Slow-Vision, Fast-Language: Training-Free Efficient Inference for dMLLMs

Anonymous ACL submission

Abstract

Diffusion-based Multimodal Large Language Models (dMLLMs) represent a promising frontier in generative AI, yet their practical deployment is severely hindered by the computational burden of iterative denoising on high-resolution visual sequences. In this work, we identify a fundamental asymmetry: visual tokens exhibit high spatial redundancy and tend to aggregate semantically, whereas text tokens drive the dynamic evolution of reasoning. Challenging the coarse "all-or-nothing" approach of existing token pruning methods, we propose **Slow-Vision, Fast-Language (SVFL)**, a training-free acceleration paradigm for dMLLMs. SVFL maintains the complete visual panorama in intermittent "slow" layers, allowing only specific visual details to be dynamically summoned by text attention for efficient interaction in frequent "fast" layers. Extensive experiments on the state-of-the-art dMLLM, **LLaDA-V**, demonstrate that SVFL achieves significant inference acceleration with negligible performance degradation. Furthermore, we verify the framework's universality on the autoregressive **LLaVA-1.5**, confirming its effectiveness across diverse generative paradigms.

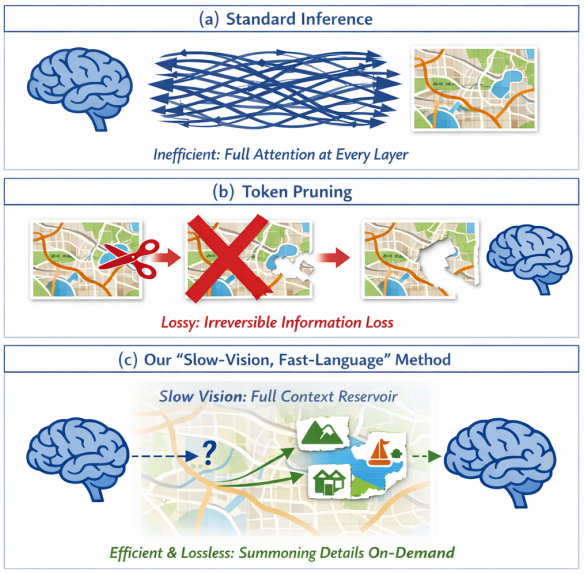

Figure 1: **Conceptual comparison. (a) Standard Inference** inefficiently re-scans the full map (visual context) at every layer. **(b) Token Pruning** permanently discards information, akin to tearing the map. **(c) Our Slow-Vision, Fast-Language** paradigm maintains the full map in a *Slow* path while dynamically "summoning" only text-relevant details for the *Fast* path, balancing efficiency and integrity.

1 Introduction

"The whole picture awaits, summoned by the word."

The landscape of Multimodal Large Language Models (MLLMs) Liu et al. (2023, 2024a); Chen et al. (2024c); Wang et al. (2024b, 2025) is undergoing a paradigm shift. While autoregressive models Radford et al. (2018, 2019); Brown et al. (2020); Touvron et al. (2023a,b); Bi et al. (2024) have long dominated the field, the emergence of diffusion-based Multimodal Large Language Models (dMLLMs) Bao et al. (2023); Xie et al. (2024); Ma et al. (2025); Nie et al. (2025), presents a compelling alternative. By treating sequence generation as a parallel iterative denoising process, dMLLMs offer distinct advantages in non-autoregressive planning and generation flexibility. However, this promising paradigm introduces severe computational bottlenecks. In contrast to the single-pass inference of autoregressive models, dMLLMs must process high-resolution visual embeddings alongside text latents within the heavy Transformer backbone at each denoising step. Given that modern visual encoders Radford et al. (2021); Zhai et al. (2023); Tschannen et al. (2025) typically yield thousands of visual tokens—orders of magnitude more than textual tokens—the quadratic complexity of attention is compounded by the iterative nature of diffusion, causing prohibitive memory consumption and latency that hinder practical deployment.

A pivotal observation is that visual information, while dense in quantity, exhibits significant spatial redundancy, whereas text drives the dynamic evolution of generation Shi et al. (2023); Chen et al. (2024a); Zhang et al. (2024). This raises a fundamental question: Is it necessary to re-calculate the full attention matrix for every visual token at every denoising step or transformer layer? Existing acceleration methods, mostly designed for autoregressive models, often resort to "token pruning"—permanently discarding visual tokens deemed redundant at early layers Chen et al. (2024a); Zhang et al. (2024); Yang et al. (2025b). While this reduces computation, it is inherently lossy for multimodal generation. Note that multimodal generation is a dynamic process where fine-grained visual details, which appear secondary in early layers, can evolve into key clues for verifying the answer in later stages.

We argue that the trade-off between "processing everything" (inefficient) and "discarding once-for-all" (lossy) is a false dichotomy. Inspired by human cognitive strategies, we propose a more refined approach. Fig. 1 illustrates this intuition through the analogy of a treasure hunter using a map (vision) to solve a riddle (language). The hunter does not obsessively re-scan every inch of the map at every second (Fig. 1a), nor do they throw away parts of the map after a single glance (Fig. 1b). Instead, the full map awaits in the periphery (Slow), acting as a reservoir of information. From this holistic view, specific visual details are summoned by the riddle's words to be actively scrutinized in the immediate thought process (Fast) (Fig. 1c).

