ReTrace: Visual Backtracking for Efficient Multimodal Large Language Models

Guangzong Si[1,2,*], **Dong Wang**[3,*], **Yifan Yu**[4], **Miao Liu**[3], **Panwang Pan**[5], **Wentao Zhu**[2,†]

[1]University of Science and Technology of China, [2]Eastern Institute of Technology, Ningbo, [3]Tsinghua University, [4]ETH Zurich, [5]Xiamen University

guangzongsi@mail.ustc.edu.cn wangd16@tsinghua.org.cn wtzhu@eitech.edu.cn

* *Equal contribution* † *Corresponding author* *Project*

Abstract

Multimodal Large Language Models (MLLMs) encode images into dense visual token sequences, creating substantial attention and KV-cache overhead. Existing visual token compression methods reduce this cost by pruning low-saliency tokens in shallow layers, but this practice couples temporary computational deactivation with irreversible information deletion. We show that this coupling breaks an intrinsic inference pattern in MLLMs. Through layer-wise attention analysis, we identify **Visual Backtracking**: visual pivots remain spatially stable, yet in middle layers they actively retrieve context from tokens that appeared unimportant earlier. Thus, early low-saliency tokens can be inactive for computation while still being useful for later reasoning. Motivated by this finding, we propose **ReTrace**, a backtracking-aware compression framework that introduces a dormant token state. ReTrace caches non-pivot tokens outside expensive layer-wise computation, restores them during the backtracking window, and prunes only after their information has been absorbed. Experiments on Qwen2.5-VL and InternVL3 demonstrate nearly 90% reductions in FLOPs and KV cache, a 2.83× prefilling speedup, and strong performance retention across complex vision-language benchmarks.

1 Introduction

Multimodal LLMs (MLLMs) (Bai et al., 2025b; Zhu et al., 2025; Liu et al., 2024a, 2023) encode visual inputs into dense token sequences. This enables fine-grained cross-modal reasoning (Chen et al., 2025; Wang et al., 2025; Bai et al., 2023; Wang et al., 2024; Aravindan et al., 2025; Han et al., 2025; Jin et al., 2025), but the sequences can reach thousands of tokens per image as resolution and context grow (Radford et al., 2021; Caron et al., 2021; Bai et al., 2025a; Dosovitskiy et al., 2021). Processing all tokens is expensive, leading to quadratic attention cost and large KV cache (Vaswani et al., 2017; Dao et al., 2022; Dao, 2024; Kwon et al., 2023).

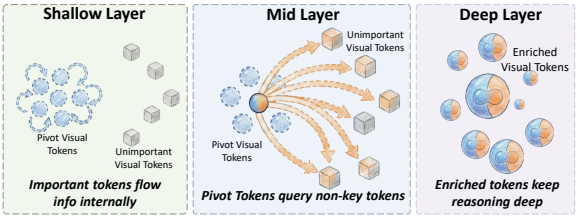


Figure 1: **Conceptual paradigm of the Visual Backtracking phenomenon.** In shallow layers, pivot tokens interact internally to build clear feature representations. In mid layers, a "backtracking" mechanism emerges, where pivot tokens strongly query unimportant visual tokens to absorb global context. Finally, in deep layers, forms *Enriched Visual Tokens* essential for complex reasoning.

A common solution is visual token pruning: tokens deemed unimportant in early layers are permanently removed or merged (Chen et al., 2024a; Vasu et al., 2025; Yang et al., 2025; Zou et al., 2025; Wen et al., 2025; Wu et al., 2026; Zhang et al., 2026; Yin et al., 2025; Shang et al., 2025; Liang et al., 2026; Bolya et al., 2023; Meng et al., 2025; Tan et al., 2025). This approach assumes that early low-saliency tokens are unimportant for the rest of inference. In practice, this assumption can fail: early pruning degrades performance on complex reasoning tasks, even when the retained tokens seem sufficient to reconstruct the visual content. This raises an interesting question:

> ***Question***
>
> *Can a token be temporarily unnecessary for computation but still useful for later reasoning?*

To answer this, we analyze not only which tokens are attended to, but also how attention flows across layers. We observe a three-phase pattern. In

Findings of the Association for Computational Linguistics: EMNLP 2026, pages 3869–3897
October 24-29, 2026 ©2026 Association for Computational Linguistics

shallow layers, task-relevant visual anchors interact mainly among themselves. In middle layers, these anchors shift attention outward toward tokens that were previously ignored. They actively retrieve context needed for global reasoning. In deep layers, attention reconcentrates on enriched anchors for task-specific decisions. We call this middle-layer outward retrieval **Visual Backtracking**, as shown in Fig. 1. It explains why early pruning fails: low-saliency tokens are not globally unimportant. They are phase-specific context sources accessed during a retrieval window that early pruning closes before it opens.

This finding reframes visual token compression. Instead of one-shot pruning, we treat it as *staged information scheduling*. Non-anchor tokens do not participate in all layers, but they should not be permanently deleted before the backtracking window. Based on this, we propose **ReTrace**. ReTrace builds a pivot buffer as surrogate receivers, caches non-pivot tokens as dormant context bypassing computation, and restores this context at the backtracking layer to approximate the missing retrieval flow. Task-driven pruning is applied only after absorption completes, deferring permanent information loss.

Our contributions are:

- We identify and quantify **Visual Backtracking**, showing that low-saliency tokens serve as phase-specific context sources during middle-layer inference. This provides a mechanistic explanation for why early pruning hurts performance.
- We formulate visual token compression as **staged information scheduling**, where tokens are cached, restored, and only later permanently pruned according to layer-wise demand.
- We implement this principle in **ReTrace** and show on Qwen2.5-VL and InternVL3 that it reduces FLOPs and KV cache by nearly 90%, achieves a 2.83× prefilling speedup, and maintains strong performance across complex vision-language benchmarks.

2 Related Work

MLLMs. MLLMs use Vision Transformers (Dosovitskiy et al., 2021) to encode images or videos into token sequences, which are then projected into the LLM's semantic space (Bai et al., 2025b; Zhu et al., 2025; Liu et al., 2023). Recent models like Qwen2.5-VL (Bai et al., 2023, 2025a; Wang et al., 2024; Bai et al., 2025b) and InternVL3 (Chen et al., 2025; Zhu et al., 2025; Wang et al., 2025) process high-resolution inputs to achieve fine-grained perception, resulting in long visual sequences. While this approach supports reasoning performance, the quadratic complexity of the self-attention mechanism increases computational overhead. Therefore, reducing visual context length without sacrificing performance is an important problem for MLLMs deployment.

Visual token pruning. Visual token pruning (Chen et al., 2024a; Vasu et al., 2025; Yang et al., 2025; Zou et al., 2025; Wen et al., 2025; Wu et al., 2026; Zhang et al., 2026; Yin et al., 2025) accelerates MLLM inference by reducing visual redundancy. Existing methods generally fall into vision-centric (Yang et al., 2025; Wen et al., 2025; Zou et al., 2025; Vasu et al., 2025; Wu et al., 2026) and text-centric (Chen et al., 2024a; Yin et al., 2025; Xing et al., 2025) approaches. Vision-centric methods filter or merge tokens before the LLM layers (Yang et al., 2025), while text-centric methods prune tokens in deeper layers based on cross-modal attention (Chen et al., 2024a). Recent works also explore hybrid strategies that combine early visual reduction with late-stage text-guided pruning (Zhang et al., 2026). G-Search (Zhao et al., 2025) further searches for layer-wise token retention schedules by exploiting the stability of token-importance rankings across layers. Our prior work, HiMAP (Yin et al., 2025), exploits the layer-wise shift between visual-textual and intra-visual information flows to guide hierarchical pruning. However, these methods ultimately make irreversible pruning decisions once visual tokens are removed, preventing their information from being accessed again in later layers. In contrast, ReTrace reveals that early low-saliency tokens can still be revisited by visual pivots in middle layers. Rather than permanently discarding them, we cache these tokens as dormant context, restore their information when needed, and apply task-guided pruning only after context absorption.

3 Motivation

Early visual token pruning reduces computation by removing low-saliency tokens from the active sequence. Yet this operation makes two decisions at once. It not only stops these tokens from partic-

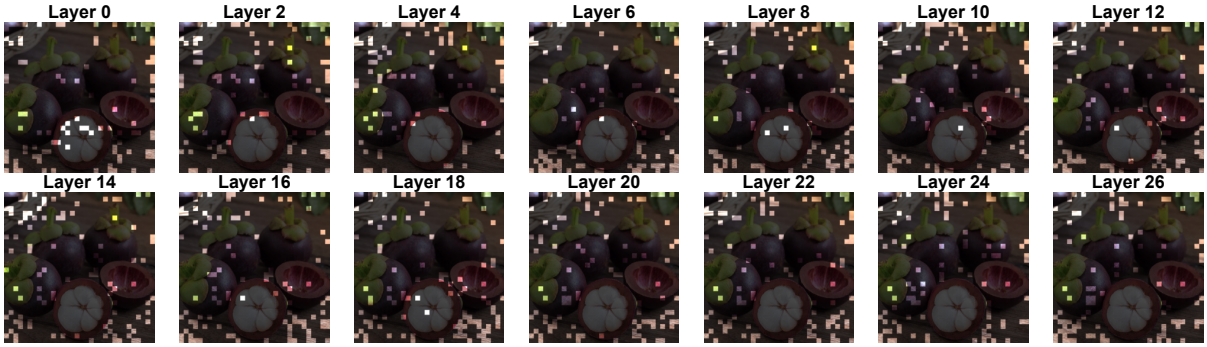


Figure 2: **Visualization of pivot tokens across different layers.** We highlight the top 10% visual tokens with the highest saliency scores. The selected regions remain highly consistent from shallow to deep layers, intuitively demonstrating the spatial stability of the model's visual attention.

ipating in expensive layer-wise computation, but also makes their information permanently inaccessible. This coupling is safe only if tokens ignored in shallow layers are never needed later.

We test this assumption by analyzing the layer-wise behavior of visual attention. Instead of only asking which tokens are important, we ask a more specific question: *after the model forms visual anchors, where do these anchors look as inference proceeds?*

Observation I: Visual pivots are spatially stable. We first examine whether the model forms stable visual anchors. If the identity of important visual tokens changes drastically across layers, then any structured analysis of attention flow would be unreliable.

Guided by textual instructions, we define the saliency score $s_l(v_i)$ of a visual token $v_i \in \mathcal{V}$ at layer l by averaging the cross-attention weights it receives from all text tokens, where $\mathbf{C}_l$ denotes the text-to-visual attention matrix at layer l:

$$s_l(v_i) = \frac{1}{|\mathcal{T}|} \sum_{t_j \in \mathcal{T}} \mathbf{C}_l(t_j, v_i) \tag{1}$$

Based on this ranking, we select the top-k tokens with the highest saliency as the pivot set $\mathcal{P}_l$, and treat the remaining tokens as the non-pivot set $\mathcal{U}_l$:

$$\mathcal{P}_l = \text{Top-}k(\{s_l(v_i) \mid v_i \in \mathcal{V}\}), \quad \mathcal{U}_l = \mathcal{V} \setminus \mathcal{P}_l \tag{2}$$

The cross-layer stability of the pivot set is defined as:

$$\text{Stability}_l = \frac{|\mathcal{P}_l \cap \mathcal{P}_{l-1}|}{|\mathcal{P}_{l-1}|} \tag{3}$$

As shown in Fig. 2, the top 10% pivot tokens consistently focus on the main visual subjects from shallow to deep layers. Fig. 3a further quantifies this stability at a 10% pivot ratio, showing that the selected pivots remain highly consistent across layers.

This result explains why early pruning can work to some extent: the model does form persistent visual anchors early. However, it also raises a sharper question. *If the pivot tokens are so stable, why does removing non-pivot tokens still hurt complex reasoning?* The issue may not be who the pivots are, but what information they need later.

