 13: **Averaged accuracy–latency Pareto frontier** across all benchmarks. Each point represents a method's mean accuracy and mean latency. DRAMA variants occupy the Pareto front, achieving superior accuracy at substantially lower latency than fixed-budget baselines.

G.2 Multimodal Diffusion Language Models.

Multimodal dLLMs [52, 28, 51] extend this framework by conditioning generation on a visual input. A frozen vision encoder maps an input image to N_v visual tokens $\mathbf{v} = (v^1, \ldots, v^{N_v})$; together with N_c instruction tokens $\mathbf{c} = (c^1, \ldots, c^{N_c})$, these form a fixed conditioning prefix that is never subject to masking. The forward and reverse processes act exclusively on the N_g generation tokens $\mathbf{x}_t$, with the model's predictive distribution conditioned on the full context: $p_\theta(x_0^i \mid \mathbf{x}_t, \mathbf{v}, \mathbf{c})$. At each denoising step, the model processes the concatenated sequence $[\mathbf{v}; \mathbf{c}; \mathbf{x}_t]$ of length $N = N_v + N_c + N_g$ through L layers of bidirectional self-attention with H heads, yielding attention tensors $\{\mathbf{A}_l^{(t)} \in \mathbb{R}^{H \times N \times N}\}_{l=1}^{L}$. The total inference cost scales as $\mathcal{O}(T \cdot L \cdot N^2)$, where N_v constitutes the dominant factor in sequence length. As $\mathbf{v}$ is fixed across all T steps, its quadratic attention cost is replicated at every step without diminishing return, leading to a structural inefficiency that does not arise in text-only dLLMs.

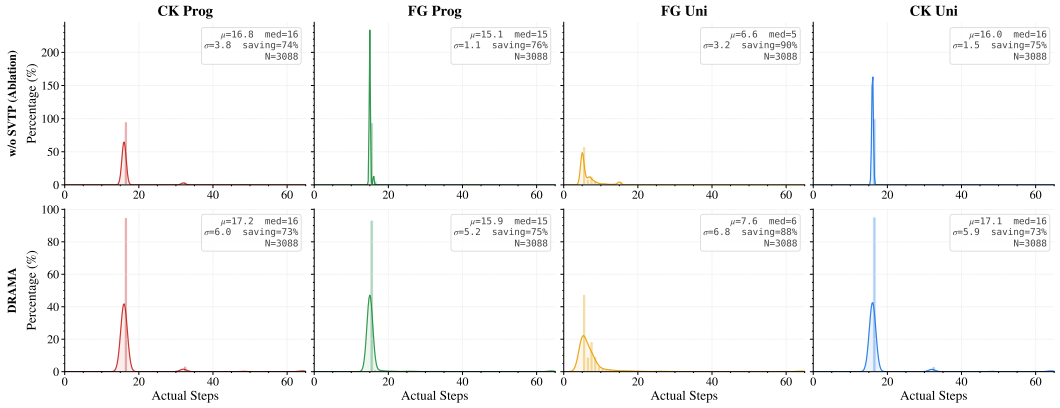


Figure 14: **Step distribution – AI2D.** FG Uni achieves 85%/79% step saving (w/o SVTP / DRAMA) owing to the concentrated spatial attention on diagram elements.

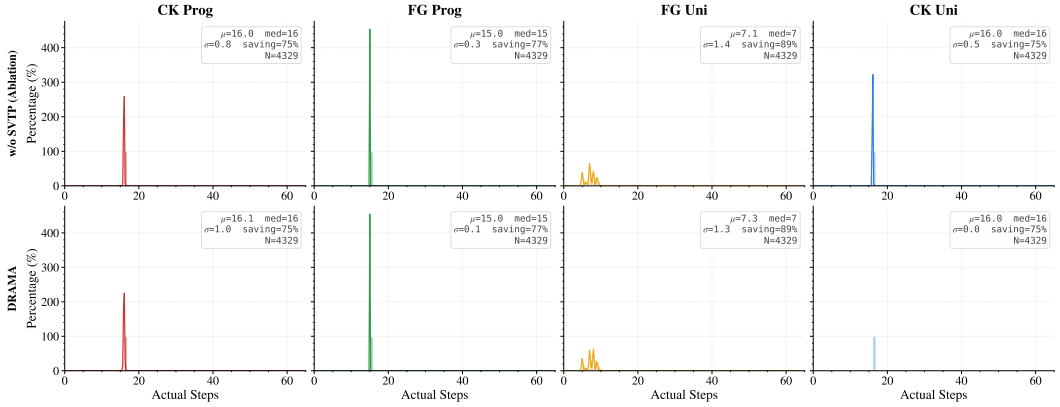


Figure 15: **Step distribution – MMBench.** The diverse visual content of MMBench leads to a slightly broader stopping distribution compared to simpler benchmarks.