Translating this intuition into a computational framework, we introduce the "Slow-Vision, Fast-Language" inference paradigm. Our core hypothesis is that the complete visual panorama should await at designated layers, serving as a stable reference pool (Slow). From this global context, only the key visual tokens are summoned by the text attention to participate in the frequent, agile updates of subsequent layers (Fast). Specifically, rather than permanently pruning tokens, we maintain the full visual context at intermittent "slow" layers. For the majority of "fast" layers, we dynamically select a subset of active visual tokens based on their relevance to the text. This strategy ensures access to complete visual grounding when necessary, avoiding information loss while significantly reducing the computational burden. Moreover, while our methodology is primarily optimized for dMLLMs, empirical evidence confirms its adaptability to autoregressive models, showing broad applicability. Overall, our main contributions are three-fold:

- **Critical Analysis:** We systematically identify the limitations of existing one-time visual token pruning strategies, elucidating how their static discarding mechanism leads to irreversible loss of fine-grained details essential for high-quality generation.
- **Novel Paradigm:** We propose the "Slow-Vision, Fast-Language" framework. By decoupling the processing frequency of visual and textual modalities, we achieve significant acceleration. This method allows the model to "rest" on heavy visual computation while staying "alert" on language generation.
- **Universal and Training-Free:** We primarily implement and validate our approach on the state-of-the-art diffusion model, LLaDA-V, achieving substantial speedups with negligible performance drop. Furthermore, to demonstrate the inherent universality of our mechanism, we extend the experiments to the classic autoregressive model LLaVA-1.5. The consistent efficacy across both architectures confirms that our SVFL addresses a fundamental redundancy in multimodal transformers that transcends specific generative paradigms.

2 Methodology

This section presents our **Slow-Vision, Fast-Language (SVFL)** framework, a training-free acceleration paradigm designed for multimodal LLMs. Our approach addresses the computational bottleneck of long visual sequences by dynamically modulating the processing frequency of visual tokens based on text-driven importance cues. The framework operates within the Transformer decoder layers, alternating between a "Slow Path" for global alignment and a "Fast Path" for efficient, sparse computation.

2.1 Overview of the SVFL Framework

Let $H^l \in \mathbb{R}^{N \times d}$ denote the input hidden states to the l-th Transformer decoder layer, where N is the sequence length and d is the hidden dimension. The sequence consists of text tokens H_{txt} and image tokens H_{img}. Our SVFL framework classifies each layer l into one of two types based on a predefined

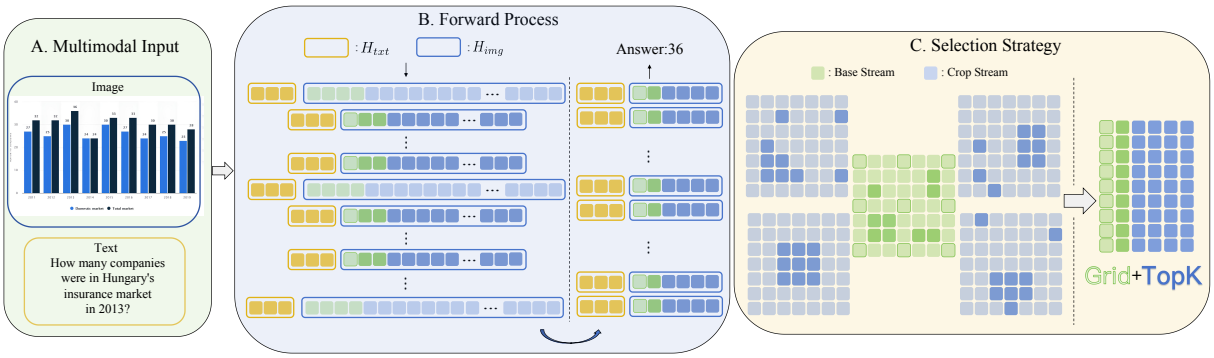

Figure 2: **Overview of SVFL.** The framework operates in three stages: **(A)** Handling dynamic resolution inputs via Base and Crop streams. **(B)** A training-free "SVFL" paradigm: The **Fast** stream (text + *selected* tokens) updates synchronously, while the **Slow** stream (*unselected*) updates only in full-interaction layers. Crucially, the Fast stream maintains a high retention rate in early layers to ensure sufficient accumulation, then aggressively reduces the selection count after the Lock Layer for efficiency. **(C)** A hybrid selection strategy (Text-Guided Top-K + Grid) that filters tokens at the Lock Layer, balancing semantic relevance with structural preservation.

configuration: *slow layer* ($l \in \mathcal{L}_{slow}$) or *fast layer* ($l \notin \mathcal{L}_{slow}$).

The overall process is governed by a funnel strategy with a locking mechanism. As illustrated in Fig. 2, early layers use a higher visual preservation ratio to capture broad semantics, while deeper layers use a lower ratio to focus on specific details. Beyond a critical lock layer L_{lock}, the selected visual indices are frozen to reduce overhead.

2.2 Slow Path: Intermittent Global Alignment

Layers in the set $\mathcal{L}_{slow}$ serve as synchronization points. Their primary role is to (1) perform full-context attention to establish global text-vision alignment, and (2) provide fresh attention weights to guide subsequent fast layers.

Before computation, if the preceding layer was a fast layer, its sparse output states H_f^{l-1} must be restored to the full sequence space. We employ a vectorized scatter operation:

$$H_s^{l-1} = Scatter(H_f^{l-1}, \mathcal{M}_{keep}), \qquad (1)$$

where $\mathcal{M}_{keep}$ is the boolean mask recording the positions of kept tokens. The restored full states are then processed by a standard Transformer block.

The Voting Strategy. Crucially, we extract the attention weights A^l from these slow layers to determine token importance. A naive approach might consider only the maximum attention from text. However, we argue that taking the maximum ignores the cumulative relevance of visual tokens attended to by multiple text tokens.

Therefore, we adopt a Summation-based Voting Strategy. We first aggregate attention scores by summing across all K attention heads to preserve signal intensity. We then calculate the importance score S_j^l for the j-th visual token by accumulating the attention it receives from all text tokens:

$$S_j^l = \sum_{i \in \mathcal{I}_{txt}} \left(\sum_{k=1}^{K} A_{k,i,j}^l \right), \qquad (2)$$

where $A^l k, i, j$ represents the attention weight from the i-th token (query) to the j-th token (key) in the k-th attention head, and $\mathcal{I}_{txt}$ is the set of indices corresponding to text tokens. This formulation ensures that visual tokens acting as common grounding points for multiple text tokens or capturing strong signals in specific heads receive higher importance scores, effectively acting as a cumulative vote from the language modality.