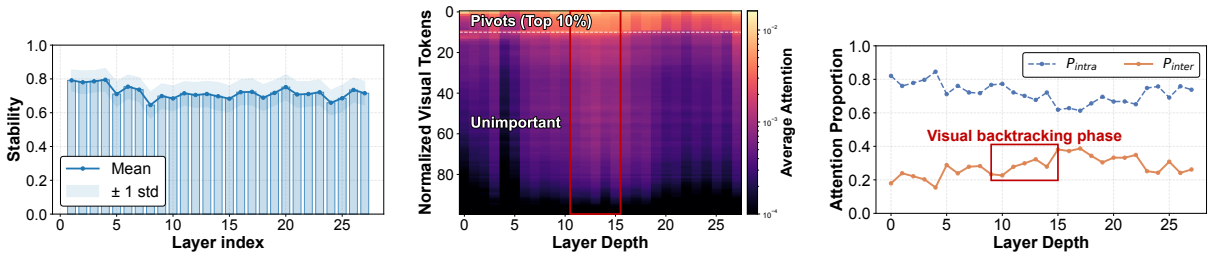
Observation II: Stable pivots look back to non-pivot context. We next study the attention flow starting from pivot tokens. Within the visual self-attention mechanism, we partition the attention originating from pivot tokens $\mathcal{P}_l$ into two parts, where $\mathbf{A}_l$ denotes the visual self-attention matrix at layer l. The first part is the internal attention among pivots, denoted as $\mathcal{F}^{(l)}_{intra}$. The second part is the outward attention from pivots to non-pivot tokens $\mathcal{U}_l$, denoted as $\mathcal{F}^{(l)}_{inter}$:

$$\begin{aligned} \mathcal{F}^{(l)}_{intra} &= \sum_{v_i \in \mathcal{P}_l} \sum_{v_j \in \mathcal{P}_l} \mathbf{A}_l(v_i, v_j), \\ \mathcal{F}^{(l)}_{inter} &= \sum_{v_i \in \mathcal{P}_l} \sum_{v_k \in \mathcal{U}_l} \mathbf{A}_l(v_i, v_k). \end{aligned} \tag{4}$$

We normalize these flows to obtain their relative proportions:

$$\begin{aligned} P^{(l)}_{intra} &= \frac{\mathcal{F}^{(l)}_{intra}}{\mathcal{F}^{(l)}_{intra} + \mathcal{F}^{(l)}_{inter}}, \\ P^{(l)}_{inter} &= \frac{\mathcal{F}^{(l)}_{inter}}{\mathcal{F}^{(l)}_{intra} + \mathcal{F}^{(l)}_{inter}}. \end{aligned} \tag{5}$$

Fig. 3c shows a clear three-phase pattern. In shallow layers, P_{intra} dominates. Pivot tokens mainly interact with each other, forming stable local visual



(a) Cross-layer pivot stability. (b) Pivot-to-visual attention heatmap. (c) Intra-/inter-pivot attention flow.

Figure 3: **Quantitative evidence of visual backtracking.** We analyze the layer-wise behavior of text-guided visual pivot tokens. (a) The selected pivots remain spatially stable across layers, indicating that the model establishes persistent visual pivots early. (b) Pivot-to-visual attention exhibits a distinct middle-layer expansion band, where pivots attend broadly to non-pivot visual tokens. (c) The normalized attention flow shows that pivots shift from pivot-internal consolidation in shallow layers to pivot-to-context retrieval in middle layers, revealing the visual backtracking window.

anchors. In middle layers, P_{inter} rises sharply. The pivots begin to attend to non-pivot tokens, including those that early saliency-based pruning would remove. This outward attention is also visible as the middle-layer expansion band in Fig. 3b. In deep layers, P_{intra} increases again, suggesting that attention reconcentrates on enriched pivots for final task-specific reasoning.

We call this middle-layer outward retrieval **Visual Backtracking**. It shows that the identity of visual pivots can be stable while their information demand is not. Non-pivot tokens are not always useful enough to remain active across all layers, but they can still serve as context sources during a specific middle-layer window. In other words, a token can be computationally inactive in shallow layers without being informationally irrelevant.

This changes how we should view early pruning. Its main risk is not only that it may select the wrong tokens. It also deletes information before the model reaches the layer where that information would naturally be retrieved. Early pruning closes the retrieval route before it opens.

Pilot Study: Restoration timing matters.

If Visual Backtracking is important, then the timing of context restoration should matter. We test this with a controlled pilot study. We keep the same retention ratio and use the same non-pivot token features, but compare two ways of restoring them:

1. **Rhythm-agnostic fusion (Base)**: Features of removed tokens are merged into retained pivots before any LLM processing begins.

2. **Backtracking-aware injection (Ours)**: Pivot tokens first pass through shallow layers alone, and the cached context is injected only when the model enters the backtracking window.

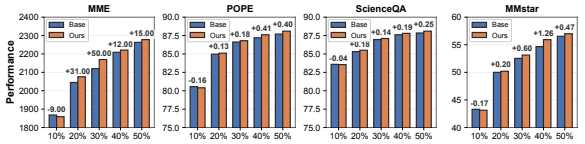


Figure 4: **Performance comparison of different feature integration paradigms.** Under identical retention ratios (10%–50%), backtracking-aware injection consistently outperforms rhythm-agnostic fusion at moderate ratios. At the extreme 10% retention rate, injecting into an undersized buffer triggers *Information Overload*, revealing the need for a multi-stage design.

As shown in Fig. 4, backtracking-aware injection gives consistent gains at moderate retention ratios, such as +50 on MME at 30% retention. Since both paradigms use the same token budget and the same non-pivot features, the gain does not come from keeping more information. It comes from restoring the information at the layer where the model naturally needs it.

The pilot study also shows a boundary case. At the extreme 10% retention ratio, delayed injection can slightly underperform early fusion. When the pivot buffer is too small, too much cached context is forced into too few receivers. This can cause **information overload** instead of useful restoration.

This suggests that visual token compression should not prune directly to the final budget at the beginning. A better design should use an intermediate buffer large enough for context absorption, then prune to a smaller reasoning core after the backtracking window. This leads to the design of **ReTrace**: cache non-pivot tokens instead of delet-

ing them, inject them when visual backtracking occurs, and prune only after their information has been absorbed.

4 Method

4.1 Overall Architecture

ReTrace decouples temporary computational deactivation from irreversible information deletion. Instead of making a one-shot keep/drop decision in shallow layers, it assigns visual tokens to three states: *active*, *dormant*, and *deleted*. Active tokens enter the LLM backbone as a compact pivot buffer. Dormant tokens bypass expensive layer-wise computation but remain cached for later restoration. Deleted tokens are removed only after their information has been absorbed.

As shown in Fig. 5, ReTrace follows a *cache–inject–prune* schedule. It first constructs a pivot buffer and caches the remaining tokens as dormant context. It then injects this context at the backtracking layer to approximate the missing pivot-to-context retrieval route. Finally, it prunes redundant enriched pivots for efficient final reasoning.

4.2 Pivot Buffer Construction

The first stage builds an intermediate receiver buffer, not the final reasoning core. Given a visual feature sequence $\mathcal{V}$ of length N and an intermediate retention ratio ρ_1, we set

$$K_1 = \lfloor \rho_1 N \rfloor. \tag{6}$$

The diagnostic pivots in Sec. 3 are computed from full-token LLM attention, which is useful for analysis but unavailable during compressed inference. We therefore construct a deployable encoder-level buffer using saliency and diversity cues. For implementation, we compute token saliency from selected vision-encoder attention layers. The buffer combines three subsets.

Global saliency keeps tokens with high encoder-level saliency scores, denoted by $s_g(v_i)$:

$$\mathcal{T}_g = \text{Top-}K_g(\{s_g(v_i) \mid v_i \in \mathcal{V}\}). \tag{7}$$

Local saliency preserves regional structures by selecting the most salient token in each spatial window $W_m \in \mathcal{W}$, using the local saliency score $s_l(v_i)$:

$$\mathcal{T}_l = \{\arg\max_{v_i \in W_m} s_l(v_i) \mid W_m \in \mathcal{W}\}. \tag{8}$$

Spatial diversity improves coverage through farthest-point sampling:

$$\mathcal{T}_d = \left\{ v^* \;\middle|\; v^* = \arg\max_{v_i \notin \mathcal{S}} \min_{v_j \in \mathcal{S}} \left(1 - \frac{v_i \cdot v_j^\top}{\|v_i\|\|v_j\|}\right) \right\}. \tag{9}$$

The final pivot buffer is

$$\mathcal{P} = \mathcal{T}_g \cup \mathcal{T}_l \cup \mathcal{T}_d, \quad |\mathcal{P}| = K_1. \tag{10}$$

Tokens outside the buffer are stored as dormant context,

$$\mathcal{D} = \mathcal{V} \setminus \mathcal{P}, \tag{11}$$

and bypass the LLM backbone until the injection stage.

4.3 Backtracking-Aware Context Injection

At the backtracking layer, ReTrace restores dormant context to approximate the pivot-to-context retrieval observed in full-token inference. Let b denote the injection layer, h_{b,p_k} the hidden state of the k-th pivot, and v_d the cached state of the d-th dormant token (after the projector, i.e., in the LLM hidden space). We compute normalized similarities:

$$\bar{h}_{b,p_k} = \frac{h_{b,p_k}}{\|h_{b,p_k}\|_2}, \quad \bar{v}_d = \frac{v_d}{\|v_d\|_2}, \quad S_{d,k} = \bar{v}_d \cdot \bar{h}_{b,p_k}^\top. \tag{12}$$

Each dormant token is assigned to its closest pivot:

$$a(d) = \arg\max_k S_{d,k}, \quad w_d = S_{d,a(d)}. \tag{13}$$

For each pivot, assigned dormant features are aggregated into a contextual increment:

$$\tilde{\Delta}_k = \frac{\sum_{d=1}^{|\mathcal{D}|} \mathbf{1}[a(d)=k]\, w_d v_d}{\sum_{d=1}^{|\mathcal{D}|} \mathbf{1}[a(d)=k]\, w_d + \epsilon}, \quad h_{b,p_k} \leftarrow h_{b,p_k} + \tilde{\Delta}_k. \tag{14}$$

The updated tokens are enriched pivots. They carry the compact active sequence forward while recovering part of the context that early pruning would remove.

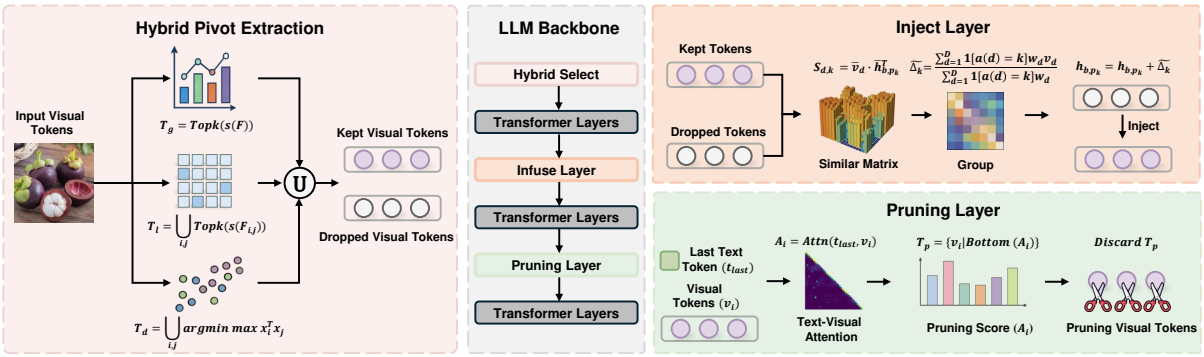


Figure 5: **Overall architecture of ReTrace.** ReTrace follows a *cache–inject–prune* schedule. **(1) Pivot Buffer Construction** builds a compact active receiver buffer at the encoder output while caching the remaining tokens as dormant context. **(2) Backtracking-Aware Context Injection** restores the missing pivot-to-context retrieval route at the backtracking layer through similarity-based routing. **(3) Post-Absorption Task Pruning** removes task-irrelevant enriched pivots after context absorption.

4.4 Post-Absorption Task Pruning

After context injection, ReTrace prunes redundant enriched pivots to reach the final budget. We use the last text token t_{last} as an instruction-aware query and score each enriched pivot p_i:

$$A_i = \text{Attn}(t_{last}, p_i). \tag{15}$$

Given the final retention ratio ρ_2, the final active token number is

$$K_2 = \lfloor \rho_2 K_1 \rfloor. \tag{16}$$

The pruning set contains the bottom $K_1 - K_2$ pivots:

$$\mathcal{T}_p = \text{Bottom-}(K_1 - K_2)(\{p_i \mid A_i\}). \tag{17}$$

Only the remaining K_2 enriched pivots are used for final reasoning. In this way, ReTrace prunes after context restoration rather than deleting raw context sources before they can be used.