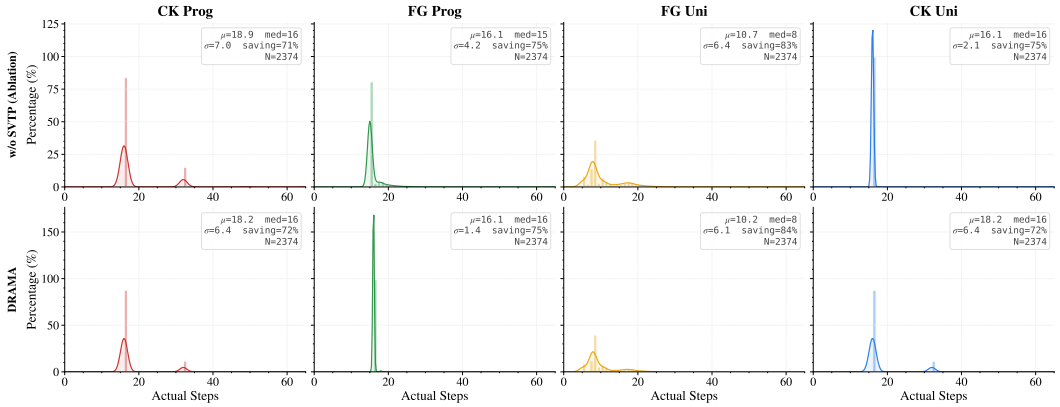


Figure 16: **Step distribution – MME.** Perceptual yes/no tasks converge rapidly, with most samples stopping within the first 15–20 steps.

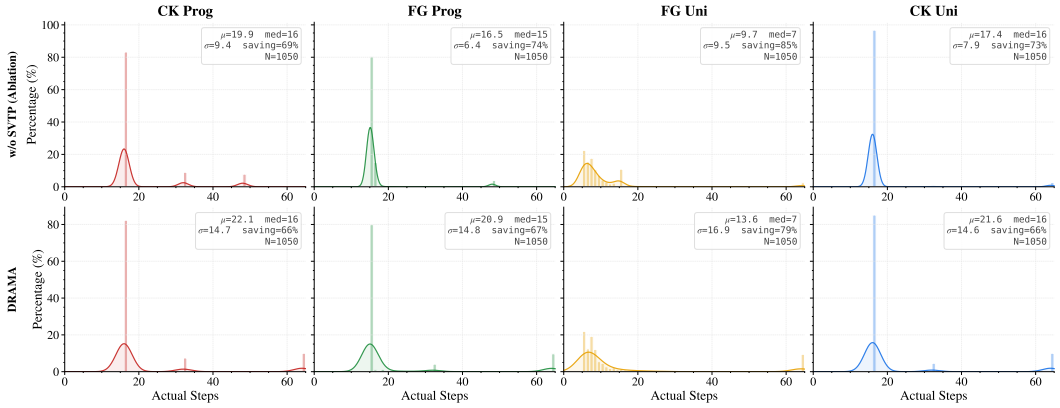


Figure 17: **Step distribution – MMMU.** Multi-discipline knowledge questions exhibit moderate step savings, with progressive scheduling providing a useful accuracy safeguard.

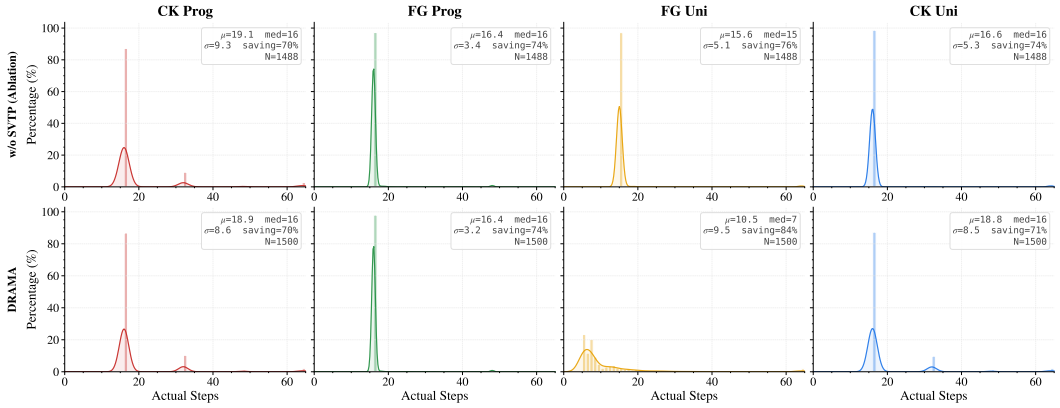


Figure 18: **Step distribution – MM-Star.** Knowledge-intensive queries show stopping patterns similar to MMMU, confirming that task difficulty correlates with stopping behaviour.