2.3 Fast Path: Visual Context Summoning

In fast layers, we perform sparse computation by selectively processing a subset of visual tokens. The core is a modality-aware token selection mechanism, denoted as $Extract(\cdot)$, which determines the binary keep mask $\mathcal{M}_{keep} \in \{0, 1\}^N$. The selection process utilizes the importance scores S derived from the most recent slow layer. We define target vision ratios ρ_1 for layers $l \leq L_{lock}$ and ρ_2 for layers $l > L_{lock}$. Our selection strategy is tailored to the structure of different visual inputs:

Modality-Aware Dual-Stream Selection. To balance structural preservation and feature salience, we employ a strategy combining grid-based sampling (structure) and Top-K selection (salience).

For Videos, we allocate a fixed proportion of the budget to grid sampling to maintain the spatiotem-

poral skeleton. The remaining budget is filled by selecting tokens with the highest importance scores S among the non-grid tokens (Top-K), effectively capturing motion and fine-grained details.

For Images with AnyRes, which consist of a base low-res view and multiple high-res crops, we apply a differentiated approach tailored to each component. For the *Base Stream*, a hybrid Grid+TopK strategy is applied to the base view to ensure a complete coarse-grained spatial reference. In contrast, for the *Crop Stream*, a pure Top-K strategy is applied to high-res crops. Since crops represent local details, we allocate the budget strictly based on attention scores S, allowing the model to "zoom in" on semantically relevant regions while discarding redundant background patches.

Sparse Computation and Locking. Once $\mathcal{M}_{keep}$ is determined, we gather the selected tokens to form a dense, shorter sequence:

$$H_f^l = H_s^l[\mathcal{M}_{keep}]. \tag{3}$$

The attention mask and position ids are adjusted accordingly. This sparse sequence is then processed efficiently by the Transformer. To further reduce the overhead of selection, if current layer $l > L_{lock}$, the mask $\mathcal{M}_{keep}$ computed at layer L_{lock} is cached and reused for all subsequent layers.

2.4 Implementation Details

Our SVFL operates as a training-free, plug-and-play module. Scatter and extract operations are fully vectorized for efficiency. To maintain compatibility with positional embeddings and attention masks, we dynamically slice and reconstruct the necessary metadata for sparse sequences, thereby preserving the spatial semantics of selected tokens.

3 Experiments

To validate the effectiveness of the SVFL framework, we conducted extensive evaluations across a broad spectrum of multimodal benchmarks.

3.1 Experimental Setup

Cross-Paradigm Models. We integrated SVFL into two representative MLLMs to demonstrate its generalization capability across distinct modeling paradigms. First, we employed **LLaDA-V** You et al. (2025), a pioneer of the emerging *diffusion-based language models*. While LLaDA-V offers novel generative capabilities, it incurs significant computational costs due to the iterative processing of its massive visual context; validating SVFL on it proves our framework's compatibility with non-autoregressive generation schemes. Second, to verify the method's universality on *autoregressive models*, we extended our experiments to LLaVA-1.5 Liu et al. (2024a). We selected this model for two strategic reasons: (1) it serves as a foundational framework in the open-source community, ensuring broad relevance; and (2) it is the primary testbed for the leading acceleration baseline, VisionZip Yang et al. (2025b) . Adopting the exact same backbone allows for a strictly fair, side-by-side comparison to demonstrate the superiority of our approach against state-of-the-art pruning methods. All experiments were conducted in a strictly training-free manner, applying our token selection mechanism directly to the official pre-trained checkpoints.

Benchmarks. We utilize a suite of **15 datasets** to evaluate capabilities in *General Understanding* (e.g., MMMU, MMStar, MMBench), *Fine-grained Perception* (e.g., DocVQA, ChartQA), and *Video Analysis* (e.g., Video-MME). Please refer to Appendix A for the complete list and details.

Configuration and Metrics. Our framework involves key hyperparameters including visual retention ratios (ρ) and the lock layer depth (L_{lock}). We adopt a standardized configuration to balance performance and efficiency, with comprehensive experimental settings provided in Appendix B. Regarding evaluation metrics, we prioritize practical throughput over theoretical proxy metrics; thus, we report the total wall-clock time required to process the complete datasets to demonstrate real-world efficiency. All experiments were conducted on NVIDIA A800 GPUs.

3.2 Main Results

We present the quantitative comparison of our SVFL framework against the standard baseline (Full Tokens) and the state-of-the-art token pruning method, VisionZip Yang et al. (2025b). Tab. 1 reports the results on the diffusion-based LLaDA-V, while Tab. 2 details the performance on the autoregressive LLaVA-1.5.

Results on LLaDA-V. Tab. 1 summarizes the quantitative results on the diffusion-based LLaDA-V. Our SVFL framework demonstrates exceptional resilience, retaining 96.9% of the upper-bound performance on average even when discarding

Image Understanding Benchmarks									
Method	MMMU val	MMMU-Pro standard	MMMU-Pro vision	MMStar test	MME perp.	SeedB image	MMB en-dev	RealworldQA	Avg.
Upper Bound: Retain 100% Tokens									
LLaDA-V	48.6 100%	35.2 100%	18.6 100%	60.1 100%	1507 100%	74.8 100%	82.9 100%	63.4 100%	100%
Retain 50% Tokens									
VisionZip	46.0 94.1%	33.4 94.1%	13.6 72.3%	50.0 83.2%	1374 91.2%	71.8 96.0%	76.4 91.9%	57.2 90.2%	89.1%
SVFL (Ours)	48.1 98.4%	34.2 96.3%	17.3 92.0%	58.4 97.2%	1463 97.1%	73.8 98.7%	82.2 98.9%	61.2 96.5%	96.9%
Retain 25% Tokens									
VisionZip	45.7 93.5%	32.4 91.3%	13.5 71.8%	46.2 76.9%	1224 81.3%	68.2 91.2%	74.5 89.7%	50.3 79.3%	84.4%
SVFL (Ours)	47.1 96.3%	32.7 92.1%	15.2 80.9%	54.3 90.3%	1413 93.8%	72.4 96.8%	78.7 94.7%	60.5 95.4%	92.5%
OCR & Video Understanding Benchmarks									
Method	AI2D	ChartQA	DocVQA val	InfoVQA val	SeedB video	MuirBench	MLVU dev	VideoMME	Avg.
Upper Bound: Retain 100% Tokens									
LLaDA-V	77.7 100%	78.3 100%	83.9 100%	66.2 100%	53.8 100%	48.3 100%	59.6 100%	56.2 100%	100%
Retain 50% Tokens									
VisionZip	71.1 91.5%	48.5 61.9%	48.6 57.9%	40.5 61.2%	52.0 96.7%	46.0 95.2%	53.5 89.8%	50.5 89.9%	80.5%
SVFL (Ours)	77.2 99.4%	75.9 96.9%	81.6 97.3%	61.6 93.1%	52.5 97.6%	47.5 98.3%	58.9 98.8%	55.4 98.6%	97.5%
Retain 25% Tokens									
VisionZip	68.5 88.2%	32.3 41.3%	44.2 52.7%	32.4 48.9%	48.1 89.4%	41.5 85.9%	55.7 93.5%	54.0 96.1%	74.5%
SVFL (Ours)	73.7 94.9%	62.9 80.3%	72.0 85.8%	44.8 67.7%	51.4 95.5%	42.4 87.8%	59.2 99.3%	55.0 97.9%	88.7%