5 Experiment

5.1 Experimental Settings

We evaluate our framework on four MLLMs (Qwen2.5-VL 7B/3B, InternVL3 8B/2B) across eight diverse vision-language benchmarks: ScienceQA (Lu et al., 2022), POPE (Li et al., 2023), MME (Fu et al., 2025), MMBench (Liu et al., 2024b), MMBench-CN (Liu et al., 2024b), MMStar (Chen et al., 2024b), MMERealWorld (MRW) (Zhang et al., 2025), and MMERealWorld-CN (MRW^{CN}) (Zhang et al., 2025). To demonstrate superiority, we comprehensively compare our approach against six recent state-of-the-art visual token compression baselines: FastV (Chen et al., 2024a), VisionZip (Yang et al., 2025), Dart (Wen et al., 2025), HoloV (Zou et al., 2025), VLMPruner (Wu et al., 2026), and Vscan (Zhang et al., 2026).

Across all evaluated models, we set the injection layer index to 8, the pruning layer index to 19, and the hybrid extraction quota ratio ($T_g : T_l : T_d$) to $1:1:2$. ρ_1 adaptively scales to match the target overall compression, while ρ_2 is strictly fixed at $1/3$. Detailed hyperparameter settings are provided in the supplementary material.

5.2 Main Results

Tabs. 1 - 4 present the quantitative evaluation of our method against existing visual token compression baselines across different models. Overall, under various compression ratios, our framework demonstrates highly competitive performance across most benchmarks and maintains the highest average performance retention. For instance, as shown in Tab. 1, on Qwen2.5-VL-7B, under the 22.2% and 11.1% visual token retention settings, our method preserves 90.7% and 84.4% of the original model's comprehensive performance, respectively (calculated by averaging the normalized scores against the full-token baseline across all benchmarks). Although it does not achieve absolute superiority on every specific sub-task, the overall performance across multiple datasets objectively validates the strong generality of our method. More results are provided in the Appendix.E.1.

Furthermore, Fig. 6a and 6b illustrate the performance degradation trends under continuous retention rate settings. Across multiple benchmarks, as the compression rate increases, our method maintains an overall leading position across all retention

Method	Venue	MMB	MMB^{CN}	MME	POPE	SQA$_I$	MMStar	MRW	MRW^{CN}	Avg.
Qwen2.5VL-7B	-	83.5	83.4	2347	87.7	88.1	63.9	57.4	50.2	100.0%
Token Reduction Setting: Retain 33.3% Tokens (↓66.7%)										
FastV (Chen et al., 2024a)	*ECCV'24*	75.7	78.8	2072	86.3	85.1	49.4	46.1	42.5	88.9%
VisionZip (Yang et al., 2025)	*CVPR'25*	83.2	80.7	2160	86.7	86.0	53.8	48.4	44.2	92.7%
Dart (Wen et al., 2025)	*EMNLP'25*	80.1	78.3	2166	85.1	84.3	48.9	45.7	39.0	88.6%
HoloV (Zou et al., 2025)	*NeurIPS'25*	78.3	78.3	2093	84.9	85.8	48.1	35.3	34.7	84.7%
VLMPruner (Wu et al., 2026)	*CVPR'26*	77.2	76.8	2235	86.1	85.7	46.3	44.5	41.7	88.5%
Vscan (Zhang et al., 2026)	*TMLR'26*	80.6	78.6	**2275**	87.2	86.4	49.4	46.6	43.5	91.3%
Ours	-	**83.5**	**81.7**	2240	**87.5**	**87.1**	**55.3**	**48.8**	**44.3**	**94.0%**
Token Reduction Setting: Retain 22.2% Tokens (↓77.8%)										
FastV (Chen et al., 2024a)	*ECCV'24*	74.9	77.8	2036	84.9	83.2	45.0	41.3	40.3	85.5%
VisionZip (Yang et al., 2025)	*CVPR'25*	81.6	79.4	2038	85.5	83.8	51.2	43.7	40.2	88.6%
Dart (Wen et al., 2025)	*EMNLP'25*	79.2	76.8	2090	82.8	83.4	46.6	40.7	36.5	85.2%
HoloV (Zou et al., 2025)	*NeurIPS'25*	76.5	75.6	2043	82.0	81.4	46.0	31.7	31.4	80.7%
VLMPruner (Wu et al., 2026)	*CVPR'26*	76.3	75.2	2091	84.7	83.1	45.2	41.2	40.3	85.5%
Vscan (Zhang et al., 2026)	*TMLR'26*	80.5	76.3	2135	86.1	85.8	47.4	44.1	41.3	88.5%
Ours	-	**81.7**	**80.8**	**2160**	**86.3**	**86.1**	**52.1**	**44.4**	**42.0**	**90.7%**
Token Reduction Setting: Retain 11.1% Tokens (↓88.9%)										
FastV (Chen et al., 2024a)	*ECCV'24*	69.2	74.9	1940	79.7	80.1	40.0	34.0	35.7	78.8%
VisionZip (Yang et al., 2025)	*CVPR'25*	**79.8**	76.8	1828	82.2	81.9	44.8	**36.5**	**37.5**	82.6%
Dart (Wen et al., 2025)	*EMNLP'25*	73.4	72.9	1877	78.1	81.6	40.5	32.9	32.1	77.7%
HoloV (Zou et al., 2025)	*NeurIPS'25*	72.4	73.8	2006	77.5	81.1	41.8	23.8	26.1	75.0%
VLMPruner (Wu et al., 2026)	*CVPR'26*	75.1	73.7	2015	80.2	81.1	42.3	35.7	34.9	80.7%
Vscan (Zhang et al., 2026)	*TMLR'26*	76.8	74.4	1990	81.3	**83.3**	42.4	33.2	32.5	80.3%
Ours	-	79.6	**77.4**	**2093**	**82.7**	82.3	**46.3**	36.4	37.2	**84.4%**

Table 1: **Performance comparisons on Qwen2.5VL-7B across 8 benchmarks.** The "Avg." metric denotes the mean relative performance retention across all datasets compared to the baseline.

Method	Venue	MMB	MMB^{CN}	MME	POPE	SQA$_I$	MMStar	MRW	MRW^{CN}	Avg.
InternVL3-8B	-	83.5	83.4	2415	90.7	98.2	63.9	62.0	60.8	100.0%
Token Reduction Setting: Retain 22.2% Tokens (↓77.8%)										
FastV (Chen et al., 2024a)	*ECCV'24*	**83.8**	83.0	**2324**	89.4	95.4	58.2	46.3	51.9	92.7%
VisionZip (Yang et al., 2025)	*CVPR'25*	80.3	82.0	2242	88.8	95.0	56.8	44.8	49.5	90.4%
HoloV (Zou et al., 2025)	*NeurIPS'25*	80.6	82.3	2201	88.7	95.4	57.0	44.7	50.0	90.5%
Ours	-	83.6	**84.3**	2319	**90.2**	**96.7**	**59.7**	**48.7**	**53.3**	**94.4%**
Token Reduction Setting: Retain 11.1% Tokens (↓88.9%)										
FastV (Chen et al., 2024a)	*ECCV'24*	79.5	78.7	2202	88.8	93.2	52.3	39.9	47.3	87.2%
VisionZip (Yang et al., 2025)	*CVPR'25*	78.7	76.8	1978	85.9	87.7	50.2	34.5	43.7	82.2%
HoloV (Zou et al., 2025)	*NeurIPS'25*	79.8	78.1	1964	87.9	89.4	51.5	36.1	44.2	83.9%
Ours	-	**80.9**	**80.7**	**2210**	**89.0**	**94.1**	**55.9**	**41.5**	**48.6**	**89.4%**

Table 2: **Performance comparisons on InternVL3-8B across 8 benchmarks.** The "Avg." metric denotes the mean relative performance retention across all datasets compared to the baseline.

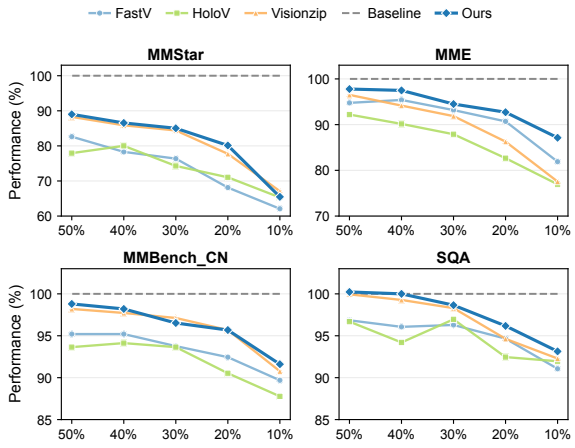
Method	MMB	MME	MMStar	MRW^{CN}	Avg.
Qwen2.5VL-3B	79.1	2157	55.9	45.1	100.0%
FastV (Chen et al., 2024a)	71.0	1698	36.6	32.2	76.3%
VisionZip (Yang et al., 2025)	**73.2**	1698	40.1	31.9	78.4%
Dart (Wen et al., 2025)	72.8	1640	36.8	29.4	74.8%
HoloV (Zou et al., 2025)	70.8	1673	38.6	25.1	72.9%
VLMPruner (Wu et al., 2026)	68.2	1735	36.0	26.7	72.6%
Vscan (Zhang et al., 2026)	69.6	1763	35.7	29.2	74.6%
Ours	72.3	**1781**	**40.3**	**32.5**	**79.5%**

Table 3: **Compact comparison on Qwen2.5VL-3B at 11.1% token retention.**

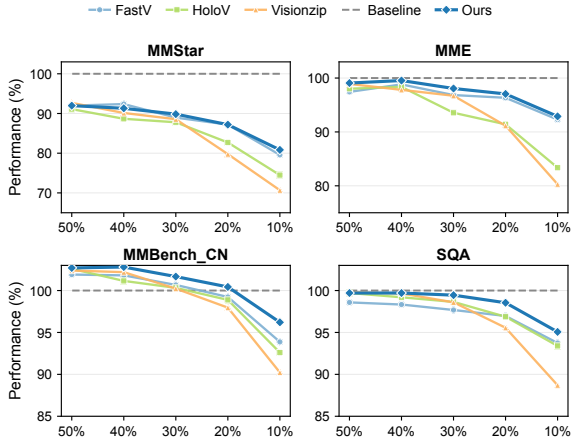
intervals. Compared to the significant performance drops observed in other baselines under extreme compression, our approach exhibits a smoother degradation trajectory. This further confirms the high practicality and robustness of our method in real-world deployments.

5.3 Ablation Results

To validate the core designs of **ReTrace**, we present two main ablations in the main paper: the effect of backtracking-aware injection and the choice of



(a) Qwen2.5-VL-7B.



(b) InternVL3-8B.

Figure 6: **Performance degradation trends across different token retention ratios.** We compare the degradation curves on Qwen2.5-VL-7B and InternVL3-8B under different visual token retention ratios.

injection layer. Additional ablations on pivot buffer construction are provided in the supplementary material.

First, we isolate the contribution of the injection mechanism. As shown in Fig. 7, adding backtracking-aware injection consistently improves performance over the no-injection variant across different benchmarks and retention ratios. This confirms that dormant context should not merely be cached, but restored when the model enters the backtracking window.

Second, we study where the injection should be placed. As shown in Fig. 8, injecting around layer 8 gives the most robust overall performance across benchmarks. Earlier injection provides context before the pivots have sufficiently formed, while later injection may miss the middle-layer retrieval window. This result supports our central claim that ReTrace benefits from restoring context at the layer where the model naturally looks back.

Method	MMB	MME	MMStar	MRW^{CN}	Avg.
InternVL3-2B	80.8	2221	60.7	52.1	100.0%
FastV (Chen et al., 2024a)	54.0	1398	36.4	25.9	59.9%
VisionZip (Yang et al., 2025)	67.7	1747	42.5	32.9	73.9%
HoloV (Zou et al., 2025)	70.1	1743	44.8	34.8	76.5%
Ours	**74.3**	**1965**	**47.9**	**37.1**	**82.6%**

Table 4: **Compact comparison on InternVL3-2B at 11.1% token retention.**

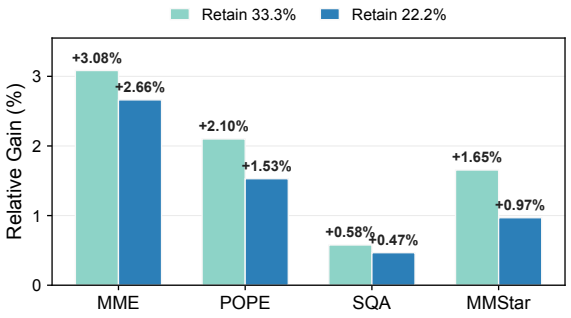


Figure 7: **Effect of backtracking-aware injection.** We report the relative gains brought by context injection under different retention ratios.

