SaToMe: Salience-Driven Token Merging for Omni-Capable Diffusion MLLMs

Sixun Liu[1,*], Pengfei Guo[1,*], Jinghao Zhang[2],
Tingyu Chen[3], Xiaofeng Zhang[4,†]
[1]North China Electric Power University; [2]Shandong University
[3]Donghua University; [4]Soochow University
[*]Equal contribution; [†]Corresponding author
sixun.liu@ncepu.edu.cn; SemiZxf@163.com
Code: https://github.com/sixun-liu/SaToMe

Abstract

Omni-capable Diffusion multimodal large language models (Diffusion MLLMs) unify text-to-image (T2I) generation and image-to-text (I2T) understanding within one backbone, but at substantial cost on both fronts: visual tokens far outnumber surrounding text, and for T2I this per-step cost compounds across many denoising steps. Token merging has been studied separately in adjacent fields, yet training-free acceleration spanning T2I+I2T in one omni-capable Diffusion MLLM remains limited. We present SaToMe, a training-free token-merging method built around a local-deviation operator that we interpret as measuring *local representational uniqueness*. The denoising trajectory exposes two natural inputs to this operator: the state h and the update direction Δh. Their spreads behave very differently along the trajectory—state-salience spread is dynamic and identifies the active merging window, while update-salience masks are window-stable and serve as a single-step protection signal for cell-level structural content. T2I uses both signals; I2T, with no temporal axis, instantiates the same uniqueness criterion on the state alone, producing a single-pass additive merge. The same I2T form transfers directly to a separate I2T-only Diffusion MLLM with a different visual representation, supporting its applicability across these visual representations. SaToMe attains +20.14% T2I wall-clock speedup on LLaDA-o at 67.5% visual-token retention while largely preserving reported generation quality, +35.5% I2T wall-clock speedup on LLaDA-o, and +27.5% I2T wall-clock speedup on LaViDa.

1 Introduction

Diffusion multimodal large language models (Diffusion MLLMs) process visual and textual information through non-autoregressive iterative denoising, and have recently shown strong potential in multimodal understanding and generation (Nie et al., 2025; Li et al., 2025c; You et al., 2026). Omni-capable Diffusion MLLMs support multiple tasks within a shared backbone, including text-to-image (T2I) generation and image-to-text (I2T) understanding, pushing multimodal unified modeling toward a new generation of architectures (You et al., 2026; Yang et al., 2025). However, their inference cost grows with both sequence length N and denoising steps T: T2I generation repeatedly processes a full visual latent sequence at every denoising step (e.g., $N = 4096$ latent tokens and $T = 50$ steps in LLaDA-o), while I2T understanding must process thousands of visual tokens produced by a vision encoder. The efficiency bottleneck is therefore substantial in both capability paths, and is particularly pronounced in T2I because of its multi-step trajectory.

Token merging is a mature training-free paradigm in ViTs and image-diffusion backbones (Bolya et al., 2023; Bolya and Hoffman, 2023; Lu et al., 2025), but methods such as ToMeSD (Bolya and Hoffman, 2023) and ToMA (Lu et al., 2025) rely on U-Net or diffusion-transformer architectural conditions that have no direct counterpart in the uniform bidirectional transformers of Diffusion MLLMs. Orthogonal acceleration efforts for diffusion language models modify caching or decoding schedules (Hu et al., 2025; Liu et al., 2025; Wang et al., 2026; Zuo et al., 2026) rather than the visual-token computation within each step. Training-free visual-token merging for omni-capable Diffusion MLLMs therefore remains an underexplored setting.

We observe that although T2I generation and I2T understanding have substantially different inference structures, they share a core decision problem: identifying which visual positions are redundant and which carry critical information. T2I generation refines a 2D latent grid over a multi-step denoising trajectory, whereas I2T understanding processes a 1D visual token sequence in a single

Findings of the Association for Computational Linguistics: EMNLP 2026, pages 41656–41672
October 24-29, 2026 ©2026 Association for Computational Linguistics

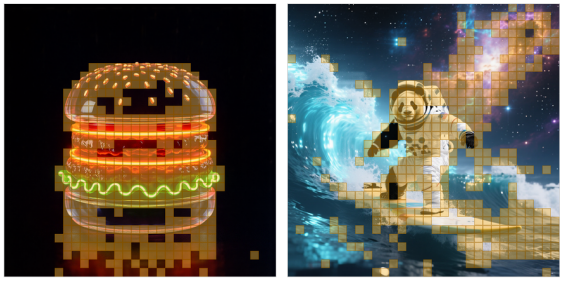


Figure 1: Salience-derived protected cells (outlined) consistently align with object boundaries and high-frequency regions across diverse T2I prompts. The same salience intuition is reused at a cell-local scope in §3.3 for protection and reconstruction-aware merging.

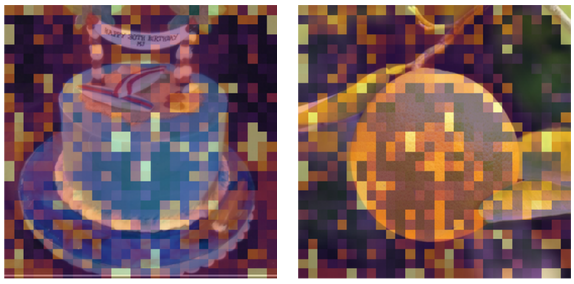


Figure 2: Salience scores visualized over representative I2T input images. The same primitive highlights object-bearing foreground regions over locally homogeneous background, providing the basis for the direct token-scope application in §3.4.

forward pass. Based on this observation, we define a local-deviation operator that we interpret as a proxy for *local representational uniqueness*: positions whose representations align with their neighborhood are locally redundant, while positions departing from local consensus carry information not captured by nearby context.

Denoising in Diffusion MLLMs naturally exposes two inputs to this operator: the state $h^{(t)}$ and the update direction $\Delta h^{(t)} = h^{(t)} - h^{(t+1)}$. We find that the two behave very differently along the trajectory. State-salience has a spread $\sigma(t)$ that profiles *when* the operator is in a discriminative regime—noise-dominated and post-convergence steps yield uniform scores—and so identifies the active merging window. Update-salience, by contrast, rapidly stabilizes after the noise-dominated phase and remains close to its commit-step mask across the active window (IoU ≥ 0.78 on a 50-prompt subset), which justifies committing a protected set once and freezing it throughout. T2I uses both signals; I2T, with no temporal axis, instantiates the same uniqueness criterion on the state alone, where high-uniqueness tokens are kept and locally redundant tokens are merged. The asymmetry between the two task instantiations is therefore one of available observations, not of method preference.

Visually, the operator's protected regions trace object boundaries across diverse T2I content (Figure 1) and highlight object-bearing regions over background on I2T inputs (Figure 2). For T2I, the pixel-level VAE decoder requires preserving cell-level structure; the operator is applied at the trajectory level for window selection and at a cell-local scope on Δh for protection and reconstruction-aware merging (Section 3.3). For I2T, the operator on h applies directly without cell-local extensions (Section 3.4).

We evaluate the full pipeline on LLaDA-o, an omni-capable backbone supporting both T2I and I2T. Because the I2T instantiation requires no specialization, we additionally apply it to LaViDa, a separate I2T-only backbone with a different visual representation paradigm, to test whether the salience primitive transfers beyond backbone-specific structure. On LLaDA-o, SaToMe achieves a +20.14% wall-clock speedup for T2I generation on 1K COCO prompts at 67.5% visual-token retention while largely preserving reported generation quality; it also achieves a +35.5% wall-clock I2T speedup at 50% retention, retaining 97.98% of the baseline four-task average. On LaViDa, the I2T instantiation reaches a +27.5% GQA-lite wall-clock speedup while retaining 99.46% of the baseline four-task average at the same 50% retention. The complete T2I sweep and matched-retention D3ToM-t comparison are reported in Table 1 and Appendix B, respectively. The method is deployed without retraining in either case.

Our contributions are summarized as follows:

- We introduce a local-deviation operator that we interpret as a proxy for local representational uniqueness, applied to two natural inputs in Diffusion MLLMs (state h and update direction Δh), with task-side instantiations differing only in the available observation budget.
- We quantify a diffusion-MLLM–specific empirical property: $s(\Delta h)$ protection masks are window-stable along the denoising trajectory while $s(h)$ masks drift; this property justifies single-step commit + freeze and is the source of the T2I instantiation's quality retention.

- We show training-free deployment of the operator across the LLaDA-o and LaViDa paradigms, including measured wall-clock speedup spanning T2I generation and I2T understanding on an omni-capable Diffusion MLLM.

2 Related Work

Token merging and visual-token reduction. Token merging was established by Bolya et al. (2023) for ViT classification via Bipartite Soft Matching, and adapted by ToMeSD (Bolya and Hoffman, 2023) to Stable Diffusion U-Nets through random 2×2 partitioning and by attention-aware ToMA (Lu et al., 2025) to Diffusion Transformers; other extensions include layer-adaptive variants (Kim et al., 2024) and adaptive U-Net windows (Lee et al., 2025). These methods rely on architecture-specific affordances—U-Net skip connections, DiT multi-resolution structures, position-aware encodings—to absorb merging errors. Adjacent vision-token pruning approaches (Rao et al., 2021; Yin et al., 2022; Liang et al., 2022; Marin et al., 2023) share the same architectural-coupling tendency. Such affordances have no direct counterpart in the uniform bidirectional transformers of omni-capable Diffusion MLLMs.