Table 1: **Performance on LLaDA-V (Diffusion Paradigm).** Comparison between the full-token upper bound, VisionZip (permanent pruning), and our SVFL (dynamic summoning). **Red** indicates the best performance among acceleration methods. Note the significant gap in OCR-heavy tasks (ChartQA, DocVQA, InfoVQA).

50% of visual tokens, significantly outperforming the state-of-the-art static pruning method, VisionZip (89.1%). The most striking advantage of SVFL is observed in benchmarks demanding high-resolution scrutiny, specifically ChartQA and DocVQA. As shown in the "Retain 25%" setting, VisionZip suffers a catastrophic performance collapse—dropping to 41.3% and 52.7% of the original capability on ChartQA and DocVQA, respectively. This empirically confirms our hypothesis: *static pruning in early layers irreversibly deletes small but critical visual cues before the language model recognizes their importance.* In contrast, SVFL effectively mitigates this loss, maintaining 80.3% and 85.8% of the performance on these tasks. By maintaining the full visual panorama in "slow layers," our method allows the model to dynamically "summon" these fine-grained details back into the computation flow when triggered by specific text queries, effectively bypassing the information bottleneck inherent in one-shot pruning. On long-context video benchmarks (VideoMME, MLVU), SVFL consistently surpasses VisionZip. For instance, on MLVU (Retain 25%), SVFL achieves a 99.3% retention rate compared to VisionZip's 93.5%. This validates the efficacy of our *grid Sampling* , which preserves the spatiotemporal skeleton of the video, ensuring that motion semantics are not disrupted by aggressive pruning.

Results on LLaVA-1.5. Tab. 2 demonstrates SVFL's superior robustness over VisionZip, especially in extreme compression regimes. With only 64 tokens (∼11%), SVFL retains **95.5%** of vanilla performance (vs. VisionZip's 92.8%) and significantly reduces hallucination on POPE (83.9 vs. 77.3). Furthermore, the **"Integrated"** setting evaluates compatibility by cascading SVFL after VisionZip pre-selection (`VZ` → `SVFL`). This setup reduces the computational load on SVFL's accumulation layers while enhancing precision. Remarkably, SVFL refines the pre-pruned tokens to

Method	GQA	MMB en-dev	MME	POPE	SQA	MMMU val	SeedB image	Avg.
Standalone Comparison (Direct Application)								
Upper Bound, 576 Tokens								
Vanilla	62.0 100%	64.1 100%	1510 100%	85.9 100%	70.4 100%	36 100%	66.2 100%	100%
Retain 192 Tokens								
VisionZip	59.2 95.5%	63.5 99.1%	1449 96.0%	85.5 99.5%	69.9 99.3%	36.1 100.3%	63.1 95.3%	97.8%
SVFL	60.4 97.4%	62.6 97.7%	1500 99.3%	85.7 99.8%	69.5 98.7%	36.7 101.9%	65.8 99.4%	99.2%
Integrated Comparison (Compatibility Test)								
Retain 128 Tokens								
VisionZip	57.7 93.1%	62.5 97.5%	1434 95.0%	83.1 96.7%	70.0 99.4%	37.2 103.3%	61.5 92.9%	96.8%
VZ → SVFL	59.3 95.6%	62.0 96.7%	1441 95.4%	85.8 99.9%	69.5 98.7%	36 100%	64.9 98.0%	97.8%
Retain 64 Tokens								
VisionZip	55.0 88.7%	59.9 93.4%	1364 90.3%	77.3 90.0%	70.1 99.6%	36.1 100.3%	57.7 87.2%	92.8%
VZ → SVFL	57.7 93.1%	58.3 91.0%	1436 95.1%	83.9 97.7%	69.1 98.2%	35.4 98.3%	62.9 95.0%	95.5%

Table 2: **Generalization and Compatibility Analysis on LLaVA-1.5.** We evaluate SVFL in two settings: *Standalone* (direct comparison) and *Integrated* (cascaded after VisionZip pre-selection). The Integrated setting demonstrates that SVFL can effectively refine pre-pruned tokens, enhancing performance even in extremely low-token regimes (64 tokens).

outperform the standalone VisionZip baseline by **2.7%**, proving its utility as a flexible "refiner" module compatible with existing methods.

3.3 Efficiency and Compatibility

Tab. 3 reports the total wall-clock time for processing full benchmarks. First, in a fair comparison using the standard PyTorch attention (Eager mode), SVFL significantly outperforms VisionZip. On ChartQA, SVFL reduces the time to 8,259s (1.69× speedup) compared to VisionZip's 10,287s (1.35×), verifying that our funnel-shaped pruning reduces the computational burden more effectively than uniform pruning. As shown in the last row, simply enabling `SDPA` (Scaled Dot Product Attention) boosts SVFL's speedup to **2.07**× (6,721s) on ChartQA. This result confirms that SVFL's structured token selection maintains favorable memory access patterns, making it highly compatible with modern kernel optimizations. This suggests that SVFL can be readily integrated with future inference engines to achieve even greater throughput.