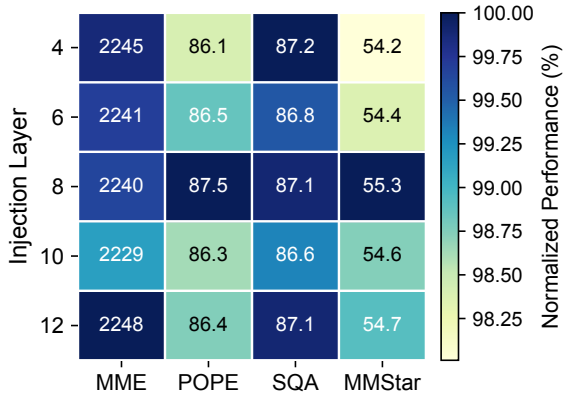


Figure 8: **Ablation on injection layer placement.** We evaluate different injection layers and observe the most balanced performance around the middle-layer backtracking window.

5.4 Efficiency Results

Tab. 5 details the hardware efficiency on Qwen2.5-VL-7B evaluated with FlashAttention. At the extreme 11.1% retention rate, our method reduces FLOPs and KV Cache by approximately 87%, achieving a 2.83× prefilling speedup. Despite this nearly 90% resource reduction, the model maintains a competitive MME score of 2093, demon-

Method	Tokens ↓	FLOPs ↓	KV ↓	MME ↑	Prefill Speedup ↑
Qwen2.5VL-7B	100%	1043.83	211.4 MB	**2347**	1.00×
+ Ours	22.2%	279.82	37.0 MB	2160	2.28×
+ Ours	**11.1%**	**148.06**	**26.7 MB**	2093	**2.83**×

Table 5: **Compact efficiency results on Qwen2.5VL-7B.** Runtime is evaluated on a single NVIDIA A100-80GB GPU with FlashAttention.

strating a highly practical accuracy-efficiency trade-off for MLLMs deployment.

5.5 Qualitative Results

As illustrated in Fig. 9, the qualitative results demonstrate that our method maintains highly robust reasoning capabilities even under significant token compression. More diverse qualitative examples are provided in the supplementary material.



User: *Describe this image.*

Ours: The image depicts a whimsical scene of three anthropomorphic animals sitting at a table, enjoying a tea party in a forest setting. The animal on the left appears to be a bear, the one in the middle is a cat, and the one on the right is a rabbit. They are all holding cups and saucers, and there are plates of cookies in front of them. The background features greenery, including grass and trees, suggesting they are in a natural, outdoor environment.

Figure 9: **Qualitative result at 11.1% visual-token retention.**

6 Conclusion

This paper identifies **Visual Backtracking** in MLLMs, showing that early low-saliency tokens can remain useful as middle-layer context sources. This finding explains why early pruning can harm complex reasoning: it treats temporary computational deactivation as irreversible information deletion. To address this, we propose **ReTrace**, a multi-stage compression framework that caches non-pivot tokens as dormant context, restores them when backtracking occurs, and prunes redundant enriched pivots only after context absorption. By reducing compute and memory overhead by nearly 90% while maintaining strong reasoning performance, ReTrace provides a practical solution for efficient long-sequence multimodal inference. This highlights the broader importance of adaptive token compression that follows the model's layer-wise information demand, rather than relying on fixed early-layer pruning decisions.

Limitations

ReTrace currently uses fixed layer indices for context injection and late-stage pruning. This fixed schedule can be suboptimal because the optimal restoration and pruning points may vary with input visual complexity and architecture-specific layer-wise dynamics. In addition, extreme compression can cause information overload when too much dormant context is injected into a small pivot buffer.

Ethical Considerations and Potential Risks

This work studies efficient inference for MLLMs and does not introduce new data involving human subjects, private information, or personally identifiable content. Our experiments use public benchmarks and publicly available model checkpoints. The main risk lies in deployment. Since ReTrace compresses visual tokens, it may change model outputs in some cases, especially under aggressive compression. Such changes should be carefully evaluated before using compressed MLLMs in high-stakes domains such as medical, legal, financial, or safety-critical applications. ReTrace is an efficiency technique and should not replace safety alignment, factuality checking, or human oversight. Faster inference may also reduce the cost of large-scale automated generation or visual analysis. This risk is shared by many model acceleration methods. Users should follow the licenses of the underlying models and datasets and apply proper safeguards in real-world deployment.

Acknowledgments

We thank the High-Performance Computing Center at Eastern Institute of Technology and Ningbo Institute of Digital Twin for providing the computing resources that supported this work. We also thank the anonymous reviewers for their constructive comments and valuable suggestions.

References

Ashwath Vaithinathan Aravindan, Abha Jha, and Mihir Kulkarni. 2025. Do vlms have bad eyes? diagnosing compositional failures via mechanistic interpretability. In *IEEE/CVF International Conference on Computer Vision, ICCV 2025 - Workshops, Honolulu, HI, USA, October 19-20, 2025*, pages 715–723. IEEE.

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*.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, and 45 others. 2025a. Qwen3-vl technical report. *Preprint*, arXiv:2511.21631.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, and 8 others. 2025b. Qwen2.5-vl technical report. *Preprint*, arXiv:2502.13923.

Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. 2023. Token merging: Your vit but faster. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. 2021. Emerging properties in self-supervised vision transformers. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 9630–9640. IEEE.

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 *Computer Vision - ECCV 2024 - 18th European Conference, Milan, Italy, September 29-October 4, 2024, Proceedings, Part LXXXI*, volume 15139 of *Lecture Notes in Computer Science*, 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 Feng Zhao. 2024b. Are we on the right way for evaluating large vision-language models? In *Advances in Neural Information Processing Systems*, volume 37, pages 27056–27087. Curran Associates, Inc.

Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, Lixin Gu, Xuehui Wang, Qingyun Li, Yiming Ren, Zixuan Chen, Jiapeng Luo, Jiahao Wang, Tan Jiang, Bo Wang, and 23 others. 2025. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. *Preprint*, arXiv:2412.05271.

Tri Dao. 2024. Flashattention-2: Faster attention with better parallelism and work partitioning. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

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.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An image is worth 16x16 words: Transformers for image recognition at scale. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, Yunsheng Wu, Rongrong Ji, Caifeng Shan, and Ran He. 2025. MME: A comprehensive evaluation benchmark for multimodal large language models. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025*.

Zhongyi Han, Gongxu Luo, Hao Sun, Yaqian Li, Bo Han, Mingming Gong, Kun Zhang, and Tongliang Liu. 2025. Alignclip: navigating the misalignments for robust vision-language generalization. *Mach. Learn.*, 114(3):58.

Shuo Jin, Siyue Yu, Bingfeng Zhang, Mingjie Sun, Yi Dong, and Jimin Xiao. 2025. Feature purification matters: Suppressing outlier propagation for training-free open-vocabulary semantic segmentation. In *IEEE/CVF International Conference on Computer Vision, ICCV 2025, Honolulu, HI, USA, October 19-25, 2025*, pages 20291–20300. IEEE.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th Symposium on Operating Systems Principles, SOSP 2023, Koblenz, Germany, October 23-26, 2023*, pages 611–626. ACM.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. 2023. Evaluating object hallucination in large vision-language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 292–305. Association for Computational Linguistics.

Yuxuan Liang, Xu Li, Xiaolei Chen, Haotian Chen, Yi Zheng, Zhe Liu, Rui Zhu, Bin Li, and Xiangyang Xue. 2026. Pyramid token pruning for high-resolution large vision-language models via region, token, and instruction-guided importance. *IEEE Trans. Circuits Syst. Video Technol.*, 36(6):8372–8384.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2024a. Improved baselines with visual instruction tuning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pages 26286–26296. IEEE.

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, Kai Chen, and Dahua Lin. 2024b. Mmbench: Is your multi-modal model an all-around player? In *Computer Vision – ECCV 2024: 18th European Conference, Milan, Italy, September 29–October 4, 2024, Proceedings, Part VI*, page 216–233, Berlin, Heidelberg. Springer-Verlag.

Pan Lu, Swaroop Mishra, Tony 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. In *Advances in Neural Information Processing Systems*.

Yu Meng, Kaiyuan Li, Chenran Huang, Chen Gao, Xinlei Chen, Yong Li, and Xiaoping Zhang. 2025. Plphp: Per-layer per-head vision token pruning for efficient large vision-language models. *arXiv:2502.14504*.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR.

Yuzhang Shang, Mu Cai, Bingxin Xu, Yong Jae Lee, and Yan Yan. 2025. Llava-prumerge: Adaptive token reduction for efficient large multimodal models. In *IEEE/CVF International Conference on Computer Vision, ICCV 2025, Honolulu, HI, USA, October 19-25, 2025*, pages 22857–22867. IEEE.

Xudong Tan, Peng Ye, Chongjun Tu, Jianjian Cao, Yaoxin Yang, Lin Zhang, Dongzhan Zhou, and Tao Chen. 2025. Tokencarve: Information-preserving visual token compression in multimodal large language models. *arXiv preprint arXiv:2503.10501*.

Pavan Kumar Anasosalu Vasu, Fartash Faghri, Chun-Liang Li, Cem Koc, Nate True, Albert Antony, Gokula Santhanam, James Gabriel, Peter Grasch, Oncel Tuzel, and Hadi Pouransari. 2025. Fastvlm: Efficient vision encoding for vision language models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11-15, 2025*, pages 19769–19780. Computer Vision Foundation / IEEE.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. 2024. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*.

Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, Zhaokai Wang, Zhe Chen, Hongjie Zhang, Ganlin Yang, Haomin Wang, Qi Wei, Jinhui Yin, Wenhao Li, Erfei Cui, and 56 others. 2025. Internvl3.5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *Preprint*, arXiv:2508.18265.

Zichen Wen, Yifeng Gao, Shaobo Wang, Junyuan Zhang, Qintong Zhang, Weijia Li, Conghui He, and Linfeng Zhang. 2025. Stop looking for "important tokens" in multimodal language models: Duplication matters more. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 9961–9980, Suzhou, China. Association for Computational Linguistics.

Zhenkai Wu, Xiaowen Ma, Zhenliang Ni, Dengming Zhang, Han Shu, Xin Jiang, and Xinghao Chen. 2026. Vlm-pruner: Buffering for spatial sparsity in an efficient vlm centrifugal token pruning paradigm. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 31952–31961.

Long Xing, Qidong Huang, Xiaoyi Dong, Jiajie Lu, Pan Zhang, Yuhang Zang, Yuhang Cao, Conghui He, Jiaqi Wang, Feng Wu, and Dahua Lin. 2025. Conical visual concentration for efficient large vision-language models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville,*

TN, USA, June 11-15, 2025, pages 14593–14603. Computer Vision Foundation / IEEE.

Senqiao Yang, Yukang Chen, Zhuotao Tian, Chengyao Wang, Jingyao Li, Bei Yu, and Jiaya Jia. 2025. Visionzip: Longer is better but not necessary in vision language models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11-15, 2025*, pages 19792–19802. Computer Vision Foundation / IEEE.

Hao Yin, Guangzong Si, and Zilei Wang. 2025. Lifting the veil on visual information flow in mllms: Unlocking pathways to faster inference. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11-15, 2025*, pages 9382–9391. Computer Vision Foundation / IEEE.

Ce Zhang, Kaixin Ma, Tianqing Fang, Wenhao Yu, Hongming Zhang, Zhisong Zhang, Haitao Mi, and Dong Yu. 2026. Vscan: Rethinking visual token reduction for efficient large vision-language models. *Transactions on Machine Learning Research*.