Diffusion-LM and LLM acceleration. Two routes exist for diffusion language models like LLaDA (Nie et al., 2025) and Dream (Ye et al., 2025), which build on discrete-diffusion language modeling (Austin et al., 2021; Lou et al., 2024; Sahoo et al., 2024): KV-cache adaptations (Hu et al., 2025; Liu et al., 2025; Ma et al., 2025) reduce per-step recomputation, while mask-scheduling methods change unmasking order—Window-Diffusion (Zuo et al., 2026) targets text-only generation, whereas LADR (Wang et al., 2026) exploits spatial locality to prioritize recovery at the generation frontier in discrete-diffusion T2I. LADR reduces decoding work at the schedule level, whereas SaToMe reduces per-step visual-token computation through token merging. Broader LLM acceleration via attention-sink streaming (Xiao et al., 2024), heavy-hitter eviction (Zhang et al., 2023), IO-aware kernels (Dao et al., 2022), and orthogonal diffusion-side directions—faster ODE solvers (Lu et al., 2022a), distillation (Salimans and Ho, 2022), and consistency-model parameterizations (Song et al., 2023)—target attention or sampling-step count, not visual-token count.

Omni-capable Diffusion MLLMs. Building on LLaDA, LaViDa (Li et al., 2025c) and LLaDA-V (You et al., 2025) add vision encoders for I2T, while LLaDA-o (You et al., 2026) and MMaDA (Yang et al., 2025) unify T2I+I2T in a single backbone; concurrent autoregressive unified models include Show-o (Xie et al., 2025) and Janus (Wu et al., 2024). Visual-token compression for autoregressive MLLMs (Alayrac et al., 2022; Liu et al., 2023, 2024; Bai et al., 2023; Chen et al., 2024b,a; Xing et al., 2024; Yang et al., 2024) depends on causal attention and does not directly transfer to bidirectional denoising. The most directly comparable work is D3ToM (Chang et al., 2026), which dynamically merges visual tokens in LaViDa via a decider-guided importance map for I2T only; Sparse-LaViDa (Li et al., 2025b) accelerates T2I+I2T on a LaViDa-O backbone (Li et al., 2025a) but requires additional training. Our work explores a training-free direction spanning both T2I and I2T in omni-capable Diffusion MLLMs.

3 Method

3.1 Preliminaries

Diffusion-based generation evolved from pixel-space modeling (Ho et al., 2020; Song et al., 2021) to latent-space frameworks (Rombach et al., 2022) and transformer architectures (Peebles and Xie, 2023; Podell et al., 2024); Diffusion MLLMs extend this lineage to unified multimodal sequences. A Diffusion MLLM processes a sequence of length N over T denoising steps in each forward pass. Inference cost therefore scales with $N \times T$ for both T2I generation and I2T understanding.

Let $\mathcal{V}$ denote the token vocabulary, $[\text{MASK}] \in \mathcal{V}$ the mask token, and C the multimodal condition. A denoising trajectory $X^{(T)}, X^{(T-1)}, \ldots, X^{(0)}$ is initialized as

$$X^{(T)} = ([\text{MASK}], \ldots, [\text{MASK}]). \tag{1}$$

At each step, the backbone p_θ predicts a conditional distribution over the masked positions,

$$p_\theta\Big(X \mid X^{(t)}, C\Big), \tag{2}$$

after which a transition operator $\mathcal{S}$ produces the next state:

$$X^{(t-1)} = \mathcal{S}\Big(X^{(t)}, p_\theta(\cdot \mid X^{(t)}, C), t\Big). \tag{3}$$

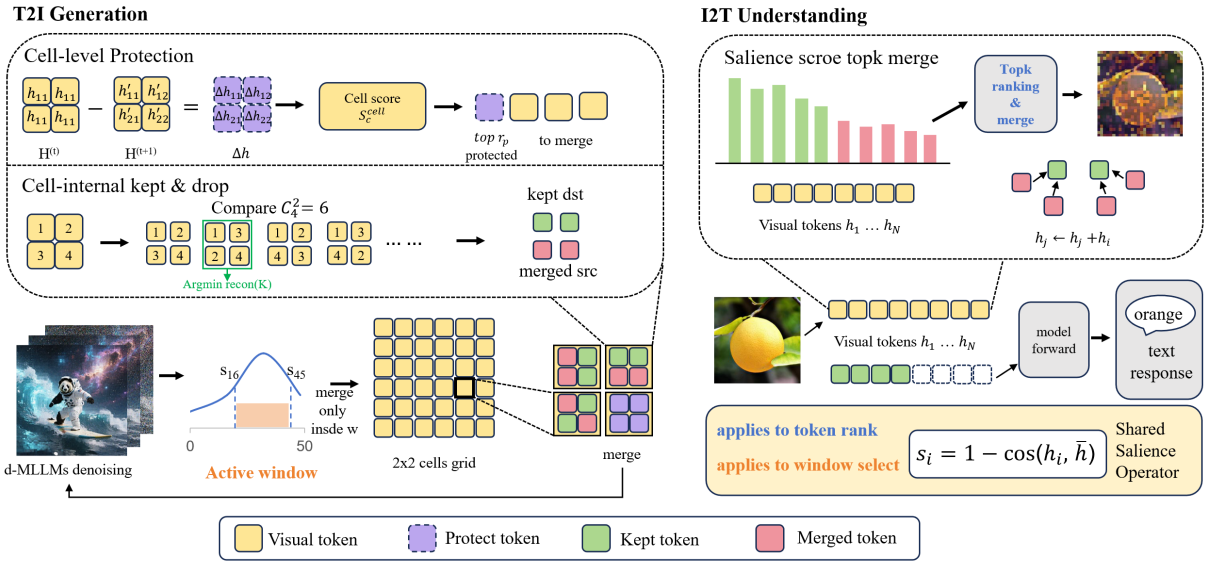


Figure 3: Overview of the SaToMe framework. A shared local-deviation operator—interpreted as local representational uniqueness—is applied to two natural inputs of the denoising forward pass (h and Δh) and instantiated as a windowed cell-level merge–restore pipeline for T2I generation, and as a direct token-level additive merge for I2T understanding.

The trajectory terminates at $X^{(0)}$; smaller t corresponds to states closer to the final output. Each forward step computes continuous hidden states $H^{(t)} \in \mathbb{R}^{N \times d}$, on which we define the local-deviation operator (Section 3.2) and instantiate task-specific merging (Sections 3.3 and 3.4; Figure 3).

3.2 Salience as Local Representational Uniqueness

Operator. Given hidden states $H = [h_1, \dots, h_N] \in \mathbb{R}^{N \times d}$ at any forward pass (timestep superscript omitted when unambiguous), we aggregate the neighborhood of position i with kernel size k and L_2 normalization (AvgPool_k for 1D sequences, AvgPool2D_k for 2D grids):

$$\bar{h}_i = \text{Norm}\left(\frac{1}{|\mathcal{N}_k(i)|} \sum_{j \in \mathcal{N}_k(i)} h_j \right), \quad (4)$$

and define a local-deviation operator

$$s_i = 1 - \cos(h_i, \bar{h}_i). \quad (5)$$

We view s_i as a proxy for *local representational uniqueness*: tokens aligned with their neighborhood are locally redundant—neighbors carry similar information—whereas tokens departing from local consensus carry information not captured by nearby context. This is consistent with local-contrast notions of structural salience in vision representations.

Trajectory-level discriminability of $s(h)$. At each denoising step t, the operator can be applied to either of two natural inputs: the state $h^{(t)}$ or the update direction $\Delta h^{(t)} = h^{(t)} - h^{(t+1)}$. The two behave very differently along the trajectory. The spread of state-salience scores,

$$\sigma(t) = q_{0.95}\left(\{s_i^{(t)}\}_{i=1}^N\right) - q_{0.05}\left(\{s_i^{(t)}\}_{i=1}^N\right), \quad (6)$$

measures whether the operator is in a discriminative regime: large $\sigma(t)$ separates locally unique positions from redundant ones; small $\sigma(t)$ indicates a uniform regime—noise-dominated or post-convergence—where merge decisions are unreliable. We use $\sigma(t)$ to localize the active interval $\mathcal{W}$ where the operator itself reports informativeness (Figure 4); window selection is therefore a self-consistency property of the operator, not an external hyperparameter.