3.4 Ablation Studies

We conducted comprehensive ablation studies on LLaDA-V to validate the effectiveness of our core design components.

Method	Time (s) ↓	Speedup ↑	Acc. ↑
Dataset: MMStar (High-Res)			
Standard (Vanilla)	1,699	1.00×	60.1
VisionZip (50%, Eager)	1,440	1.18×	50.0
SVFL (50-25%, Eager)	1,468	1.16×	53.7
SVFL (50-25%, **SDPA**)	**1,067**	**1.59**×	**54.3**
Dataset: ChartQA (Fine-Grained)			
Standard (Vanilla)	13,932	1.00×	78.3
VisionZip (50%, Eager)	10,287	1.35×	48.7
SVFL (50-25%, Eager)	8,259	1.69×	62.6
SVFL (50-25%, **SDPA**)	**6,721**	**2.07**×	**62.9**

Table 3: **Throughput Analysis and System Compatibility.** We report the full comparison metrics on MMStar and ChartQA. By stacking the datasets, we highlight that **SVFL (SDPA)** significantly outperforms both the Standard baseline and VisionZip. Notably, on the computation-heavy ChartQA, our method achieves a **2.07**× **speedup** with substantially higher accuracy retention than VisionZip.

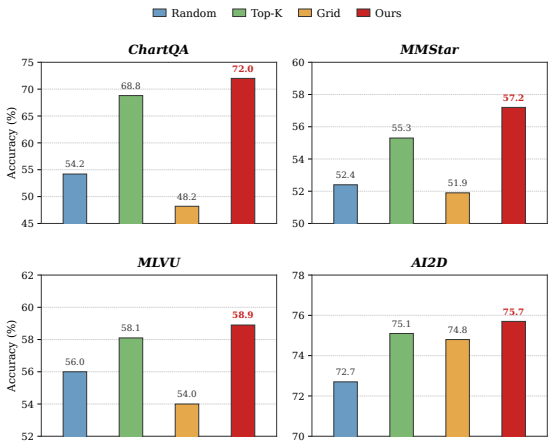

Figure 3: **Impact of Token Selection Strategies.** We compare our hybrid approach (`Grid+TopK`) against single-stream baselines (`Random`, `TopK`, `Grid`) across four benchmarks. While `TopK` captures semantics and `Grid` preserves layout, neither suffices alone. Our hybrid strategy consistently yields the best performance, demonstrating the necessity of coupling semantic salience with a spatial skeleton.

Impact of Selection Strategies. To validate the design of our Dual-Stream Selector, we evaluate four pruning strategies: `Random`, `TopK` (semantic-only), `Grid` (structure-only), and our hybrid `Grid+TopK`. The results, illustrated in Fig. 3, reveal two critical insights: **(1) Structure alone is insufficient for fine-grained tasks.** Surprisingly, the `Grid` strategy, which uniformly preserves the spatial skeleton, performs poorly on ChartQA (48.2), even falling behind `Random` selection (54.2). This suggests that in high-resolution document understanding, critical information (e.g., small text labels

Strategy	Avg. Ratio	Retention Ratios		Benchmarks						Avg.
		ρ_1 (Pre)	ρ_2 (Post)	MMMU	MMStar	AI2D	ChartQA	MLVU	VideoMME	
Baseline (Upper Bound)	100%	1.00	1.00	48.9	60.1	77.7	78.3	59.6	56.2	63.5
Iso-FLOPs Constraint (Strict Efficiency)										
Uniform	50%	0.50	0.50	47.7	56.3	76.0	71.8	59.2	54.9	61.0
Mild Funnel	50%	0.60	0.40	48.9	57.2	75.7	72.0	58.9	54.6	61.2
Aggressive Funnel	50%	0.75	0.25	47.7	54.9	73.7	65.1	58.8	54.6	59.1
Relaxed Constraint (High Fidelity)										
High-Fidelity Funnel	62.5%	0.75	0.50	48.1	58.4	77.2	75.9	58.9	55.4	62.3

Table 4: **Impact of Funnel Strategy.** We compare different budget allocation strategies against the uncompressed **Baseline** (Top Row). The **Mild Funnel** (60-40) offers the optimal balance under strict Iso-FLOPs (Avg. 50%). Crucially, the **High-Fidelity** setting ($\rho_1 = 0.75, \rho_2 = 0.50$) demonstrates that by slightly relaxing the constraint, we recover **98.2%** of baseline (Avg. 62.3 vs. 63.5). Note the significant recovery on ChartQA (75.9), which approaches the upper bound (78.3), proving that the pre-lock density ($\rho_1 = 0.75$) effectively preserves critical details.

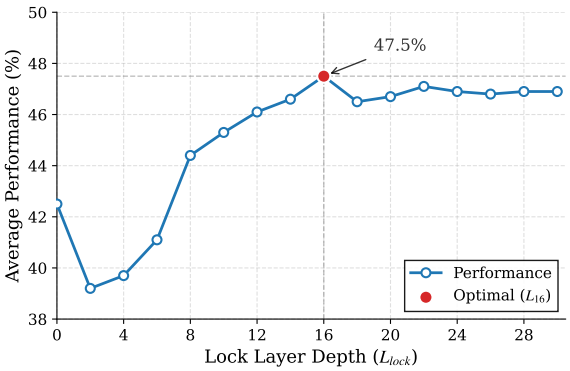

Figure 4: **Impact of Lock Layer (L_{lock}).** The performance on MuirBench exhibits a distinct "climb-and-saturate" trajectory. It initially rises as text-vision alignment matures, peaks at Layer 16 (47.5%), and subsequently plateaus, validating $L_{lock} = 16$ as the optimal trade-off point.

or data points) is sparse and non-uniform; a rigid grid blindly discards these high-frequency details. **(2) Synergy between Semantics and Structure.** While `TopK` significantly outperforms `Grid` by capturing salient regions, it risks losing global context. SVFL's `Grid+TopK` strategy achieves the best of both worlds, boosting performance to **72.0** on ChartQA and **57.2** on MMStar. This confirms that effectively resolving visual misalignment requires a "hybrid anchor": using `TopK` to catch the drifted proxy tokens containing semantic details, while using `Grid` to maintain the global scene composition.