Yifan Zhang, Huanyu Zhang, Haochen Tian, Chaoyou Fu, Shuangqing Zhang, Junfei Wu, Feng Li, Kun Wang, Qingsong Wen, Zhang Zhang, Liang Wang, and Rong Jin. 2025. Mme-realworld: Could your multimodal LLM challenge high-resolution real-world scenarios that are difficult for humans? In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

Shiyu Zhao, Zhenting Wang, Felix Juefei-Xu, Xide Xia, Miao Liu, Xiaofang Wang, Mingfu Liang, Ning Zhang, Dimitris N. Metaxas, and Licheng Yu. 2025. Accelerating multimodal large language models by searching optimal vision token reduction. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11-15, 2025*, pages 29869–29879. Computer Vision Foundation / IEEE.

Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, Zhangwei Gao, Erfei Cui, Xuehui Wang, Yue Cao, Yangzhou Liu, Xingguang Wei, Hongjie Zhang, Haomin Wang, Weiye Xu, and 32 others. 2025. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. *Preprint*, arXiv:2504.10479.

Xin Zou, Di Lu, Yizhou Wang, Yibo Yan, Yuanhuiyi Lyu, Xu Zheng, Linfeng Zhang, and Xuming Hu. 2025. Don't just chase "highlighted tokens"in mllms: Revisiting visual holistic context retention. In *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pages 39800–39832. Curran Associates, Inc.

A Overview

This supplementary material provides additional details and results to support the main manuscript. The contents are organized as follows.

We first describe the experimental configurations in Sec. B and present the pseudocode of our method in Sec. C. Next, we include additional motivation experiments in Sec. D, followed by more experimental results in Sec. E, including extended quantitative evaluations across different model scales. Further ablation studies are provided in Sec. F.

We also analyze the theoretical complexity and empirical latency overhead of our method in Sec. G. In addition, we visualize the pivot buffer construction in Sec. H and provide more qualitative examples in Sec. I. Finally, we report details of the benchmarks and MLLMs in Secs. J and K, declare the use of AI assistance in Sec. L.

B Settings Details

Across InternVL-8B and Qwen2.5VL-7B, the injection and pruning layer indices are empirically set to 8 and 19, respectively. The hybrid extraction quota ratio ($T_g : T_l : T_d$) is configured to $1 : 1 : 2$. To match the target overall compression, ρ_1 scales adaptively while ρ_2 remains strictly fixed at $1/3$. Additionally, for local saliency extraction, we employ a 4×4 grid window to effectively capture regional geometric variations without introducing excessive computational overhead. The specific combinations of ρ_1 and ρ_2 for various target average retention ratios are detailed in Tab. 6.

Global saliency is computed from the final vision-encoder layer for all models. For local saliency, we use a model-specific layer choice. In Qwen2.5-VL, local saliency is computed from the 6-th vision-encoder layer to better preserve regional structures. In InternVL3, we use the final vision-encoder layer, since its tiling-based high-resolution processing already maintains local details within each tile. All models are evaluated with single-run greedy decoding, with sampling disabled.

C Algorithm

We provide the detailed pseudocode of **ReTrace** in Alg. 1.

D More Motivation Experiment

We include several additional analyses to better understand Visual Backtracking. The first part checks

Average Retention Ratio	ρ_1	ρ_2
11.1%	13.7%	33.3%
22.2%	27.4%	33.3%
33.3%	41.17%	33.3%

Table 6: Hyperparameter configurations for different target compression levels. The late-stage retention ρ_2 is strictly fixed, while the mid-stage retention ρ_1 adaptively scales to meet the overall average retention ratio.

whether the pattern remains stable when the pivot ratio changes. The second part gives more visual examples of the backtracking process. The last part uses 3D feature visualization to show how visual tokens are organized across layers. Together, these results support the main observation: MLLMs do not simply keep using the same visual anchors, but revisit broader visual context in the middle layers.

D.1 Robustness Across Pivot Ratios

We further examine whether the observed backtracking pattern depends on the choice of pivot ratio. In the main analysis, pivots are defined by selecting the top-ranked visual tokens according to text-guided saliency. Here, we repeat the same analysis under top 10%, 15%, and 20% pivot ratios.

Fig. 10 shows that pivot tokens remain highly stable across layers for all three settings. This indicates that the model forms persistent visual anchors early, and this conclusion is not tied to a specific top-k threshold.

Fig. 11 visualizes the attention from pivot tokens to all visual tokens. Across different pivot ratios, we consistently observe a middle-layer expansion band, where pivots attend more broadly to non-pivot visual regions. Fig. 12 further confirms this trend: pivot-to-non-pivot attention peaks in the middle layers and then decreases in deeper layers.

These results show that Visual Backtracking is robust to the pivot ratio. The phenomenon is therefore unlikely to be an artifact of a particular top-k choice. Instead, it reflects a stable layer-wise pattern: visual pivots first consolidate, then retrieve context from non-pivot regions, and finally reconcentrate for task-specific reasoning.

D.2 Visualization of the Visual Backtracking Phenomenon

To provide a concrete illustration of the visual backtracking mechanism, we visualize the layer-wise attention dynamics. In 13 and 14, the left panels highlight the identified top-k pivot tokens on the input images, while the right matrices display the attention heatmaps from these pivots to all visual tokens across 28 transformer layers. A distinct structural shift is observable throughout the network depth. In both the shallow and deep layers, the attention remains highly concentrated on the highly salient tokens. However, in the intermediate layers, the attention distribution undergoes a drastic spatial expansion, actively querying a broader set of seemingly unimportant tokens. This layer-wise observation directly validates that mid-layer computations heavily rely on broad contextual layouts for visual reasoning, structurally precluding the aggressive pruning of these less salient tokens at early stages.

D.3 3D Visualization of Feature Manifold

To evaluate the representational integrity of different token reduction strategies, we project the hidden states of visual tokens from intermediate layers into a 3D space using t-SNE. As shown in Fig. 15, the full token set establishes the reference distribution manifold. Baseline methods, including global top selection and pre-merging, exhibit structural deviations from this reference, frequently resulting in localized clusters that fail to span the original feature space. In contrast, the application of intermediate token injection yields a spatial distribution that consistently covers the reference manifold across most observed layers. This indicates that the injection mechanism preserves the topological structure of the visual features, preventing the representational collapse observed in standard pruning approaches.

E More Results

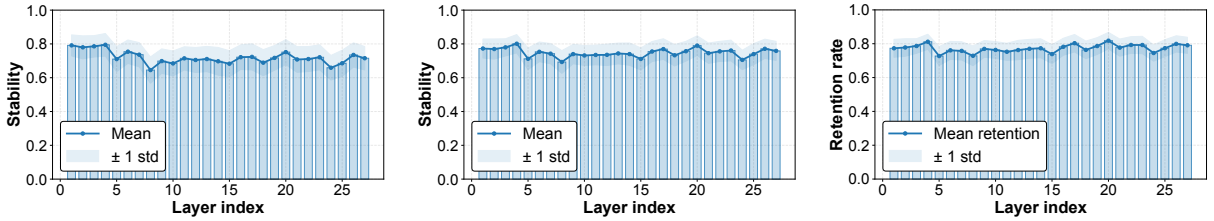
This section extends our quantitative evaluations across diverse parameter scales, presenting detailed performance metrics for both smaller MLLMs (2B/3B) and a larger 32B regime.

E.1 More Main Results

We provide here the results of the experimental models used in the main text under additional compression-ratio settings, as shown in the Tab. 7, Tab. 8 and Tab. 9.

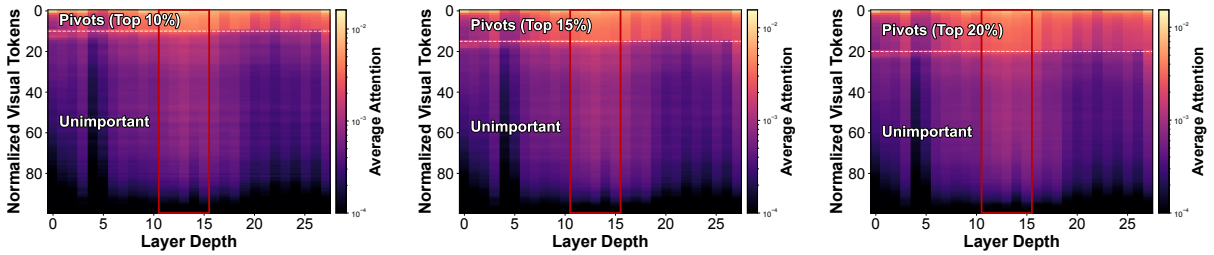
E.2 More Quantitative Results on Smaller MLLMs

In this section, we provide extended quantitative evaluations on Qwen2.5VL-3B and InternVL3-2B



(a) Top 10% (b) Top 15% (c) Top 20%

Figure 10: **Cross-layer spatial stability of pivot tokens under varying pivot ratios.** The cross-layer stability of pivot tokens remains consistently high across different top-k ratio.



(a) Top 10% (b) Top 15% (c) Top 20%

Figure 11: **Attention heatmap depicting the backtracking band.** The attention heatmap from pivot tokens to all visual tokens consistently reveals a distinct vertical bright band in the middle layers, regardless of the **pivot ratio**.

to complement the main manuscript. As detailed in Tab. 8 and Tab. 9, we report the performance across multiple token retention settings.

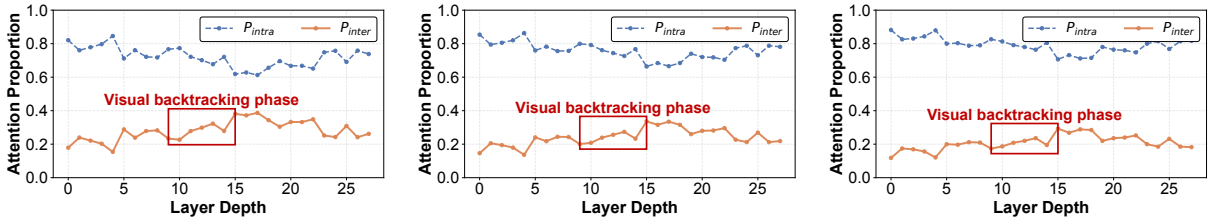
The comprehensive results further demonstrate the robust superiority of ReTrace. Across all compression scales, our method consistently outperforms recent state-of-the-art baselines including FastV, VisionZip, and Vscan. The performance gap becomes particularly evident under highly restricted token budgets. For instance, on the InternVL3-2B model at an 11.1% retention rate, ReTrace achieves an average score of 82.5%, exceeding the strongest baseline (HoloV) by +5.9%. This consistent margin across diverse architectures and sparsity levels confirms that our hybrid extraction strategy effectively balances inference acceleration and reasoning capacity, independent of the model's parameter scale.

E.3 Scalability to Larger Regimes (32B)

To validate the scalability of our approach, we further extend the evaluation to the 32B-parameter regime using Qwen2.5-VL-32B. As reported in Tab. 10, Tab. 11, and Tab. 12, our method consistently outperforms recent visual-token reduction baselines, including FastV (Chen et al., 2024a), PDrop (Xing et al., 2025), and Vscan (Zhang et al., 2026), across all tested retention ratios and benchmarks. The advantage is especially clear under aggressive compression. At the extreme 11% retention ratio, our method achieves 82.21 on MMBench, 80.67 on MMBench-CN, and 2037 on MME, outperforming the strongest baseline Vscan by +4.12, +3.44, and +22 points, respectively. The gains become even more pronounced on MME at moderate compression levels: at 22% and 33% retention, our method surpasses Vscan by +117 and +85 points, indicating that the proposed cache–inject–prune schedule better preserves the visual evidence needed by large-scale MLLMs. Moreover, when the retention ratio reaches 50%, our method nearly matches the uncompressed 100% upper bound, with only a 0.52-point drop on MMBench, a 1.04-point drop on MMBench-CN, and an 18-point drop on MME. These results suggest that the proposed multi-stage mechanism remains effective and robust when scaled to massive MLLMs, maintaining strong accuracy while substantially reducing the visual token budget.

E.4 Comparison with Additional Compression Methods

The baselines selected in the main experiments already cover representative visual-token compression paradigms. For completeness, we further include additional methods under different retention ratios. As shown in Tab. 13, ReTrace continues to maintain strong overall performance across diverse



(a) Top 10% (b) Top 15% (c) Top 20%

Figure 12: **Temporal evolution of normalized attention proportions.** The temporal evolution shows an abrupt "Visual Backtracking" peak (orange curve) where pivot tokens strongly query the background across different **top-k ratio settings**.