Window-stability of $s(\Delta h)$. Applied to $\Delta h^{(t)}$, the operator yields a substantially more stable signal: slowly-varying noise cancels via subtraction. The protected mask derived from $s(\Delta h)$ retains mean IoU 0.86 (min 0.78) across the active window against the commit-step mask, while the corresponding $s(h)$ mask drifts to IoU 0.29 by late steps

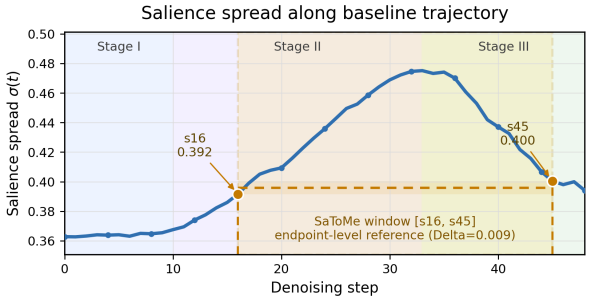


Figure 4: Salience-spread profile $\sigma(t)$ along the denoising trajectory. Three regimes are visible: Stage I plateau ($\sigma{\approx}0.36$, noise-dominated), Stage II ramp to peak $\sigma{\approx}0.475$ at $s33$ (structure emerging), Stage III decay (refinement). The active window $\mathcal{W} = [s16, s45]$ is the contiguous interval where $\sigma(t)$ exceeds the early plateau; its endpoints are read directly from the curve rather than tuned.

(Appendix D). This supports committing the protected set once at the first window step and freezing it throughout (§3.3).

Task instantiations differ only in observation budget. T2I exposes both inputs: $\sigma(t)$ on h tells *when* merging is reliable, and $s(\Delta h)$ at the commit step tells *where* to protect (§3.3). I2T is single-pass with no temporal axis, so only $s(h)$ is available; the same uniqueness criterion directly ranks tokens (§3.4). The asymmetry between the two task instantiations is one of available observations, not of method preference.

3.3 Application to T2I Generation

T2I generation must produce a pixel-level image through a VAE decoder, so cell-level structural information must be preserved. We instantiate the operator first at the trajectory level (window selection via $\sigma(t)$, §3.2) and then at a cell-local scope for protection and reconstruction-aware merging within that window.

Active window. The baseline $\sigma(t)$ profile exhibits three regimes (Figure 4): a Stage I plateau (noise-dominated), a Stage II ramp (structure emerging), and a Stage III decay (refinement). We take the active window as the contiguous interval where $\sigma(t)$ sits above the early plateau,

$$\mathcal{W} = [t_{\text{start}}, t_{\text{end}}], \quad \sigma(t) > \tau_\sigma \;\text{ for }\; t \in \mathcal{W}, \quad (7)$$

yielding $\mathcal{W}{=}[s16, s45]$ on LLaDA-o T2I: the endpoints are the first and last steps where the operator is discriminative, read off the curve rather than tuned. Endpoint failure modes are in Appendix E.

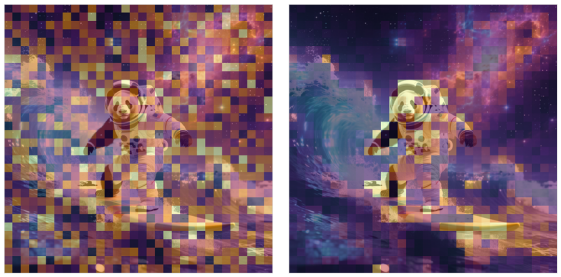


Figure 5: Comparing the operator applied to h (left) versus Δh (right) at a representative step within the active window. The Δh map concentrates more tightly on structure-bearing cells, consistent with the trajectory-stability result in Appendix D.

Cell-local protection. Within $\mathcal{W}$, we partition the 2D latent grid into non-overlapping 2×2 cells. A cell's protection score averages pairwise cosine dissimilarities of Δh within the cell:

$$S_c^{\text{cell}} = 1 - \frac{1}{6} \sum_{\substack{i,j \in c \\ i<j}} \cos(\Delta h_i, \Delta h_j). \quad (8)$$

The top-r_p cells form the protected set

$$\mathcal{P} = \text{TopK}_{r_p \cdot |\mathcal{C}|}\left(\{S_c^{\text{cell}}\}_{c \in \mathcal{C}}\right), \quad (9)$$

committed at the first step of $\mathcal{W}$ and frozen throughout; per-step refresh is unnecessary because $s(\Delta h)$ masks are window-stable (§3.2, Appendix D). The protection ratio r_p is swept in Table 1.

Cell-local merge and restoration. For each unprotected cell, two destinations $\mathcal{K}$ and two sources $\mathcal{D} = \text{cell} \setminus \mathcal{K}$ are chosen by enumerating all $\binom{4}{2}{=}6$ splits and minimizing the source-to-destination reconstruction error:

$$\mathcal{K}^* = \arg\min_{|\mathcal{K}|=2} \left(1 - \frac{1}{|\mathcal{D}|} \sum_{i \in \mathcal{D}} \max_{j \in \mathcal{K}} \cos(h_i, h_j)\right). \quad (10)$$

Sources are removed from the LLM forward pass, then restored toward their nearest destination

$$j^*(i) = \arg\max_{j \in \mathcal{K}^*} \cos(h_i, h_j), \quad (11)$$

via a soft α-blend (default $\alpha = 0.95$; ablations in Appendix E):

$$\tilde{h}_i^{\text{src}} = \alpha \cdot h_i^{\text{src}} + (1 - \alpha) \cdot h_{j^*(i)}^{\text{dst}}. \quad (12)$$

3.4 Application to I2T Understanding

For I2T there is no denoising trajectory, so only $s(h)$ is available and applies directly at token scope.

The operator does not depend on attention specialization, so we can read it at the earliest LLM-side layer (ℓ^*=1 on LaViDa, pre-layer-1 on LLaDA-o; a pilot sweep found these indistinguishable). Let $\mathcal{I}$ index the visual tokens ($N_v = |\mathcal{I}|$); the lowest top-r fraction by salience (most locally redundant) form sources $\mathcal{D}$, the rest become destinations $\mathcal{K}$:

$$\mathcal{D} = \mathrm{TopK}_{rN_v}\{-s_i \mid i \in \mathcal{I}\}, \quad \mathcal{K} = \mathcal{I} \setminus \mathcal{D}. \tag{13}$$

Each source token is additively merged into its nearest destination, then physically removed:

$$j^*(i) = \arg\max_{j \in \mathcal{K}} \cos(h_i, h_j), \quad h_{j^*(i)} \leftarrow h_{j^*(i)} + h_i. \tag{14}$$

No unmerging is needed; this simplicity makes the I2T instantiation a natural testbed for cross-backbone transfer (Section 4).

4 Experiments

4.1 Experimental Setup

Backbones We evaluate on two Diffusion MLLMs with continuous visual representations: LLaDA-o (You et al., 2026) for T2I and I2T, and LaViDa (Li et al., 2025c) for I2T only. They use different paradigms—continuous VAE latents and SigLIP embeddings (Zhai et al., 2023)—forming a cross-paradigm validation setting. Our method is training-free and runs on a single 96GB GPU; full per-task configurations are in Appendix A.

Baselines and references For T2I, no published training-free token-merging baseline exists for omni-capable Diffusion MLLMs, so the primary comparison is against the unmodified LLaDA-o backbone; as a negative control probing task-constraint–aware design, we add ToMeSD-port, a direct adaptation of ToMeSD's (Bolya and Hoffman, 2023) random 2×2 partitioning to LLaDA-o (no protection, no trajectory window, no restoration). For I2T, we compare against the concurrent training-free D3ToM (Chang et al., 2026), whose decider-attention is complementary to our salience scoring.

Benchmarks and protocol T2I is evaluated on 1K COCO prompts (Lin et al., 2014) with FID (Heusel et al., 2017), IS (Salimans et al., 2016), and GenEval (Ghosh et al., 2023), supplemented by qualitative comparisons (Figure 6). I2T is evaluated on GQA-lite (Hudson and Manning, 2019), ScienceQA (Lu et al., 2022b), AI2D (Kembhavi et al., 2016), and MMMU (Yue et al., 2024). T2I uses default $r_p = 0.35$ (67.5% retention), within the moderate-retention range adopted by prior T2I token-merging work (Bolya and Hoffman, 2023; Kim et al., 2024; Lu et al., 2025; Lee et al., 2025); the complete sweep is reported in Table 1. I2T reports a 3-point retention sweep at 50%, 25%, and 10%, matching the aggressive range used by Chang et al. (2026).

r_p	Retain ρ	FID ↓	IS ↑	GenEval ↑	Speedup ↑
Baseline	100%	*76.63*	*24.40*	*0.790*	—
0.75	87.5%	74.9	24.25	0.786	+7.9%
0.55	77.5%	73.4	24.08	0.779	+14.1%
0.35	67.5%	72.53	23.89	0.766	+20.14%
0.15	57.5%	78.5	23.20	0.735	+26.0%
0.00	50.0%	86.3	22.13	0.690	+30.3%

Table 1: T2I quality and wall-clock speedup for SaToMe on 1K COCO across protection ratios; the baseline is unmodified LLaDA-o and the shaded row is the default setting.

4.2 Main Results

T2I generation Table 1 reports the complete LLaDA-o T2I sweep on 1K COCO. At r_p=0.35, SaToMe maintains the reported generation quality under +20.14% wall-clock speedup (Table 3); ToMA and ToMeSD transfer diagnostics are reported in Appendix F. Qualitative samples are in Figure 6.