Impact of Lock Layer. We varied the lock layer parameter L_{lock} from 0 to 30 to observe its impact on model performance. As illustrated in Fig. 4, the results exhibit a distinct "climb-and-saturate" trajectory. Setting L_{lock} too early (e.g., layers 2-6) results in suboptimal performance ($< 42\%$). Interestingly, locking at Layer 2 (39.2%) performs worse than locking immediately after the first attention calculation ($L_{lock} = 0$, 42.5%). This counter-intuitive dip suggests that the attention mechanism undergoes a volatile realignment phase in the earliest layers, rendering the importance rankings temporarily less reliable than the initial extraction. Performance then improves steadily from Layer 8 to 14 as the text-vision alignment matures, reaching a distinct peak at Layer 16 (47.5%). Beyond this point (Layers 18-30), performance plateaus around 47%, indicating that the visual importance distribution has stabilized. Therefore, we select $L_{lock} = 16$ as the optimal equilibrium, maximizing generation quality while freezing the visual indices early enough to save computational costs in the subsequent half of the network.

Impact of Funnel Strategy. We first establish the performance upper bound using the uncompressed Baseline (Avg. 63.5), as shown in the top row of Tab. 4. Under the strict Iso-FLOPs constraint (Avg. 50%), the Mild Funnel ($\rho_1 = 0.6, \rho_2 = 0.4$) achieves the best trade-off (Avg. 61.2), while the Aggressive Funnel ($\rho_1 = 0.75, \rho_2 = 0.25$) suffers a severe drop on ChartQA (65.1 vs. 78.3), indicating a capacity collapse. However, the potential of our method is fully realized in the High-Fidelity setting ($\rho_1 = 0.75, \rho_2 = 0.50$). By restoring the post-lock capacity, the model effectively closes the gap with the Baseline. Most notably, on the detail-sensitive ChartQA benchmark, the score surges to **75.9**, recovering within **2.4 points** of the lossless baseline (78.3). Overall, this configuration achieves **98.2%** of the Baseline (62.3/63.5) with significantly reduced computation. This confirms that the dense early accumulation is indeed capturing the necessary details, and SVFL can approach lossless performance given a moderate token budget.

4 Discussion

4.1 Revisiting Visual Misalignment

VisionZip Yang et al. (2025b) correctly identifies that visual encoders often aggregate information into "proxy tokens" that spatially drift from the original subject, prompting them to abandon text-guided selection to avoid noise. While we acknowledge this phenomenon in early layers, we argue that bypassing textual guidance introduces "Semantic Blindness." As evidenced by our experiments (Tab. 1), purely visual pruning struggles on detail-sensitive tasks like ChartQA, where critical information (e.g., tiny axis labels) is visually inconspicuous and effectively discarded. This highlights a fundamental limitation of static methods: visual salience is not equivalent to semantic relevance.

4.2 Temporal Accumulation over Avoidance

Instead of abandoning text-guided selection to avoid misalignment, SVFL resolves it through a "Slow-Fast" accumulation paradigm, positing that prior observed failures were largely symptoms of *premature evaluation*. As evidenced by the stabilization of attention maps in deeper layers (Fig. 4), our pre-lock accumulation stage provides a necessary computational buffer, allowing the LLM to effectively re-map textual queries to the correct visual features—even if they have drifted into "proxy tokens." Furthermore, SVFL complements this dynamic selection with a static Grid Stream, ensuring the preservation of the image's spatial skeleton. By marrying structural integrity (via the grid) with delayed, accumulation-based semantic selection (via the text), SVFL effectively mitigates misalignment without sacrificing the fine-grained precision required for complex reasoning.

5 Related Work

5.1 Vision-Language Models

The paradigm of Multimodal Large Language Models (MLLMs) has shifted significantly towards aligning powerful visual encoders with Large Language Models (LLMs) Liu et al. (2023, 2024a); Yang et al. (2025a); You et al. (2025). Prominent architectures, such as LLaVA Liu et al. (2024a,b), typically project features from a frozen visual encoder (e.g., CLIP Radford et al. (2021), SigLIP Zhai et al. (2023)) into the LLM's embedding space. To enhance fine-grained perception capabilities, recent trends advocate for increasing visual resolution or employing dynamic tiling strategies Liu et al. (2024b), which exponentially increases the number of visual tokens. Furthermore, the emergence of generative-based models like LLaDA-V You et al. (2025) extends this architecture to diffusion processes. However, this resolution scaling imposes a severe computational burden due to the quadratic complexity of the self-attention mechanism in Transformers, creating a pressing need for efficiency optimization that does not sacrifice the model's ability to perceive intricate details.

5.2 Training-free Visual Token Reduction

Given the training costs, training-free methods have gained traction. Text-guided approaches, such as FastV Chen et al. (2024a) and SparseVLM Zhang et al. (2024), utilize the attention weights from the LLM to identify and prune irrelevant visual tokens. However, these methods typically perform pruning based on early-layer attention maps, which are often noisy and unstable, leading to the premature discard of critical information. Conversely, vision-centric approaches like VisionZip Yang et al. (2025b) propose selecting tokens based solely on the visual encoder's self-attention to avoid text-vision misalignment. Yet, by ignoring textual guidance, such methods introduce "semantic blindness," failing to retain visually inconspicuous but semantically critical elements (e.g., small text in charts). In contrast, our SVFL adopts a "Slow-Fast" paradigm that allows for sufficient text-vision alignment before locking the mask, effectively balancing semantic precision with inference efficiency.