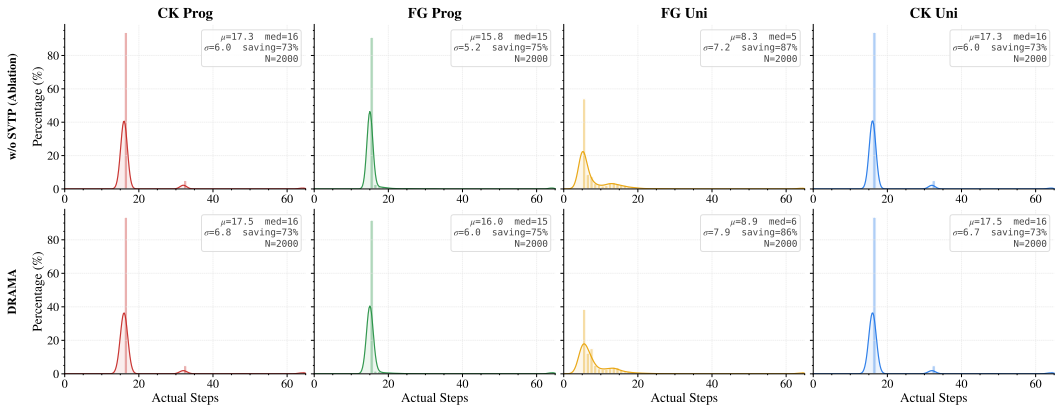


Figure 19: **Step distribution – SEED-Bench.** The diverse scene understanding tasks in SEED-Bench produce a broad stopping distribution, yet SGAS still achieves substantial step savings.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction state DRAMA's contributions (spatial/temporal redundancy analysis, SVTP, SGAS, and the reported speedup); these are substantiated by the analyses in Sections 3.1 and 3.2 and the empirical results in Table 1.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Limitations are discussed in Appendix A.1: DRAMA targets inference-time efficiency on top of a frozen backbone and does not extend the base model's intrinsic reasoning capabilities.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: The output-stationarity argument underlying SGAS (Section 3.3.2) is derived from the empirical JSD plateau ($\bar{J}_l(t) \to 0$, Section 3.2) via Pinsker's inequality; the assumption, decomposition and bound are stated in the main text, with the formal derivation expanded in the appendix.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Our Experimental Details section specifies the backbone, hardware, decoding strategy, and denoising budgets; Appendix B provides the prompt template (Table 3), full SGAS thresholds (Table 5), and the stopping algorithm (Algorithm 1).

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: All six evaluation benchmarks are publicly available and properly cited; our code, configurations, and reproduction scripts will be released upon acceptance.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Section 4 reports the backbone, hardware, decoding setup, and denoising budgets; data splits are listed in Appendix B.1 (Table 2) and per-variant SGAS thresholds in Appendix B.3. DRAMA is training-free, so no optimizer is involved.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: We report single-run results because (i) decoding is deterministic, (ii) all six benchmarks are evaluated on the full set or a fixed sampled subset (SEED-Bench 2k), so there is no run-to-run variance from data sampling. This is consistent with prior training-free dLLM acceleration work.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Section 4 specifies the compute setup (single NVIDIA RTX PRO 6000, 96 GB, bfloat16, batch size 1) and reports total compute: 747.99 GPU-hours for the full study (including baselines, ablations, and preliminary runs) and 82.18 GPU-hours for the four reported DRAMA variants. Per-sample latency for each configuration on every benchmark is reported in Table 1.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have reviewed the NeurIPS Code of Ethics; the work uses publicly released models and benchmarks, involves no human subjects, and raises no privacy or fairness concerns beyond those already inherent to the base model.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.

- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Positive impact (lowered latency and accessibility of multimodal dLLM inference) is discussed throughout the paper; potential negative societal impact (lowered barrier to large-scale generation of misleading multimodal outputs) is discussed in Appendix A.2.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: We do not release any new model, dataset, or pretrained weights. DRAMA is a training-free inference-time wrapper applied to existing public models and therefore introduces no misuse risk beyond that of the base model.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: All used assets (e.g., code, data, models) are publicly released and properly cited: LLaDA-V [52], AI2D [22], MMBench [33], MME [13], MMMU [54], MM-Star [8], and SEED-Bench [26].

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [N/A]

Justification: We do not introduce any new dataset, model, or pretrained checkpoint; the contribution is an inference-time acceleration framework applied to existing public models.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: The work involves no crowdsourcing or human-subjects research; all evaluation is conducted on publicly available benchmarks.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.

- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: No human-subjects research is conducted, so IRB approval does not apply.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs are not used as a core methodological component of this work; any usage is limited to writing and editing assistance, which the policy does not require to be declared.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.