I2T cross-paradigm transfer Table 2 reports I2T across four benchmarks at three retention points on two backbones. At 50% retention—directly comparable to D3ToM-t—SaToMe is essentially lossless ($\geq$97.9% of the 4-task baseline average on both backbones) and improves over D3ToM-t on GQA-lite and SQA. The GQA-lite advantage persists at 25%; the two methods are comparable at 10%. Wall-clock saving grows monotonically as retention drops on both backbones, with LLaDA-o consistently above LaViDa (Table 3)—analyzed in §4.3.

4.3 Efficiency Analysis

We analyze efficiency through both theoretical FLOPs and measured wall-clock. Both task instantiations reduce the per-step visual-token count—for T2I, each unprotected 2×2 cell shortens from 4 to 2 tokens while protected cells (fraction r_p) keep all 4, giving retained ratio $\rho_{\text{T2I}} = 1 - \frac{1}{2}(1 - r_p)$;

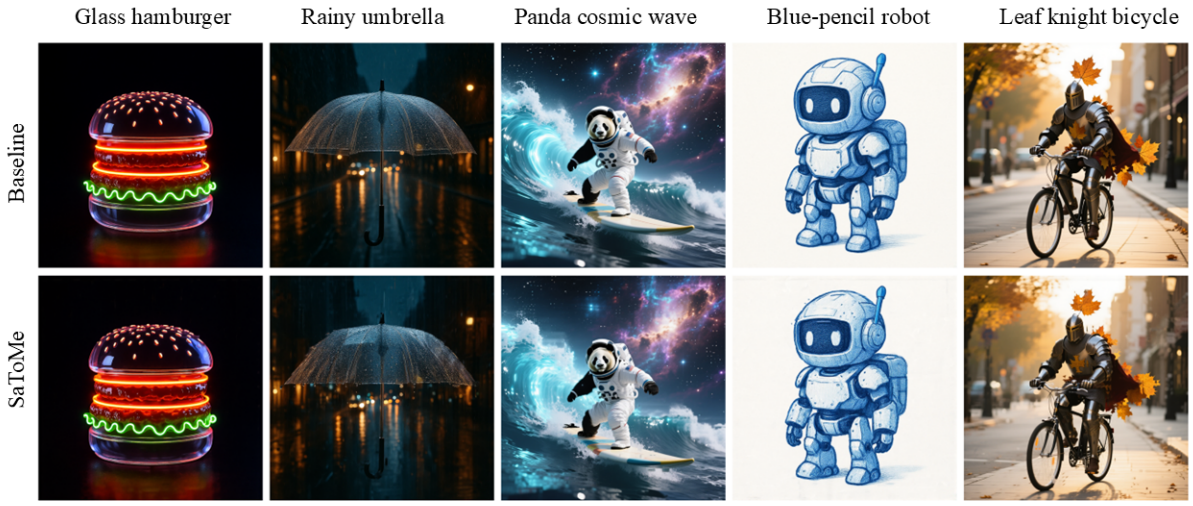


Figure 6: Qualitative T2I comparison on representative prompts, checking that token merging preserves visual structure, style, and fine details.

Method	GQA	SQA	AI2D	MMMU	Avg Retain ↑
Panel A — LaViDa (SigLIP-encoder path)					
Baseline, Retain 100%					
LaViDa baseline	*66.20*	*72.19*	*69.17*	*44.00*	*100%*
Retain 50%					
D3ToM-t	63.00	72.38	67.76	**43.00**	96.85%
Ours	**66.40**	**72.73**	**68.07**	**43.00**	**99.46%**
Retain 25%					
D3ToM-t	60.40	71.64	66.19	42.33	95.23%
Ours	**64.00**	**71.84**	**66.29**	**42.78**	**97.36%**
Retain 10%					
D3ToM-t	**62.60**	**71.44**	65.87	**42.67**	**95.31%**
Ours	**62.60**	69.26	**66.55**	41.33	95.30%
Panel B — LLaDA-o (continuous-VAE path)					
Baseline, Retain 100%					
LLaDA-o baseline	*68.20*	*71.99*	*80.60*	*44.00*	*100%*
Retain 50% / 25% / 10%					
Ours (50%)	66.80	70.25	78.59	43.78	97.98%
Ours (25%)	65.80	67.97	73.38	39.44	93.55%
Ours (10%)	62.60	64.11	66.90	38.56	88.10%

Table 2: I2T accuracy under a 3-point retention sweep on two backbones; efficiency in Table 3. D3ToM-t is evaluated under the same pipeline as our method. Blue shading identifies SaToMe rows; bold values mark the better entry within matched LaViDa blocks, with ties bolded in both rows.

for I2T, the top-r fraction is merged and physically removed—and the merging module itself adds only a cost linear in the visual-token count (salience scoring, cell-local enumeration, nearest-destination assignment), with no global pairwise matching. The realized speedup factorizes as $S_{\text{wall}} = S_{\text{FLOPs}} \cdot \eta$: a theoretical FLOPs saving S_{FLOPs} scaled by a conversion rate η that captures how much of the baseline runtime lies in the transformer compute that merging actually shortens. This separates ideal compute savings from realized wall-clock gains.

Theoretical FLOPs. With L layers, merging inserted after ℓ^*, and T denoising steps, the merged compute cost and FLOPs saving are

$$\begin{aligned} C_{\text{ours}} &= T\big[\ell^* C_{\text{layer}}(N) \\ &\quad + (L-\ell^*) C_{\text{layer}}(N_{\text{merged}})\big] + C_{\text{ovh}}, \\ S_{\text{FLOPs}} &= 1 - \frac{C_{\text{ours}}}{TL \cdot C_{\text{layer}}(N)}, \end{aligned} \tag{15}$$

with per-layer cost C_{layer}, overhead C_{ovh}, and merged length N_{merged} derived in Appendix G. T2I repeats this saving across the trajectory; I2T applies it once. The conversion rate depends on the share of runtime spent in the shortened transformer path. Non-mergeable components such as visual encoding, data handling, and framework overhead therefore make wall-clock gains backbone-dependent.

Cross-backbone efficiency. The wall-clock gap between the two backbones decomposes into an algorithmic term and a system term. On the algorithmic side, LLaDA-o carries a higher visual-token fraction and merges earlier—pre-LLM (ℓ^*=0) versus LaViDa's ℓ^*=1—so more downstream layers run on the shortened sequence and S_{FLOPs} rises from 44.5% to 49.8%. On the system side, its lower non-mergeable overhead lifts η from 61.8% to 71.3%: LaViDa's SigLIP emits a fixed 729 tokens per image, whereas LLaDA-o's NaViT-style encoder emits a larger, image-dependent count, concentrating a greater share of per-step cost in the shortenable transformer path. The two terms compound, so visual-token merging yields larger realized acceleration on visual-token-dominated backbones whose compute concentrates in the transformer path. The merge map is deterministic, so the same source-to-destination assignment applies to cached visual key/value tensors and is compatible with KV-cache decoding. Under the matched LaViDa no-KV protocol, SaToMe has lower wall-clock time than D3ToM-t at all three retention points.

Shallow-merge advantage. Because our operator reads local hidden-state deviation rather than attention scores, it can be applied at the earliest LLM-side layer (ℓ^*=1 on LaViDa, pre-layer-1 on LLaDA-o), placing essentially all transformer compute downstream of the merge point; concurrent D3ToM (Chang et al., 2026) similarly merges shallowly (ℓ^*=3). Attention-score pruning instead couples to attention maturation (FastV (Chen et al., 2024a) prunes at layer 2 on AR LLaVA), and direct ports of attention-output merging to LLaDA-o T2I fail across the axes we tested (Appendix F). Ablations are reported in Appendix E; cache-on composition is reported in Appendix C.

5 Conclusion

We presented SaToMe, a training-free token merging method for omni-capable Diffusion MLLMs built around a local-deviation operator interpreted as local representational uniqueness. The denoising trajectory exposes two operator inputs—state h and update direction Δh—whose distinct trajectory behaviors determine *when* merging is reliable and *where* structural cells must be protected; T2I uses both signals, I2T uses only $s(h)$. SaToMe delivers meaningful wall-clock acceleration on LLaDA-o T2I/I2T and LaViDa I2T while preserving generation fidelity and downstream accuracy, indicating that uniqueness-driven merging is a practical training-free route across the generation–understanding spectrum of Diffusion MLLMs.