6 Conclusion

In this paper, we presented SVFL, a training-free framework designed to reconcile inference efficiency with fine-grained information preservation in Multimodal Large Language Models. By introducing a dynamic summoning mechanism governed by accumulative attention, SVFL effectively retrieves critical visual details from early "slow layers" while accelerating computation in later stages. Our experiments on both LLaDA-V and LLaVA-1.5 demonstrate that SVFL significantly outperforms state-of-the-art static pruning methods. Most notably, it successfully mitigates the "fine-grained collapse" observed in previous works, retaining over 80% of the original performance on resolution-sensitive benchmarks like ChartQA and DocVQA even under aggressive compression regimes.

Limitations

Despite the effectiveness of SVFL in balancing efficiency and fine-grained perception, we identify two primary limitations in our current implementation.

Dependency on Attention Weights. Our method relies on extracting and accumulating attention scores from the Multi-Head Self-Attention (MHSA) layers to guide the token summoning process. However, modern LLM architectures like Qwen3 Yang et al. (2025a) and high-efficiency inference backends default to using FlashAttention Dao et al. (2022). FlashAttention optimizes memory and speed by computing the attention output without materializing the full $N \times N$ attention matrix, thereby making the intermediate weights inaccessible. Consequently, applying SVFL to such models currently requires disabling FlashAttention or relying on slower, standard attention implementations. Our future work will explore custom CUDA kernels or gradient-based approximations to retrieve importance scores compatible with FlashAttention.

Theoretical Bound on Acceleration. SVFL follows a "Slow-Fast" paradigm, where the pre-lock accumulation stage (Layers $0 \rightarrow L_{lock}$) is essential for accumulating accurate semantic signals. Inevitably, this imposes a higher computational baseline compared to methods that perform immediate pruning at the input layer; theoretically, under identical retention ratios, SVFL incurs higher latency. However, this design trade-off is strategic: the robust semantic grounding established in the slow stage enables us to employ **significantly more aggressive reduction ratios** (e.g., 25% retention) in the subsequent fast stage without the performance collapse often seen in early-pruning methods. As evidenced in Tab. 3 (Eager mode), even with the pre-lock overhead, SVFL at lower retention ratios achieves inference speeds comparable to aggressive pruning baselines (e.g., 1.16× vs. 1.18×) while yielding superior accuracy (53.7% vs. 50.0%).

We utilized AI assistants (e.g., Gemini) to assist with text polishing. All scientific claims and experimental results were verified by the authors.

References

Fan Bao, Shen Nie, Kaiwen Xue, Chongxuan Li, Shi Pu, Yaole Wang, Gang Yue, Yue Cao, Hang Su, and Jun Zhu. 2023. One transformer fits all distributions in multi-modal diffusion at scale. In *International Conference on Machine Learning*, pages 1692–1717. PMLR.

Xiao Bi, Deli Chen, Guanting Chen, Shanhuang Chen, Damai Dai, Chengqi Deng, Honghui Ding, Kai Dong, Qiushi Du, Zhe Fu, and 1 others. 2024. Deepseek llm: Scaling open-source language models with longtermism. *arXiv preprint arXiv:2401.02954*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

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*, pages 19–35. Springer.

Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Jiaqi Wang, Yu Qiao, Dahua Lin, and 1 others. 2024b. Are we on the right way for evaluating large vision-language models? *Advances in Neural Information Processing Systems*, 37:27056–27087.

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, and 1 others. 2024c. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 24185–24198.

Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. 2022. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in neural information processing systems*, 35:16344–16359.

Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, and 1 others. 2025a. Mme: A comprehensive evaluation benchmark for multimodal large language models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, and 1 others. 2025b. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 24108–24118.

Drew A Hudson and Christopher D Manning. 2019. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 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*, pages 235–251. Springer.

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

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. 2023b. Evaluating object hallucination in large vision-language models. *arXiv preprint arXiv:2305.10355*.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2024a. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 26296–26306.

Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. 2024b. Llavanext: Improved reasoning, ocr, and world knowledge.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916.

Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, and 1 others. 2024c. Mmbench: Is your multi-modal model an all-around player? In *European conference on computer vision*, pages 216–233. Springer.

Pan Lu, Swaroop Mishra, Tanglin Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. 2022. Learn to explain: Multimodal reasoning via thought chains for science question answering. *Advances in Neural Information Processing Systems*, 35:2507–2521.

Yiyang Ma, Xingchao Liu, Xiaokang Chen, Wen Liu, Chengyue Wu, Zhiyu Wu, Zizheng Pan, Zhenda Xie, Haowei Zhang, Xingkai Yu, and 1 others. 2025. Janusflow: Harmonizing autoregression and rectified flow for unified multimodal understanding and generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 7739–7751.

Ahmed Masry, Xuan Long Do, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. 2022. Chartqa: A benchmark for question answering about charts with visual and logical reasoning. In *Findings of the association for computational linguistics: ACL 2022*, pages 2263–2279.

Minesh Mathew, Viraj Bagal, Rubèn Tito, Dimosthenis Karatzas, Ernest Valveny, and CV Jawahar. 2022. Infographicvqa. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1697–1706.

Minesh Mathew, Dimosthenis Karatzas, and CV Jawahar. 2021. Docvqa: A dataset for vqa on document images. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 2200–2209.

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*.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, and 1 others. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, and 1 others. 2018. Improving language understanding by generative pre-training.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, and 1 others. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Dachuan Shi, Chaofan Tao, Ying Jin, Zhendong Yang, Chun Yuan, and Jiaqi Wang. 2023. Upop: Unified and progressive pruning for compressing vision-language transformers. In *International Conference on Machine Learning*, pages 31292–31311. PMLR.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, and 1 others. 2023a. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, and 1 others. 2023b. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, and 1 others. 2025. Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786*.

Fei Wang, Xingyu Fu, James Y Huang, Zekun Li, Qin Liu, Xiaogeng Liu, Mingyu Derek Ma, Nan Xu, Wenxuan Zhou, Kai Zhang, and 1 others. 2024a. Muirbench: A comprehensive benchmark for robust multi-image understanding. *arXiv preprint arXiv:2406.09411*.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin

Wang, Wenbin Ge, and 1 others. 2024b. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*.