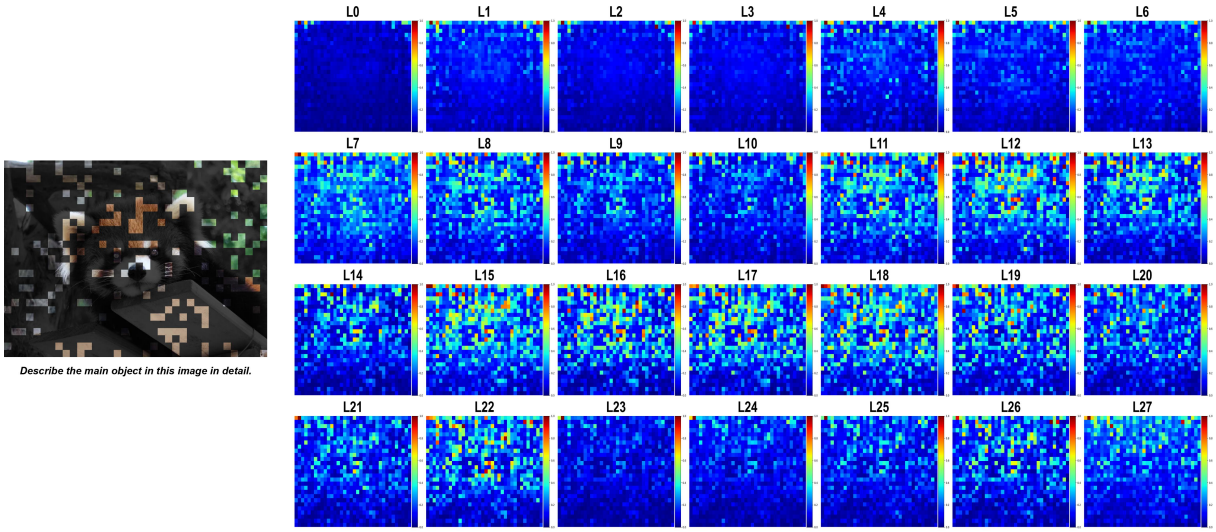


Figure 13: **Layer-wise visualization of the visual backtracking phenomenon.** We present the attention heatmaps from pivot tokens to all visual tokens across all transformer layers. The visualization clearly demonstrates a distinct dynamic: while attention remains highly localized on main subjects in the shallow and deep layers, it undergoes a dramatic spatial expansion in the mid-layers to actively query seemingly unimportant background tokens.

benchmarks.

F More Ablation Results

We present detailed ablation studies analyzing the optimal layer selection and p_2.

F.1 Effect of Layer Selection.

To determine the optimal stage for executing intermediate layer pruning, as shown in Tab. 16, we conduct an ablation study across different layers. Pruning too early results in a notable performance drop. This occurs because the model requires a sufficient number of layers to digest and model the initial injection of visual tokens; truncating the sequence before these features are fully processed leads to the loss of critical contextual clues. Conversely, delaying the pruning to deeper layers yields sub-optimal outcomes, which can be attributed to the over-homogenization of spatial features. The results indicate that layer 19 provides the optimal balance, ensuring adequate digestion of injected visual information while avoiding feature homogenization.

F.2 Impact of p_2.

We further investigate the intermediate pruning coefficient p_2, which dictates the proportion of tokens retained at the intermediate stage. In this ablation, the global token retention budget is strictly fixed at 11.1%. Setting $p_2 = 1/3$ yields the optimal performance. A smaller coefficient drops too many tokens at the intermediate stage, causing an irreversible loss of spatial and structural details. Conversely, a larger coefficient retains an excessive number of tokens intermediately.

G Efficiency

This section comprehensively evaluates the computational efficiency of our method, providing both a

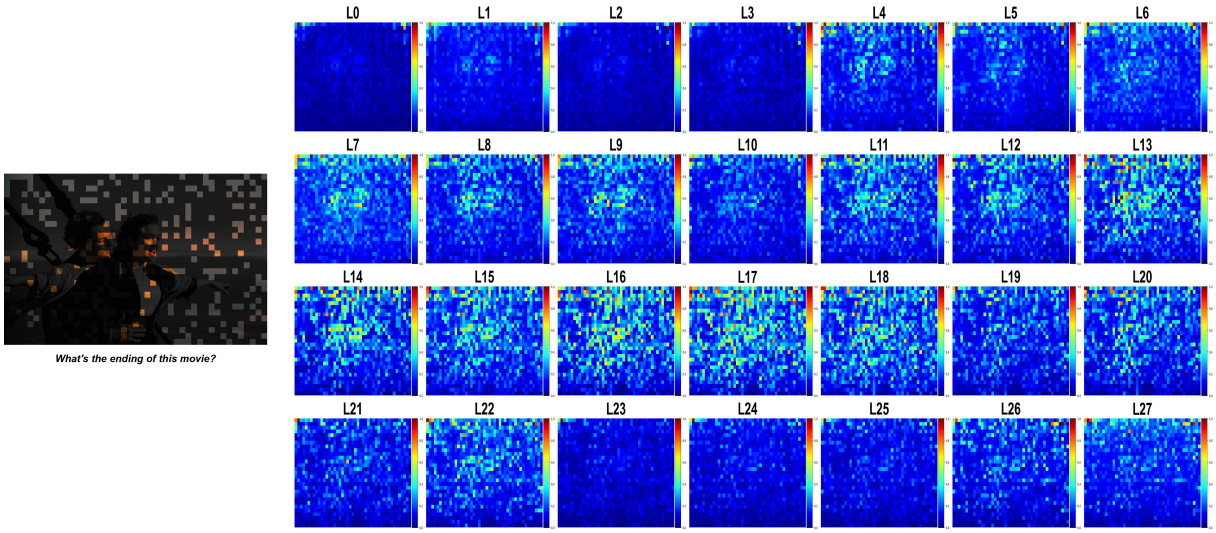


Figure 14: **Layer-wise visualization of the visual backtracking phenomenon.**

Method	Venue	MMB	MMB^{CN}	MME	POPE	SQA$_I$	MMStar	MRW	MRW^{CN}	Avg.
InternVL3-8B	-	83.5	83.4	2415	90.7	98.2	63.9	62.0	60.8	100.0%
Token Reduction Setting: Retain 33.3% Tokens (↓66.7%)										
FastV (Chen et al., 2024a)	*ECCV'24*	84.2	84.3	2347	90.0	96.2	59.2	50.2	54.7	95.1%
VisionZip (Yang et al., 2025)	*CVPR'25*	81.3	84.3	2290	89.4	97.1	60.5	50.8	53.8	94.2%
HoloV (Zou et al., 2025)	*NeurIPS'25*	82.4	84.2	2285	89.7	97.3	60.6	50.2	53.8	94.4%
Ours	-	**85.1**	**85.7**	**2351**	**90.5**	**97.5**	**61.2**	**53.0**	**55.9**	**96.8%**
Token Reduction Setting: Retain 22.2% Tokens (↓77.8%)										
FastV (Chen et al., 2024a)	*ECCV'24*	**83.8**	83.0	**2324**	89.4	95.4	58.2	46.3	51.9	92.7%
VisionZip (Yang et al., 2025)	*CVPR'25*	80.3	82.0	2242	88.8	95.0	56.8	44.8	49.5	90.4%
HoloV (Zou et al., 2025)	*NeurIPS'25*	80.6	82.3	2201	88.7	95.4	57.0	44.7	50.0	90.5%
Ours	-	83.6	**84.3**	2319	**90.2**	**96.7**	**59.7**	**48.7**	**53.3**	**94.4%**
Token Reduction Setting: Retain 11.1% Tokens (↓88.9%)										
FastV (Chen et al., 2024a)	*ECCV'24*	79.5	78.7	2202	88.8	93.2	52.3	39.9	47.3	87.2%
VisionZip (Yang et al., 2025)	*CVPR'25*	78.7	76.8	1978	85.9	87.7	50.2	34.5	43.7	82.2%
HoloV (Zou et al., 2025)	*NeurIPS'25*	79.8	78.1	1964	87.9	89.4	51.5	36.1	44.2	83.9%
Ours	-	**80.9**	**80.7**	**2210**	**89.0**	**94.1**	**55.9**	**41.5**	**48.6**	**89.4%**

Table 7: **Performance comparisons on InternVL3-8B across 8 benchmarks.**

theoretical complexity analysis and an empirical latency profiling of the key algorithmic components.

G.1 Theoretical Efficiency Analysis

We provide a theoretical analysis to show the efficiency gain of **ReTrace**. Let N be the number of input visual tokens, L the number of LLM layers, and D the hidden dimension. Dense inference processes all visual tokens through all layers. Its attention and MLP cost can be written as

$$\mathcal{O}_{\text{base}} = L \cdot \mathcal{O}(N^2 D + N D^2). \tag{18}$$

The KV cache memory scales as

$$\mathcal{M}_{\text{base}} = \mathcal{O}(LND). \tag{19}$$

ReTrace reduces the active visual sequence in two stages. Given the intermediate retention ratio ρ_1, the first active length is

$$K_1 = \rho_1 N. \tag{20}$$

After post-absorption pruning with retention ratio ρ_2, the final active length becomes

$$K_2 = \rho_2 K_1 = \rho_1 \rho_2 N. \tag{21}$$

Let p denote the late pruning layer. From layer 1 to $p-1$, ReTrace uses K_1 active visual tokens. From layer p to L, it uses K_2 active visual tokens. The total computational cost is therefore

$$\begin{aligned}\mathcal{O}_{\text{ReTrace}} = {} & (p-1) \cdot \mathcal{O}(K_1^2 D + K_1 D^2) \\ & + (L - p + 1) \cdot \mathcal{O}(K_2^2 D + K_2 D^2) \\ & + \mathcal{O}_{\text{overhead}}.\end{aligned} \tag{22}$$

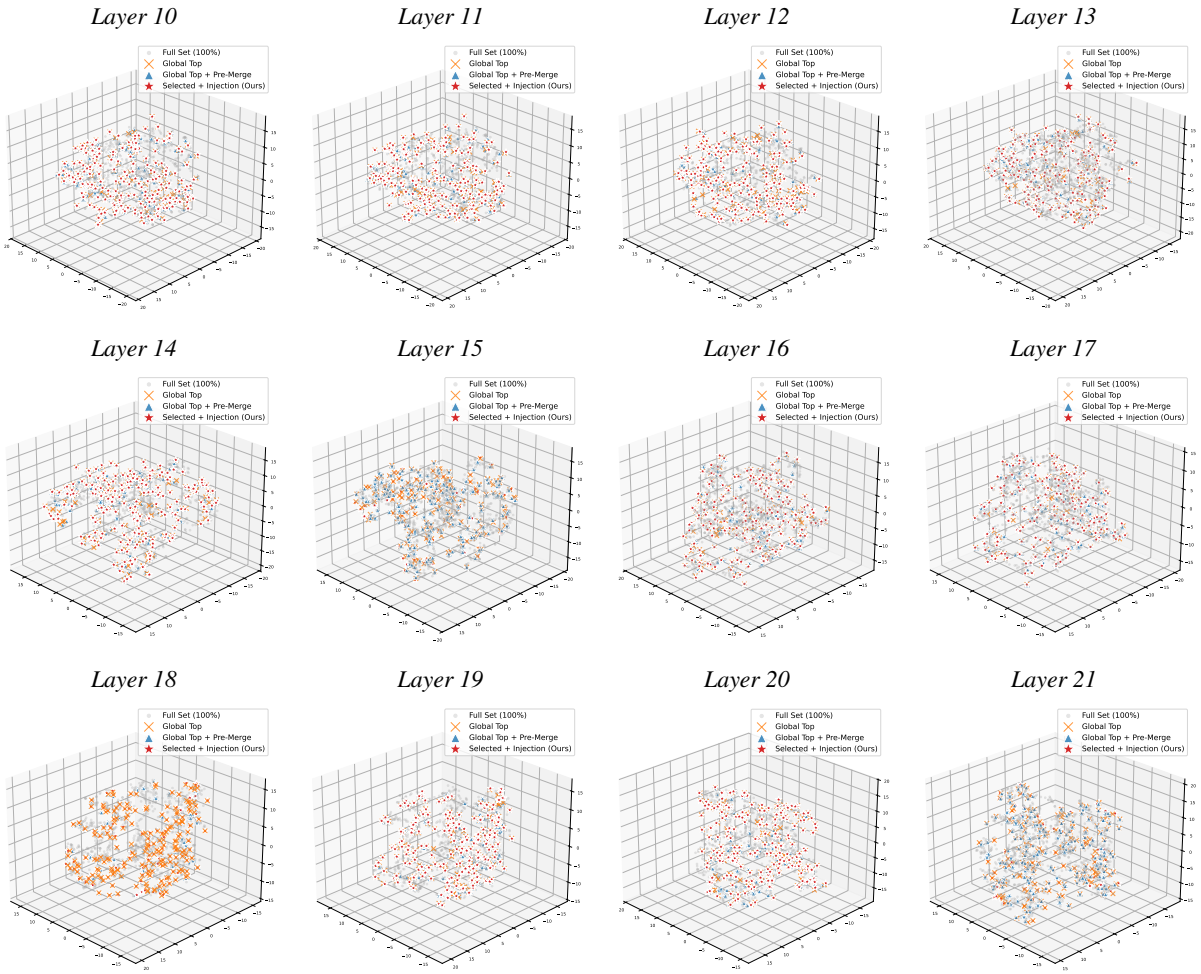


Figure 15: **Evolution of 3D feature representations across intermediate layers.**

The overhead comes from three one-time operations. Pivot buffer construction is performed at the encoder output. Context injection computes similarities between dormant tokens and active pivots, with cost

$$\mathcal{O}\big((N-K_1)K_1D\big). \tag{23}$$

Post-absorption pruning uses a single text-to-visual query, with cost

$$\mathcal{O}(K_1D). \tag{24}$$

These operations are executed once rather than at every LLM layer, so they are small compared with repeated dense self-attention in deep models.