Method (Retain)	**Wall** (s)	**Speedup** ↑	**FLOPs** ↓
T2I — LLaDA-o, 1K COCO			
Baseline (100%)	*26807.5*	—	*1.000*
Ours (67.5%)	**21407.6**	**+20.14%**	**0.790**
I2T — LaViDa, GQA-lite			
Baseline (100%)	*552.17*	—	*1.000*
D3ToM-t (50%)	411.23	+26.4%	0.609
D3ToM-t (25%)	333.26	+40.4%	0.416
D3ToM-t (10%)	279.14	+50.1%	0.302
Ours (50%)	**400.57**	**+27.5%**	**0.555**
Ours (25%)	**305.28**	**+44.7%**	**0.337**
Ours (10%)	**245.96**	**+55.5%**	**0.207**
I2T — LLaDA-o, GQA-lite			
Baseline (100%)	*113.0*	—	*1.000*
Ours (50%)	**72.9**	**+35.5%**	**0.502**
Ours (25%)	**60.3**	**+46.6%**	**0.265**
Ours (10%)	**47.1**	**+58.3%**	**0.127**

Table 3: Wall-clock and FLOPs across T2I and I2T (single 96GB GPU; same pipeline within each panel). LaViDa I2T uses a matched 500-sample GQA-lite no-KV protocol; D3ToM-t speedups are computed against its same-pipeline no-merge baseline (558.89s; within 1.2% of the LaViDa baseline). Per-row FLOPs ratios are derived in Appendix G.

Limitations

Open omni-capable Diffusion MLLMs are concentrated at the 8B scale; our evaluation covers two representative backbones (LLaDA-o, LaViDa) and does not extend to other scales or newer omni models. The T2I window $\mathcal{W}$ is derived from the $\sigma(t)$ profile (§3.2–§3.3), but the protection ratio r_p=0.35 and soft-blend α=0.95 come from pilot inspections and may need re-estimation on backbones with markedly different denoising schedules or visual encoders. T2I quality is reported on 1K COCO prompts at a single seed; broader multi-seed and compositional evaluations remain future work.

References

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooee, Marianne Monteiro, Jacob L. Menick, Sebastian Borgeaud, and 8 others. 2022. Flamingo: A visual language model for few-shot learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 23716–23736.

Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. 2021. Structured denoising diffusion models in discrete state-spaces. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. 2023. Qwen-VL: A versatile vision-language model for understanding, localization, text reading, and beyond. *arXiv preprint arXiv:2308.12966*.

Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. 2023. Token merging: Your ViT but faster. In *International Conference on Learning Representations (ICLR)*.

Daniel Bolya and Judy Hoffman. 2023. Token merging for fast stable diffusion. In *CVPR Workshop on Efficient Deep Learning for Computer Vision*.

Shuochen Chang, Xiaofeng Zhang, Qingyang Liu, and Li Niu. 2026. D^{3}ToM: Decider-guided dynamic token merging for accelerating diffusion MLLMs. In *AAAI Conference on Artificial Intelligence (AAAI)*.

Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang. 2024a. An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models. In *European Conference on Computer Vision (ECCV)*.

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, Bin Li, Ping Luo, Tong Lu, Yu Qiao, and Jifeng Dai. 2024b. InternVL: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. 2022. FlashAttention: Fast and memory-efficient exact attention with IO-awareness. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Dhruba Ghosh, Hannaneh Hajishirzi, and Ludwig Schmidt. 2023. GenEval: An object-focused framework for evaluating text-to-image alignment. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. 2017. GANs trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems (NeurIPS)*.

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

Drew A. Hudson and Christopher D. Manning. 2019. GQA: A new dataset for real-world visual reasoning and compositional question answering. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6700–6709.

Aniruddha Kembhavi, Mike Salvato, Eric Kolve, Minjoon Seo, Hannaneh Hajishirzi, and Ali Farhadi. 2016. A diagram is worth a dozen images. In *European Conference on Computer Vision (ECCV)*, pages 235–251.

Minchul Kim, Shangqian Gao, Yen-Chang Hsu, Yilin Shen, and Hongxia Jin. 2024. Token fusion: Bridging the gap between token pruning and token merging. In *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*.

Min-Jeong Lee, Hee-Dong Kim, and Seong-Whan Lee. 2025. Local representative token guided merging for text-to-image generation. *arXiv preprint arXiv:2507.12771*.

Shufan Li, Jiuxiang Gu, Kangning Liu, Zhe Lin, Zijun Wei, Aditya Grover, and Jason Kuen. 2025a. LaViDa-O: Elastic large masked diffusion models for unified multimodal understanding and generation. *arXiv preprint arXiv:2509.19244*.

Shufan Li, Jiuxiang Gu, Kangning Liu, Zhe Lin, Zijun Wei, Aditya Grover, and Jason Kuen. 2025b. SparseLaViDa: Sparse multimodal discrete diffusion language models. *arXiv preprint arXiv:2512.14008*.

Shufan Li, Siwen Zhu, Yiqi Chen, Fengqi Zhu, Xinyi Zhang, Chao You, Jun Chen, and Eric Xing. 2025c. LaViDa: Large diffusion language models with visual instruction tuning. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Youwei Liang, Chongjian Ge, Zhan Tong, Yibing Song, Jue Wang, and Pengtao Xie. 2022. Not all patches are what you need: Expediting vision transformers via token reorganizations. In *International Conference on Learning Representations (ICLR)*.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. Microsoft COCO: Common objects in context. In *European Conference on Computer Vision (ECCV)*, pages 740–755.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2024. Improved baselines with visual instruction tuning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Zhiyuan Liu, Yicun Yang, Yaojie Zhang, Junjie Chen, Chang Zou, Qingyan Wei, Shaobo Wang, Yichen Zhu, and Linfeng Zhang. 2025. dLLM-Cache: Accelerating diffusion large language models with adaptive caching. *arXiv preprint arXiv:2506.06295*.

Aaron Lou, Chenlin Meng, and Stefano Ermon. 2024. Discrete diffusion modeling by estimating the ratios of the data distribution. In *International Conference on Machine Learning (ICML)*.

Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. 2022a. DPM-Solver: A fast ODE solver for diffusion probabilistic model sampling in around 10 steps. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Pan Lu, Swaroop Mishra, Tanglin Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Øyvind Tafjord, Peter Clark, and Ashwin Kalyan. 2022b. Learn to explain: Multimodal reasoning via thought chains for science question answering. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Wenbo Lu, Shaoyi Zheng, Yuxuan Xia, and Shengjie Wang. 2025. ToMA: Token merge with attention for diffusion models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 40930–40951. PMLR.

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

Dmitrii Marin, Jen-Hao Rick Chang, Anurag Ranjan, Anish Prabhu, Mohammad Rastegari, and Oncel Tuzel. 2023. Token pooling in vision transformers for image classification. In *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*.

Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, Jun Zhou, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. 2025. Large language diffusion models. *arXiv preprint arXiv:2502.09992*.

William Peebles and Saining Xie. 2023. Scalable diffusion models with transformers. In *IEEE/CVF International Conference on Computer Vision (ICCV)*.

Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. 2024. SDXL: Improving latent diffusion models for high-resolution image synthesis. In *International Conference on Learning Representations (ICLR)*.

Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. 2021. DynamicViT: Efficient vision transformers with dynamic token sparsification. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-resolution image synthesis with latent diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Subham Sekhar Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin T. Chiu, Alexander Rush, and Volodymyr Kuleshov. 2024. Simple and effective masked diffusion language models. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. 2016. Improved techniques for training GANs. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Tim Salimans and Jonathan Ho. 2022. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations (ICLR)*.

Jiaming Song, Chenlin Meng, and Stefano Ermon. 2021. Denoising diffusion implicit models. In *International Conference on Learning Representations (ICLR)*.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. 2023. Consistency models. In *International Conference on Machine Learning (ICML)*.

Chenglin Wang, Yucheng Zhou, Shawn Chen, Tao Wang, and Kai Zhang. 2026. LADR: Locality-aware dynamic rescue for efficient text-to-image generation with diffusion large language models. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 27170–27183. Association for Computational Linguistics.

Chengyue Wu, Xiaokang Chen, Zhiyu Wu, Yiyang Ma, Xingchao Liu, Zizheng Pan, Wen Liu, Zhenda Xie, Xingkai Yu, Chong Ruan, and Ping Luo. 2024. Janus: Decoupling visual encoding for unified multimodal understanding and generation. *arXiv preprint arXiv:2410.13848*.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. 2024. Efficient streaming language models with attention sinks. In *International Conference on Learning Representations (ICLR)*.

Jinheng Xie, Weijia Mao, Zechen Bai, David Junhao Zhang, Weihao Wang, Kevin Qinghong Lin, Yuchao Gu, Zhijie Chen, Zhenheng Yang, and Mike Zheng Shou. 2025. Show-o: One single transformer to unify multimodal understanding and generation. In *International Conference on Learning Representations (ICLR)*.

Long Xing, Qidong Huang, Xiaoyi Dong, Jiajie Lu, Pan Zhang, Yuhang Zang, Yuhang Cao, Conghui He, Jiaqi Wang, Feng Wu, and Dahua Lin. 2024. PyramidDrop: Accelerating your large vision-language models via pyramid visual redundancy reduction. *arXiv preprint arXiv:2410.17247*.

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

Senqiao Yang, Yukang Chen, Zhuotao Tian, Chengyao Wang, Jingyao Li, Bei Yu, and Jiaya Jia. 2024. VisionZip: Longer is better but not necessary in vision language models. *arXiv preprint arXiv:2412.04467*.