Yi Wang, Xinhao Li, Ziang Yan, Yinan He, Jiashuo Yu, Xiangyu Zeng, Chenting Wang, Changlian Ma, Haian Huang, Jianfei Gao, and 1 others. 2025. Internvideo2. 5: Empowering video mllms with long and rich context modeling. *arXiv preprint arXiv:2501.12386*.

x.ai. 2024. Grok-1.5 Vision preview. https://x.ai/news/grok-1.5v/. [Online]. Available: https://x.ai/news/grok-1.5v/.

Jinheng Xie, Weijia Mao, Zechen Bai, David Junhao Zhang, Weihao Wang, Kevin Qinghong Lin, Yuchao Gu, Zhijie Chen, Zhenheng Yang, and Mike Zheng Shou. 2024. Show-o: One single transformer to unify multimodal understanding and generation. *arXiv preprint arXiv:2408.12528*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025a. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Senqiao Yang, Yukang Chen, Zhuotao Tian, Chengyao Wang, Jingyao Li, Bei Yu, and Jiaya Jia. 2025b. Visionzip: Longer is better but not necessary in vision language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 19792–19802.

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*.

Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, and 1 others. 2024. 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.

Xiang Yue, Tianyu Zheng, Yuansheng Ni, Yubo Wang, Kai Zhang, Shengbang Tong, Yuxuan Sun, Botao Yu, Ge Zhang, Huan Sun, and 1 others. 2025. Mmmu-pro: A more robust multi-discipline multimodal understanding benchmark. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15134–15186.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. 2023. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 11975–11986.

Yuan Zhang, Chun-Kai Fan, Junpeng Ma, Wenzhao Zheng, Tao Huang, Kuan Cheng, Denis Gudovskiy, Tomoyuki Okuno, Yohei Nakata, Kurt Keutzer, and 1 others. 2024. Sparsevlm: Visual token sparsification for efficient vision-language model inference. *arXiv preprint arXiv:2410.04417*.

Junjie Zhou, Yan Shu, Bo Zhao, Boya Wu, Shitao Xiao, Xi Yang, Yongping Xiong, Bo Zhang, Tiejun Huang, and Zheng Liu. 2024. Mlvu: A comprehensive benchmark for multi-task long video understanding. *arXiv e-prints*, pages arXiv–2406.

A Dataset Details

We construct a comprehensive evaluation suite comprising 15 datasets. These are strategically categorized to probe specific model capabilities:

General Image Understanding & Reasoning. To assess broad-coverage capabilities, we utilize MMMU (Val/Pro) (Yue et al., 2024, 2025), MM-Star (Chen et al., 2024b), MMBench (Liu et al., 2024c), SEED-Bench (Li et al., 2023a), MME (Fu et al., 2025a), and RealWorldQA (x.ai, 2024). Furthermore, we include GQA (Hudson and Manning, 2019), ScienceQA (SQA) (Lu et al., 2022), and POPE (Li et al., 2023b) as standard baselines to evaluate object hallucination and spatial reasoning capabilities in autoregressive settings.

Fine-grained Perception. Standard benchmarks often miss high-frequency details. We address this by testing on DocVQA (Mathew et al., 2021), InfoVQA (Mathew et al., 2022), ChartQA (Masry et al., 2022), and AI2D (Kembhavi et al., 2016). These document and chart-oriented tasks require precise visual parsing and are particularly sensitive to token dropping, serving as a stress test for our dual-stream selection mechanism.

Video & Temporal Analysis. For long-context capabilities, we employ Video-MME (Fu et al., 2025b), MLVU (Zhou et al., 2024), and Muir-Bench (Wang et al., 2024a). These tasks verify whether the model maintains temporal coherence and effectively retrieves key frames from redundant video streams.

B Detailed Parameter Configurations

In this section, we provide the precise hyperparameter settings used in our experiments. We adapt our configuration strategy based on the architectural characteristics of the underlying models, specifically distinguishing between dynamic-resolution and fixed-resolution frameworks.

Config	L_{lock}	ρ_1	ρ_2	$\mathcal{L}_{slow}$
Standard	-	1.0	1.0	All
Retain 50%	16	0.75	0.50	0, 6, 8, 10, 12, 16
Retain 25%	16	0.50	0.25	0, 6, 8, 10, 12, 16

Table 5: **Hyperparameters for Ratio-based Scenarios.** We report the lock layer depth (L_{lock}) and retention ratios (ρ) for different stages. The specific slow layers are denoted by $\mathcal{L}_{slow}$.

Config	L_{lock}	N_{vz}	ρ_1	ρ_2	$\mathcal{L}_{slow}$
Standard	-	-	576	576	All
Retain 128	16	384	256	128	0, 6, 8, 10, 12, 16
Retain 64	16	192	128	64	0, 6, 8, 10, 12, 16

Table 6: **Hyperparameters for Fixed-count Scenarios.** $\rho_{1,2}$ represent the exact target number of retained tokens in the subsequent stages. N_{vz} **denotes the number of tokens preserved from the original 576 visual tokens after VisionZip pre-selection.** $\mathcal{L}_{slow}$ specifies the indices of layers processed with the slow path.

Ratio-based Settings. For models employing the *AnyRes* technique, such as LLaDA-V, the number of input visual tokens varies dynamically based on the aspect ratio and resolution. To adapt to this variability, we adopt a Ratio-based strategy (as shown in Tab. 5). In these scenarios, ρ_1 and ρ_2 represent the *proportion* of retained tokens relative to the dynamic visual input before and after the lock layer, respectively, thereby ensuring consistent information density across different resolutions.

Fixed-count Settings. Conversely, for frameworks with a fixed input resolution, such as LLaVA-1.5 (which typically processes 576 visual tokens per image), we employ a Fixed-count strategy to strictly limit the computational budget (as detailed in Tab. 6). Here, ρ_1 and ρ_2 denote the *exact number* of retained tokens before and after the lock layer. Additionally, N_{vz} specifies the number of tokens preserved from the original 576 patches using VisionZip during the pre-selection phase.