When the quadratic attention term dominates, the theoretical FLOPs ratio can be approximated as

$$\begin{aligned}\eta_{\text{FLOPs}} &\approx \frac{(p-1)K_1^2+(L-p+1)K_2^2}{LN^2}\\ &= \rho_1^2\left[\frac{p-1}{L}+\frac{L-p+1}{L}\rho_2^2\right].\end{aligned} \tag{25}$$

Similarly, the KV cache ratio is

$$\begin{aligned}\eta_{\text{KV}} &= \frac{(p-1)K_1+(L-p+1)K_2}{LN}\\ &= \rho_1\left[\frac{p-1}{L}+\frac{L-p+1}{L}\rho_2\right].\end{aligned} \tag{26}$$

This analysis shows that ReTrace obtains a compounding reduction from both ρ_1 and ρ_2. In particular, the attention cost is suppressed by the quadratic factor ρ_1^2, and further reduced after late-stage pruning through ρ_2^2. This matches our empirical efficiency results, where aggressive compression yields about 87% FLOPs reduction and a large KV-cache saving.

G.2 Overhead of Buffer Construction and Context Injection

We profile the additional runtime cost introduced by **ReTrace**, including spatial-diversity buffer construction and backtracking-aware context injection. The profiling is conducted on the MME benchmark

Method	Venue	MMB	MMB^{CN}	MME	POPE	SQA$_I$	MMStar	MRW	MRW^{CN}	Avg.
Qwen2.5VL-3B	-	79.1	78.1	2157	87.3	81.0	55.9	53.1	45.1	100.0%
Token Reduction Setting: Retain 33.3% Tokens (↓66.7%)										
FastV (Chen et al., 2024a)	*ECCV'24*	73.7	73.4	2026	84.2	80.3	44.4	39.2	36.7	88.9%
VisionZip (Yang et al., 2025)	*CVPR'25*	76.0	73.6	1964	84.2	79.8	**48.0**	40.2	37.7	90.2%
Dart (Wen et al., 2025)	*EMNLP'25*	75.3	73.7	1950	82.8	78.9	36.8	40.3	36.0	86.7%
HoloV (Zou et al., 2025)	*NeurIPS'25*	76.1	74.0	2015	83.5	81.2	44.1	36.3	33.5	87.7%
VLMPruner (Wu et al., 2026)	*CVPR'26*	75.8	73.2	1991	83.9	81.1	45.7	39.7	35.2	89.1%
Vscan (Zhang et al., 2026)	*TMLR'26*	72.8	72.5	1977	82.7	80.1	41.2	34.3	33.5	85.3%
Ours	-	**76.4**	**74.2**	**2091**	**85.5**	**82.4**	47.5	**41.4**	**39.0**	**92.2%**
Token Reduction Setting: Retain 22.2% Tokens (↓77.8%)										
FastV (Chen et al., 2024a)	*ECCV'24*	72.8	71.9	1918	82.3	80.2	42.1	34.7	33.4	85.1%
VisionZip (Yang et al., 2025)	*CVPR'25*	75.1	71.8	1795	83.4	**81.2**	44.0	37.7	35.2	86.7%
Dart (Wen et al., 2025)	*EMNLP'25*	74.9	70.6	1804	80.0	78.9	36.8	36.3	33.6	83.3%
HoloV (Zou et al., 2025)	*NeurIPS'25*	74.7	71.9	1795	81.1	79.6	41.7	32.7	29.7	82.9%
VLMPruner (Wu et al., 2026)	*CVPR'26*	73.2	71.1	1935	79.2	79.7	41.2	34.1	32.7	84.1%
Vscan (Zhang et al., 2026)	*TMLR'26*	70.6	68.5	1838	80.9	79.7	40.4	32.6	32.4	82.3%
Ours	-	**75.7**	**72.5**	**2003**	**83.6**	81.1	**45.7**	**38.0**	**36.6**	**89.0%**
Token Reduction Setting: Retain 11.1% Tokens (↓88.9%)										
FastV (Chen et al., 2024a)	*ECCV'24*	71.0	69.9	1698	79.0	79.0	36.6	30.8	32.2	80.1%
VisionZip (Yang et al., 2025)	*CVPR'25*	**73.2**	70.0	1698	**81.0**	79.8	40.1	31.3	31.9	81.7%
Dart (Wen et al., 2025)	*EMNLP'25*	72.8	65.8	1640	73.8	78.9	36.8	29.6	29.4	77.6%
HoloV (Zou et al., 2025)	*NeurIPS'25*	70.8	69.0	1673	76.4	77.5	38.6	26.1	25.1	76.6%
VLMPruner (Wu et al., 2026)	*CVPR'26*	68.2	66.7	1735	74.1	73.9	36.0	27.5	26.7	75.4%
Vscan (Zhang et al., 2026)	*TMLR'26*	69.6	63.6	1763	76.2	78.1	35.7	28.8	29.2	77.2%
Ours	-	72.3	**70.6**	**1781**	77.4	**80.0**	**40.3**	**32.2**	**32.5**	**82.1%**

Table 8: **Performance comparisons on Qwen2.5VL-3B across 8 benchmarks.** The "Avg." metric denotes the mean relative performance retention across all datasets compared to the baseline.

with 2,374 test samples. As shown in Tab. 19, under the 11.1% retention setting, the two operations take only 0.7 ms and 0.1 ms per sample, accounting for 0.42% and 0.06% of the total latency. Across all evaluated sparsity levels, their combined overhead stays below 0.5%. This confirms that ReTrace adds negligible runtime overhead while preserving the practical speedup from visual token compression.

H Visualization of Pivot Buffer Construction

To better understand the pivot buffer used in **ReTrace**, we visualize the retained token maps under different sparsity levels. As shown in Fig. 16, the constructed buffer consistently covers the main visual subjects while preserving spatially distributed context regions, even under aggressive compression. This is important for ReTrace: the buffer is not intended to be the final reasoning core, but to provide active receivers for later dormant-context restoration.

Our buffer combines global saliency (T_g), local saliency (T_l), and spatial diversity (T_d). These components serve different roles. T_g anchors dominant semantic regions, T_l preserves local structures, and T_d improves scene coverage. Together, they reduce the risk that dormant context will be routed into an overly narrow set of receivers during the injection stage.

We further compare the default allocation ($T_g : T_l : T_d = 1 : 1 : 2$) with several ablation variants. As shown in Fig. 18 - Fig. 24, isolated strategies show clear failure modes. Using only global saliency (1:0:0) over-concentrates tokens on salient foreground objects and loses background layout. Using only spatial diversity (0:0:1) spreads tokens widely but misses core semantic subjects. Partial combinations, such as removing spatial diversity (1:1:0) or local saliency (1:0:1), still lead to weaker coverage or less stable local detail preservation. The default 1:1:2 allocation offers a more balanced spatial-semantic buffer, which supports the later backtracking-aware injection step.

I More Qualitative Results

We provide additional qualitative examples from LLaVA-Bench to evaluate the generation quality in open-ended scenarios. Aggressive token reduction often discards critical visual details, leading to object omissions or fabricated hallucinations in

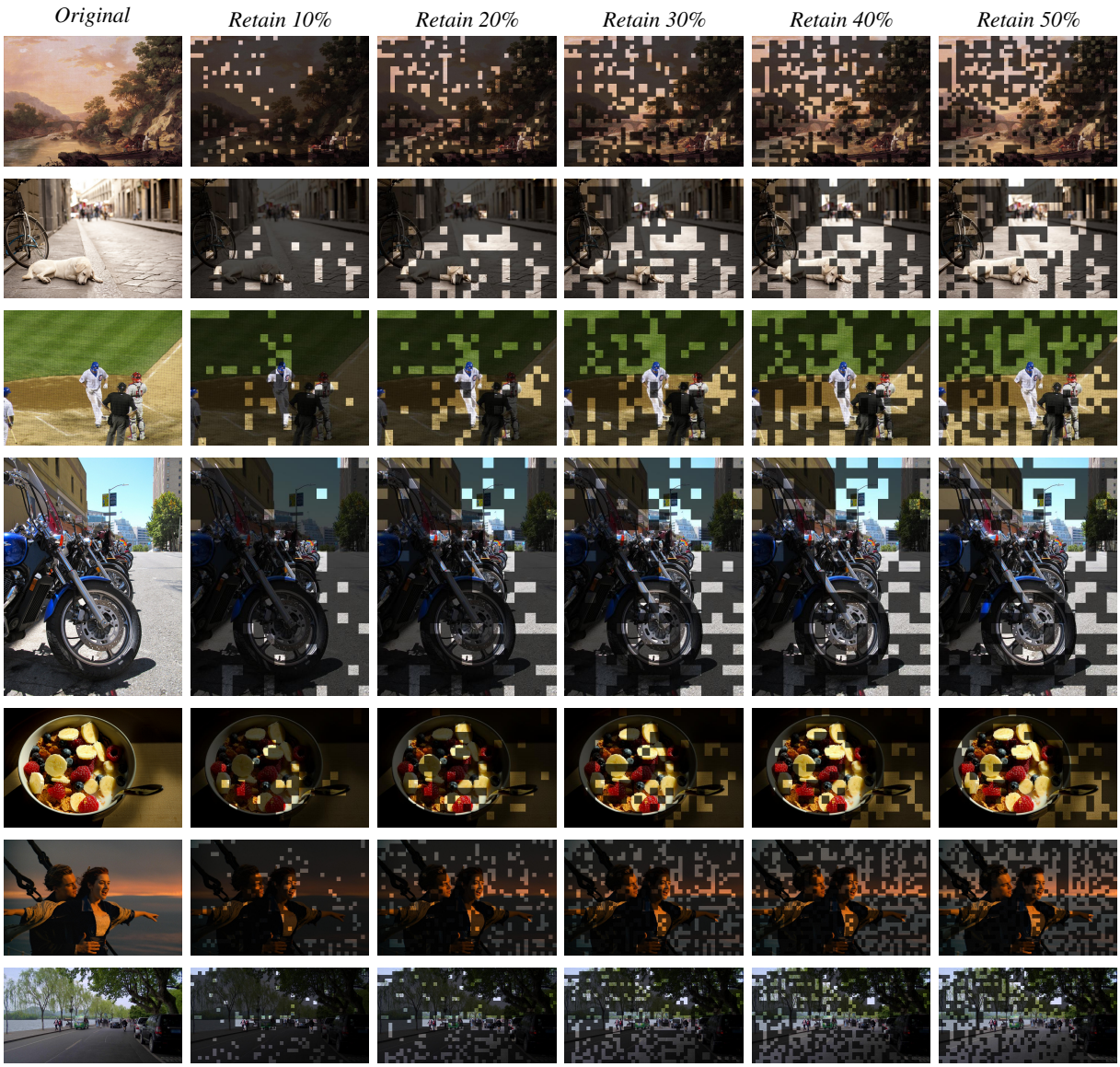


Figure 16: **Visualization of pivot buffer construction under different sparsity levels.** As the token retention ratio increases from 10% to 50%, the buffer progressively covers more visual regions while consistently preserving core semantic objects, local structures, and distributed context. This balanced receiver set supports later backtracking-aware context injection.