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

Hongxu Yin, Arash Vahdat, Jose M. Alvarez, Arun Mallya, Jan Kautz, and Pavlo Molchanov. 2022. A-ViT: Adaptive tokens for efficient vision transformer. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

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

Zebin You, Xiaolu Zhang, Jun Zhou, Chongxuan Li, and Ji-Rong Wen. 2026. LLaDA-o: An effective and length-adaptive omni diffusion model. *arXiv preprint arXiv:2603.01068*.

Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, Cong Wei, Botao Yu, Ruibin Yuan, Renliang Sun, Ming Yin, Boyuan Zheng, Zhenzhu Yang, Yibo Liu, Wenhao Huang, and 3 others. 2024. MMMU: A massive multi-discipline multimodal understanding and reasoning benchmark for expert AGI. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9556–9567.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. 2023. Sigmoid loss for language image pre-training. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 11975–11986.

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark W. Barrett, Zhangyang Wang, and Beidi Chen. 2023. H2O: Heavy-hitter oracle for efficient generative inference of large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 34661–34710.

Fengrui Zuo, Zhiwei Ke, Yiming Liu, Wenqi Lou, Chao Wang, and Xuehai Zhou. 2026. Window-diffusion: Accelerating diffusion language model inference with windowed token pruning and caching. *arXiv preprint arXiv:2601.20332*.

A Implementation Details

Hardware and software All experiments run on a single NVIDIA RTX PRO 6000 (96GB) with model weights fully resident in GPU memory. Backbone weights are loaded in bfloat16 precision. T2I generation uses the LLaDA-o pipeline; I2T evaluation uses the `lmms-eval` framework with backbone-specific model wrappers.

Backbones LLaDA-o is an omni-capable Diffusion MLLM supporting both T2I generation and I2T understanding within a shared backbone, with a continuous-VAE visual representation. The diffusion backbone has $L = 32$ transformer layers, hidden dimension $d = 4096$, FFN intermediate width $m = 12288$, and 32 attention heads. For T2I, the backbone operates on a 64×64 latent grid ($N_v = 4096$ tokens) producing 1024×1024 images by default. For I2T, LLaDA-o uses a SigLIP-NaViT vision tower with dynamic patchification (stride 14, max resolution 980, min 378, max pixels 2,007,040); the visual token count therefore varies per image and is not fixed. LaViDa shares the same diffusion backbone configuration ($L = 32$, $d = 4096$, $m = 12288$) but uses a SigLIP-so400m-patch14-384 vision tower, producing 729 visual tokens (27×27 grid) per 384×384 image.

T2I generation The denoising trajectory contains $T = 50$ steps with classifier-free guidance scale 4.0. The active merging window is $\mathcal{W} = [\text{s16}, \text{s45}]$, covering 30 of the 50 steps; merging is disabled outside $\mathcal{W}$. The window endpoints were identified from the salience-spread curve (Figure 4) as the contiguous interval where $\sigma(t)$ is clearly above the early-stage plateau, and validated against the qualitative endpoint-failure modes in Figure 8. Within the window, the 64×64 latent grid is partitioned into non-overlapping 2×2 cells. The default protection ratio is $r_p = 0.35$ (67.5% retention), chosen as the operating point that balances retention and qualitative fidelity in our pilot inspection. Source positions are restored through a soft α-blend with $\alpha = 0.95$; this value was selected from the operator ablation in Appendix E Table 7, where $\alpha = 1.0$ leaves persistent source noise and $\alpha = 0.95$ was the only setting producing clean outputs across all pilot prompts. The salience score uses a $k = 11$ kernel for the AvgPool2D neighborhood aggregation, chosen as a small odd kernel that captures local context without over-smoothing; cell-level protection uses 2×2 intra-cell cosine isolation rather than AvgPool.

I2T understanding LaViDa applies merging after the first LLM layer (ℓ^*=1); LLaDA-o applies merging before the first LLM layer (ℓ^*=0, i.e., pre-LLM on the projected visual tokens). The merge ratio r matches the target visual-token retention (e.g., $r = 0.5$ at 50% retention). Decoding follows the D3ToM evaluation protocol with backbone-specific output lengths $|O|$: 16 for GQA-lite, 64 for AI2D and MMMU, 100 for ScienceQA. Generation uses `prefix_lm`=False, `block_length`= 64, and `step_ratio`= 0.5 unless otherwise specified. For the matched LaViDa GQA-lite timing sweep, we use `max_new_tokens`= 16, `block_length`= 16, `step_ratio`= 1.0, `schedule`=`shift`, and disable KV cache. The salience score uses a $k = 11$ kernel for the AvgPool1D neighborhood aggregation.

Evaluation protocol T2I evaluation uses 1K COCO captions sampled with seed 42; FID and IS are computed against the same caption set; GenEval uses its standard prompt suite. I2T evaluation uses GQA-lite, ScienceQA, AI2D, and MMMU through the `lmms-eval` framework; the matched LaViDa GQA-lite timing sweep uses the first 500 samples. Wall-clock measurements use `warmup_runs`= 1 followed by `timed_runs`= 2, with the median reported unless a task-specific timing sweep is specified. The T2I quality comparison uses paired seed 42; other results follow their task-specific protocols.

Reproducibility The implementation and configuration files are publicly available at https://github.com/sixun-liu/SaToMe.

B Matched I2T Comparison

We report matched-retention results for SaToMe and D3ToM-t on LLaDA-o I2T. The shared no-merge baseline has a four-task average of 66.20 and a wall time of 113.0 seconds.

C KV-Cache Composition

With KV caching enabled, SaToMe provides additional wall-clock acceleration relative to the corresponding cache-only baseline, without a measured accuracy change.

Retention	Method	GQA	SQA	AI2D	MMMU	Avg.	Wall (s)	Speedup
50%	SaToMe	66.80	70.25	78.59	43.78	64.86	72.9	+35.5%
50%	D3ToM-t	63.40	69.90	78.28	43.78	63.84	74.1	+34.4%
25%	SaToMe	65.80	67.97	73.38	39.44	61.65	60.3	+46.6%
25%	D3ToM-t	62.20	67.77	73.28	38.99	60.56	65.2	+42.3%
10%	SaToMe	62.60	64.11	66.90	38.56	58.04	47.1	+58.3%
10%	D3ToM-t	62.60	66.29	66.22	39.90	58.75	53.2	+52.9%

Table 4: Matched-retention D3ToM-t comparison on LLaDA-o I2T under the same evaluation pipeline.

Backbone	Visual tokens	Extra speedup	Accuracy change
LaViDa	729 fixed	+7.5%	0.0 pp
LLaDA-o	$\approx$ 1,912 avg.	+18.0%	0.0 pp

Table 5: KV-cache composition results relative to cache-only inference.

anchor t	IoU($s(\Delta h)$, ref= $s16$)	IoU($s(h)$, ref= $s16$)
$s0$	0.360 ± 0.089	0.595 ± 0.045
$s12$	$\mathbf{0.863 \pm 0.056}$	0.837 ± 0.029
$s24$	$\mathbf{0.883 \pm 0.046}$	0.659 ± 0.034
$s36$	$\mathbf{0.833 \pm 0.054}$	0.398 ± 0.031
$s48$	$\mathbf{0.710 \pm 0.080}$	0.288 ± 0.026

Table 6: Protection-mask IoU against the $s16$ reference (50 prompts). Inside the active window $[s16, s45]$, $s(\Delta h)$ stays close to the commit-step mask (mean 0.86, min 0.78); $s(h)$ drifts. Bold marks $s(\Delta h)$ anchors inside or adjacent to the window.

D Trajectory Stability of Protection Masks

We quantify how stable the protected-cell mask is across denoising steps in order to justify the single-step commit used in §3.3: the protected set is selected at the first window step $t{=}s16$ and frozen throughout. For each of 50 random prompts from the COCO/FID-1K pool, we run a baseline (no-merge) trajectory, compute $s(\Delta h)$ and $s(h)$ cell-protection scores (Equation 8) at five anchor steps $t \in \{s0, s12, s24, s36, s48\}$, form the top-$r_p$ ($r_p{=}0.35$) protected set at each anchor, and report the IoU of each anchor's mask against the $s16$ reference mask actually used by the method.

Findings. After the initial noise-dominated phase, $s(\Delta h)$ rapidly stabilizes and remains close to the commit-step mask: in $[s12, s36]$ the IoU stays at 0.83–0.88, and across the active window endpoints ($s16$ to $s45$) the mean IoU is 0.86 with minimum 0.78. By contrast, $s(h)$ is competitive only near the commit step ($s12$: 0.84) and drifts to 0.40 at $s36$ and 0.29 at $s48$, indicating that the state-salience map is bound to the instantaneous noise composition. The noise-dominated step $s0$ is an honest exception—$s(\Delta h)$ there is no more stable than $s(h)$—which is consistent with the operator-discriminability picture in §3.2: the regime before structure emerges is uniform under both inputs.

Implication. Within the active window selected by $\sigma(t)$, $s(\Delta h)$ provides a trajectory-stable protection signal, supporting the single-step commit and freeze in §3.3. The contrast with $s(h)$'s late drift further indicates that the update direction, not the instantaneous state, is the right input to the operator for the protection role; raw h remains a useful ablation but is not equivalent to Δh as a protection signal.

E Failure Taxonomy and Component Analysis

We organize the observed T2I failures around three mechanism-level axes: window selection, token protection, and source restoration. These controlled diagnostics explain the corresponding design choices; they are not estimates of dataset-level failure frequencies. We isolate the first two axes qualitatively (Figure 8 and Figure 9) and the restoration operator quantitatively (Table 7).

Window selection Failure modes (Figure 8): starting too early disrupts image structure before it forms and causes style drift; ending too late damages details and leaves residual noise. The window [s16, s45] balances these effects and matches the reliable middle of the salience-spread curve.

Protection necessity Without protection, salience-only merging removes structural cells along object contours (Figure 9, left); with Δh protection, update-direction–salient cells are preserved (right). The choice of Δh over raw h_t is illustrated in Figure 5.

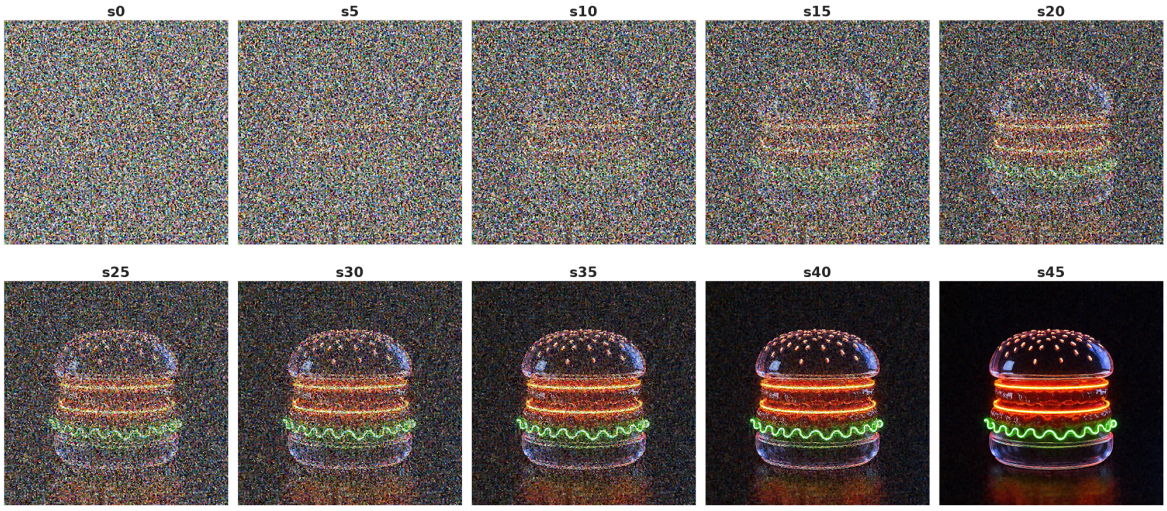


Figure 7: Denoising trajectory for a representative T2I prompt at ten anchor steps (s0–s45, step 5). The pure-noise plateau (s0–s10) yields the uniform $\sigma(t)$ regime described in §3.2; structural content begins to emerge around s15, matching the active-window onset at s16; s35–s45 contains the final refinement. Together with Table 6, this trajectory illustrates why $s(\Delta h)$ protection masks remain stable inside $[s16, s45]$ while $s(h)$ drifts with the changing state composition.

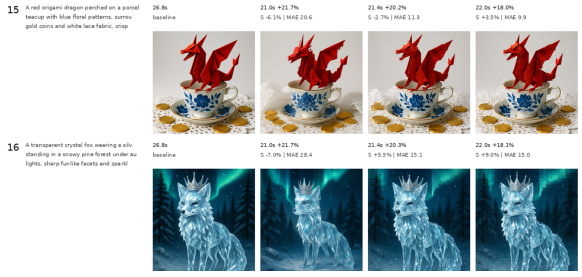


Figure 8: Window endpoint failure modes. Starting before s16 disrupts image structure before it forms and causes style drift; ending after s45 leaves visible noise and damages fine details. The window $[s16, s45]$ preserves baseline-like generation across diverse prompts.

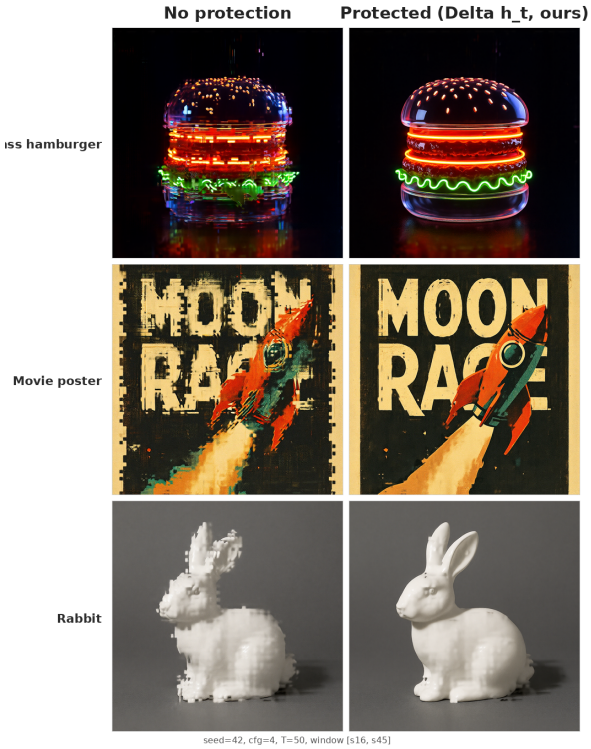


Figure 9: Protection necessity at the first merge step ($t{=}s16$), three representative prompts. *Left, no protection:* salience-only merging removes structural cells along object contours. *Right, Δh (ours):* update-direction salience preserves them. The Δh vs raw-h comparison appears in Figure 5.

Merge operator Table 7 ablates source-position restoration: hard copy ($\alpha = 1.0$) leaves persistent noise; soft blend ($\alpha = 0.95$, default) eliminates it; β-weighted source merging performs worse than prune-only; fixing the source-destination pair from the first step causes block-like artifacts. Per-step reselection with soft restoration is the only configuration that consistently avoids artifacts.

F T2I Token-Merging Transfer Diagnostics

Scope. ToMA and ToMeSD were developed for different diffusion architectures and interfaces. We therefore treat the following LLaDA-o experiments as direct-port diagnostics, rather than matched reproductions or native-backbone rankings.

ToMA. A direct ToMA (Lu et al., 2025) port covering all denoising steps and most layers

Variant	Outcome
Hard copy ($\alpha = 1.0$)	5/5 noisy; source positions retain initial noise.
Soft blend ($\alpha = 0.95$)	**5/5 clean; default setting.**
Source merge ($\beta = 0.5$)	Worse than prune-only; injection is harmful.
Fixed first-step pair	5/5 blocky; per-step reselection is needed.

Table 7: Source-position restoration operator ablation. Each variant generates the same 5 prompts; the soft-blend $\alpha = 0.95$ default is the only configuration that consistently avoids artifacts.

Method	Speedup	FID ↓	IS ↑	GenEval ↑
LLaDA-o baseline	—	76.63	24.40	0.790
Direct ToMA port	+13.0%	91.2	21.5	0.640

Table 8: Direct ToMA-port diagnostic on LLaDA-o T2I.

with cross-step reuse achieves $+13.0\%$ wall-clock speedup, but changes the LLaDA-o baseline from FID/IS/GenEval $76.63/24.40/0.790$ to $91.2/21.5/0.640$.

Setup. ToMeSD (Bolya and Hoffman, 2023) merges visual tokens through a random 2×2 partition with attention-output unmerge, relying on U-Net skip connections to absorb the resulting reconstruction error. LLaDA-o T2I has no such affordance, so any direct port must choose simultaneously a layer-depth scope, a temporal scope, and whether to extend merging through the FFN. We sweep four configurations on 5 standard prompts (seed 42, T=50, CFG= 4), each isolating one expansion axis relative to a conservative shallow-window reference. Speedups are wall-clock against the unmodified LLaDA-o baseline (≈ 26.8s/prompt).

Failure axes. Table 9 summarizes the four direct-port variants and Figure 10 shows the qualitative outcome on five representative prompts. The shallow-window variant yields only $+4.4\%$ wall-clock while introducing visible colored speckle. Expanding to all 32 attention layers reaches $+9.5\%$ but produces catastrophic speckle that renders several prompts barely readable. Expanding temporal coverage to all 50 denoising steps yields $+7.3\%$ with worsening drift residue. Extending the design to also merge through the FFN at layers $0-15$ reaches $+13.1\%$ but the outputs become unusable. No single configuration achieves both meaningful speedup and acceptable visual quality; SaToMe (second row of Figure 10) by contrast matches the baseline visually across all prompts at $+20.14\%$ speedup.

Implication for SaToMe. The three failure axes match three design choices we ultimately settled on:

- *Trajectory window.* The full-trajectory variant shows that applying merging at noise-dominated early steps degrades quality without compensating speedup—consistent with $\sigma(t)$ falling below the discriminative regime there (§3.2).
- *Cell-local protection.* The all-attention variant shows that expanding random-partition merging to deeper attention layers removes structural content along object contours; the cell-local Δh protection in §3.3 reserves these cells across the entire active window.
- *Physical removal vs. attention-output unmerge.* The attention+FFN variant shows that extending attention-output merging through the FFN amplifies unmerge error into visible noise; SaToMe instead physically removes source tokens, so subsequent attention and FFN both operate on the shorter sequence without channel-specific unmerge.

Design exploration. Before settling on uniform pre-layer token removal, we considered separate attention-only and attention+FFN variants and a sweep over merge-layer depth. The four variants represent representative points of that exploration. The final design is the simplest configuration we found that consistently retains baseline-like quality at meaningful speedup; more elaborate per-channel designs did not improve on it on this backbone.

G Detailed FLOPs Derivation

This appendix derives the FLOPs cost expressions used in Section 4.3.

Sequence decomposition Let the per-step sequence length N be decomposed into visual, prompt, and response tokens:

$$N = N_v + N_p + N_r. \tag{16}$$

T2I retention For T2I, each unprotected 2×2 cell is shortened from 4 tokens to 2, while protected cells keep all 4. Let $\mathcal{C}$ be the set of cells, $\mathcal{P}$ the

Variant	Steps	Layers (channel)	Speedup	Visual outcome
Shallow-window	$[s16,s45)$	$0-15$ (attn)	$+4.4\%$	speckle; recognizable
All attention	$[s16,s45)$	$0-31$ (attn)	$+9.5\%$	catastrophic speckle
Full trajectory	$[s0,s50)$	$0-15$ (attn)	$+7.3\%$	drift residue worsens
Attention + FFN	$[s16,s45)$	$0-15$ (attn+MLP)	$+13.1\%$	unusable noise

Table 9: ToMeSD-style direct-port diagnostics on LLaDA-o T2I. Each row changes one expansion axis relative to the shallow-window variant; visual outcomes are from qualitative inspection of generated samples.

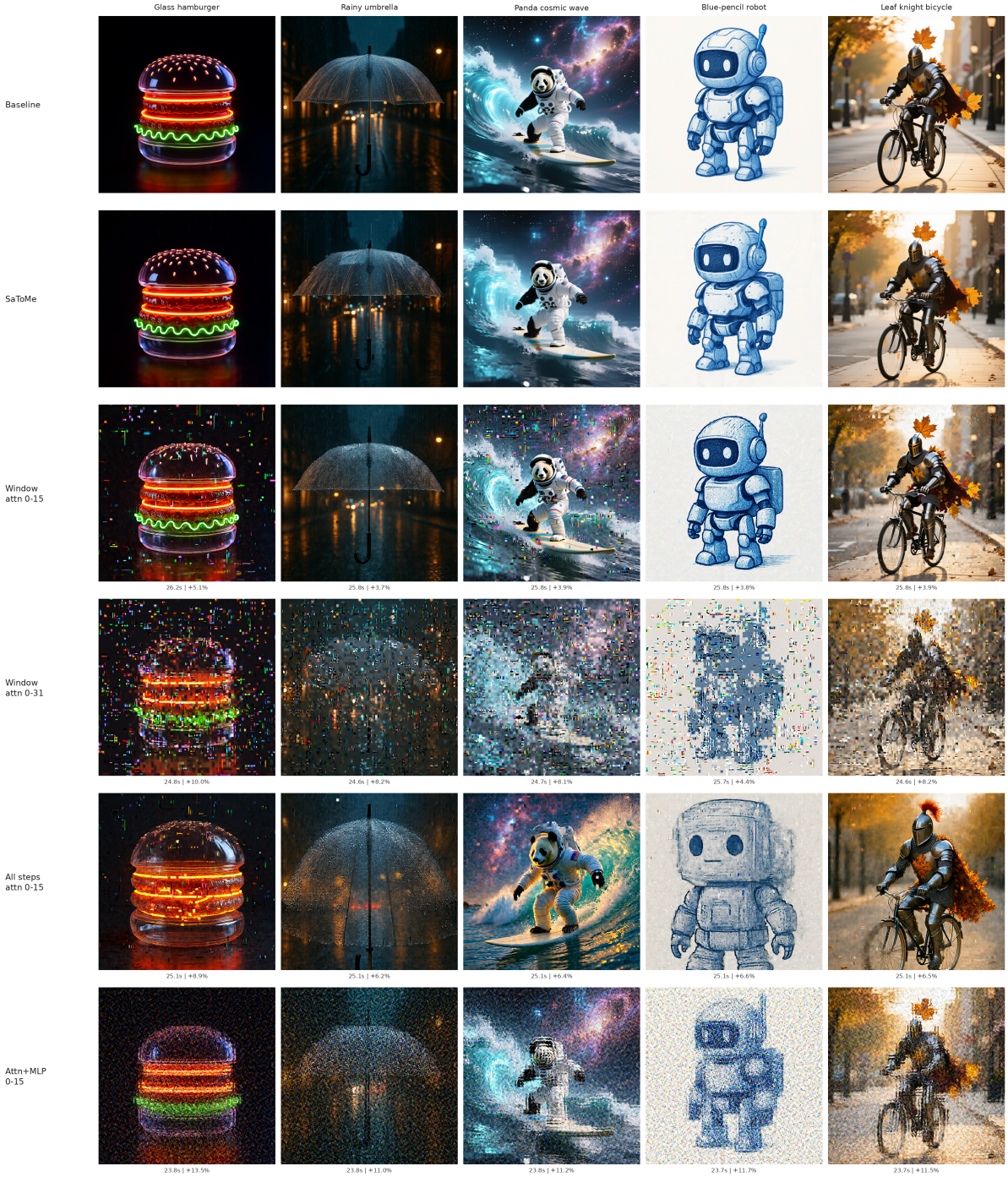


Figure 10: Qualitative comparison on five representative T2I prompts (columns). Rows from top: *Baseline* (unmodified LLaDA-o); *SaToMe* (ours, r_p=0.35, $+20.14\%$ wall-clock); and the four ToMeSD-port variants described in Table 9. Per-image wall-clock speedup is annotated in each panel. The variants exhibit visible degradation at lower speedups, whereas SaToMe (second row) matches the baseline visually at $+20.14\%$.

protected set, and $r_p = |\mathcal{P}|/|\mathcal{C}|$ the protection ratio. The retained visual-token ratio after merging is

$$\rho_{\text{T2I}} = 1 - \tfrac{1}{2}(1 - r_p). \tag{17}$$

I2T merged length For I2T, the top-r fraction of visual tokens is merged and removed:

$$N^{\text{I2T}}_{\text{merged}} = N - r \cdot N_v. \tag{18}$$

Per-layer cost The per-layer cost for a sequence of length n is approximated as

$$C_{\text{layer}}(n) = \underbrace{4nd^2 + 2n^2 d}_{\text{self-attention}} + \underbrace{3ndm}_{\text{FFN}}, \tag{19}$$

where d is the hidden size and m is the FFN intermediate width.

Merging overhead The merging overhead is linear in the number of visual tokens and includes salience scoring, cell-local enumeration, and nearest-destination assignment:

$$C_{\text{overhead}} = O(N_v \cdot k \cdot d), \tag{20}$$

where k is the neighborhood kernel size.

Per-row FLOPs ratios. The values in the FLOPs column of Table 3 instantiate the expressions above with backbone-specific parameters. With L=32, d=4096, m=12288 (Appendix A), the per-step ratio is

$$r_{\text{step}}(N, N_v, r, \ell^*) = \frac{\ell^* C_{\text{layer}}(N) + (L - \ell^*) C_{\text{layer}}(N - rN_v)}{L\, C_{\text{layer}}(N)}, \tag{21}$$

where $r \in \{0.5, 0.75, 0.9\}$ corresponds to retention $\{50\%, 25\%, 10\%\}$. For LaViDa I2T, the SigLIP-so400m-patch14-384 encoder emits a fixed N_v=729; with ℓ^*=1 (post-layer-1 merge) and total prefill $N \approx 806$ on GQA-lite (including the N_r=16 generation budget), Eq. 21 yields the published 0.555 at 50% and $0.337/0.207$ at $25\%/10\%$. For LLaDA-o I2T, the NaViT-style encoder emits an image-dependent N_v averaging $\approx 1{,}912$ on GQA-lite; with pre-LLM merging (ℓ^*=0), Eq. 21 yields $0.502/0.265/0.127$. For LLaDA-o T2I, merging is active only on T_{active}=30 of T=50 denoising steps within the window [s16, s45], so the effective per-trajectory ratio averages full and merged steps:

$$r^{\text{eff}}_{\text{T2I}} = \tfrac{T - T_{\text{active}}}{T} + \tfrac{T_{\text{active}}}{T} \cdot r_{\text{step}}(N, N_v, 1 - \rho_{\text{T2I}}, 0), \tag{22}$$

which at N_v=4096, r_p=0.35 (so ρ_{T2I}=0.675), and $N \approx 4156$ gives 0.790, consistent with the observed $+20.14\%$ wall-clock speedup. The merging overhead C_{overhead} is linear in N_v and contributes $< 0.1\%$ of the per-step cost in all configurations; we omit it from the reported ratios.