Method	Venue	MMB	MMB^{CN}	MME	POPE	SQA$_I$	MMStar	MRW	MRW^{CN}	Avg.
InternVL3-2B	-	80.8	79.0	2221	89.9	95.7	60.7	53.8	52.1	100.0%
Token Reduction Setting: Retain 33.3% Tokens (↓66.7%)										
FastV (Chen et al., 2024a)	*ECCV'24*	67.2	65.5	1570	75.3	79.6	43.5	25.5	33.2	73.3%
VisionZip (Yang et al., 2025)	*CVPR'25*	78.9	76.3	2041	87.2	93.2	51.8	33.5	41.9	88.6%
HoloV (Zou et al., 2025)	*NeurIPS'25*	79.2	76.0	1997	88.0	92.9	52.8	34.2	42.6	88.9%
Ours	-	**80.6**	**76.9**	**2085**	**89.2**	**94.2**	**53.7**	**37.1**	**44.9**	**91.5%**
Token Reduction Setting: Retain 22.2% Tokens (↓77.8%)										
FastV (Chen et al., 2024a)	*ECCV'24*	63.7	61.8	1472	70.6	77.6	40.5	21.5	29.5	68.3%
VisionZip (Yang et al., 2025)	*CVPR'25*	75.6	73.2	1949	85.2	90.5	49.2	28.7	38.4	83.9%
HoloV (Zou et al., 2025)	*NeurIPS'25*	76.0	66.7	1934	85.9	90.9	50.0	29.4	39.2	83.5%
Ours	-	**78.9**	**75.5**	**2072**	**88.1**	**93.5**	**53.5**	**32.2**	**41.2**	**88.7%**
Token Reduction Setting: Retain 11.1% Tokens (↓88.9%)										
FastV (Chen et al., 2024a)	*ECCV'24*	54.0	50.8	1398	64.7	74.5	36.4	15.6	25.9	60.3%
VisionZip (Yang et al., 2025)	*CVPR'25*	67.7	66.3	1747	79.4	84.4	42.5	19.7	32.9	74.1%
HoloV (Zou et al., 2025)	*NeurIPS'25*	70.1	66.7	1743	81.8	85.8	44.8	22.6	34.8	76.6%
Ours	-	**74.3**	**71.7**	**1965**	**84.4**	**89.8**	**47.9**	**27.5**	**37.1**	**82.5%**

Table 9: **Performance comparisons on InternVL3-2B across 8 benchmarks.** The "Avg." metric denotes the mean relative performance retention across all datasets compared to the baseline.

Model	Method	11%	22%	33%	50%	100%
32B	FastV	70.13	78.09	81.01	85.40	87.37
	PDrop	73.22	79.04	82.74	86.03	87.37
	Vscan	78.09	82.47	85.31	86.17	87.37
	Ours	**82.21**	**85.48**	**86.42**	**86.85**	**87.37**

Table 10: **MMBench accuracy** under different visual token retention ratios. Higher is better.

the generated text. As illustrated in Fig. 17, ReTrace effectively mitigates this challenge by preserving spatial diversity and mid-layer structural features, thereby preventing the degradation of visual grounding. The generated responses demonstrate accurate comprehension of complex scenes without fabricating ungrounded contents. These results indicate that our method achieves significant inference acceleration while maintaining a text-visual consistency strictly comparable to the unpruned baseline, without exacerbating hallucination issues.

J Benchmark Details

We provide a detailed description of benchmarks here.

ScienceQA (Lu et al., 2022) is a benchmark focused on multimodal question answering in the scientific domain. Code use MIT License, Datasets use CC BY-NC-SA 4.0.

POPE (Li et al., 2023) is designed to quantitatively evaluate the object hallucination phenomenon in MLLMs. The testing task is strictly formulated as a binary "Yes/No" classification, forcing the model to determine whether a specific object exists in the image. In terms of scale, thousands of test samples are typically constructed for different source datasets. The samples are generated using three negative sampling strategies: random, popular, and adversarial, to accurately detect whether the model tends to fabricate non-existent visual content based on linguistic priors. MIT License.

MME (Fu et al., 2025) is a comprehensive benchmark for evaluating the perception and cognition abilities of MLLMs. The task format is uniformly designed as concise binary "Yes/No" questions to minimize the interference of prompt engineering. The benchmark includes 14 independent subtasks, categorized into perception (covering object detection, OCR, color recognition, etc.) and cognition (covering commonsense reasoning, code reasoning, mathematical calculation, etc). To prevent training data leakage, its instruction-answer pairs are constructed and verified entirely through rigorous manual annotation. Academic research only.

MMBench (Liu et al., 2024b) aims to evaluate the all-around capabilities of MLLMs at a fine-grained level. The task format consists of single-

Model	Method	11%	22%	33%	50%	100%
32B	FastV	67.61	76.12	79.81	83.42	86.00
	PDrop	71.31	77.65	80.72	84.28	86.00
	Vscan	77.23	80.76	84.45	84.78	86.00
	Ours	**80.67**	**83.93**	**84.53**	**84.96**	**86.00**

Table 11: **MMBench-CN accuracy** under different visual token retention ratios. Higher is better.

Model	Method	11%	22%	33%	50%	100%
32B	FastV	1536	1924	2124	2316	2432
	PDrop	1536	1797	2078	2279	2432
	Vscan	2015	2189	2288	2378	2432
	Ours	**2037**	**2306**	**2373**	**2414**	**2432**

Table 12: **MME total scores** under different visual token retention ratios. Higher is better.

choice questions, covering 20 specific visual and cognitive capability dimensions, such as object localization, spatial relations, and social reasoning. Apache-2.0 license.

MMBench-CN (Liu et al., 2024b) is the Chinese branch of the MMBench evaluation framework. It is specifically used to independently test the cross-modal comprehension and reasoning performance of multimodal models in a Chinese linguistic context. Apache-2.0 license.

MMStar (Chen et al., 2024b) is an evaluation benchmark specifically designed to test the visual dependency of MLLMs. The task format comprises multiple-choice questions, containing 1,500 high-quality samples manually filtered to ensure that the questions cannot be answered using purely text-based instructions. The test categories cover 6 core capabilities and 18 detailed evaluation dimensions. The benchmark introduces two data metrics: multi-modal gain and multi-modal leakage, which are used to quantitatively detect data leakage issues during the pre-training phase of MLLMs. Academic research only.

MMERealWorld (Zhang et al., 2025) is an evaluation benchmark targeting high-resolution real-world scenarios. The primary evaluation formats are multiple-choice and objective question-answering, with testing scenarios covering 43 complex real-world categories to challenge the limits of fine-grained visual recognition. The dataset contains 13,366 high-resolution captured images equipped with 29,429 manually annotated question-answer pairs by professionals, focusing on testing the parsing performance of MLLMs in actual, complex, and high-definition environments. MME-RealWorld is only used for academic research.

MMERealWorld-CN (Zhang et al., 2025) is the Chinese branch of MMERealWorld. MME-RealWorld-CN is only used for academic research.

K Model Card

We provide a detailed description of MLLMs here.

Qwen2.5-VL (Bai et al., 2025b) adopts dynamic-resolution visual processing with a native Vision Transformer and window attention, enabling it to handle images of varying sizes and aspect ratios. It also supports visual grounding by producing points or bounding boxes for object localization. The model is further optimized for structured document understanding, including forms, tables, charts, diagrams, and layouts, and extends to long-video comprehension with second-level temporal localization. Apache-2.0 License.

InternVL3 (Zhu et al., 2025) follows the "ViT-MLP-LLM" paradigm and introduces a native multimodal pre-training strategy, where multimodal data and large-scale textual corpora are jointly optimized within a single pre-training stage. It incorporates Variable Visual Position Encoding (V2PE), which uses smaller and more flexible position increments for visual tokens to better support long multimodal contexts. MIT License.

L AI Use

During the preparation of this work, the authors used AI tools solely to improve the linguistic expression and grammatical correctness of the manuscript. After using this service, the authors

Retention	Method	MMB	MMB-CN	MME	POPE	SQA	MMStar	MRW	MRW-CN	Avg.
22.2%	HiMap	78.3	77.2	1935	83.7	84.2	48.3	40.9	39.5	85.7%
22.2%	PDrop	79.7	79.4	2017	83.3	85.3	49.3	42.3	39.4	87.2%
22.2%	G-Search	80.3	78.9	2103	84.6	83.9	50.5	41.8	39.9	87.9%
22.2%	**ReTrace**	**81.7**	**80.8**	**2160**	**86.3**	**86.1**	**52.1**	**44.4**	**42.0**	**90.7%**
11.1%	HiMap	77.2	75.1	1847	79.9	81.1	43.3	34.2	33.1	79.7%
11.1%	PDrop	76.9	75.5	1932	80.2	81.2	44.0	35.0	34.9	81.0%
11.1%	G-Search	77.5	74.9	2021	80.1	80.7	44.6	35.4	34.5	81.5%
11.1%	**ReTrace**	**79.6**	**77.4**	**2093**	**82.7**	**82.3**	**46.3**	**36.4**	**37.2**	**84.4%**

Table 13: **Comparison under different visual-token retention ratios.** ReTrace consistently outperforms representative visual-token compression methods across both retention settings.

Layer Index	MME	POPE	SQA	MMStar
4	2245	86.1	**87.2**	54.2
6	2241	86.5	86.8	54.4
8 (Ours)	2240	**87.5**	87.1	**55.3**
10	2229	86.3	86.6	54.6
12	**2248**	86.4	87.1	54.7

(a) Injection layer index.

$T_g : T_l : T_d$	MME	POPE	SQA	MMStar
1:1:0	2232	87.4	86.9	**55.5**
1:0:1	2249	85.8	86.5	54.4
1:1:2 (Ours)	2240	**87.5**	87.1	55.3
1:0:0	2244	87.4	87.2	54.4
0:0:1	**2279**	87.4	**87.5**	54.9

(b) Pivot extraction strategies.

Table 14: **Ablation studies on injection layer placement and pivot extraction strategies (Qwen2.5-VL-7B).**

reviewed and edited the content as needed and take full responsibility for the technical accuracy and original contributions of the publication.

Method	Retention: 33.3%				Retention: 22.2%			
	MME	POPE	SQA	MMStar	MME	POPE	SQA	MMStar
w/o inject	2173	85.7	86.6	54.4	2104	85.0	85.7	51.6
***w/* inject (Ours)**	**2240**	**87.5**	**87.1**	**55.3**	**2160**	**86.3**	**86.1**	**52.1**

Table 15: **Ablation study on the injection mechanism.**



Q: Describe this photo in detail.

A: The image shows a group of mangosteen fruits, which are known for their distinctive appearance and flavor. The outer skin of the mangosteen is a deep purple color, while the interior is segmented with a white flesh that has a unique star-like pattern when cut open. The fruit in the foreground is cut in half, revealing its inner structure, which consists of several segments with a white, fleshy texture. The outer skin appears to be smooth and slightly bumpy, typical of mangosteens.

No Hallucination.



Q: Which iconic movie scene is being parodied in the meme?

A: The meme appears to parody the iconic presentation scene from Disney's "The Lion King," in which Rafiki lifts the young Simba into the air. In this version, Elon Musk is shown holding up a dog in a similar pose, creating a humorous visual reference to the original scene.

No Hallucination.



Q: Describe this painting in detail.

A: This is a famous painting titled "Mona Lisa" by Leonardo da Vinci, one of the most recognizable and enigmatic works of art in the world. The painting depicts a woman who is believed to be Mona Lisa di Giocondo, though her identity remains uncertain. The subject is shown in a three-quarter view, with her head slightly tilted to the right and her gaze directed slightly upwards and to the left, creating an air of mystery and intrigue. The woman's expression is often described as enigmatic, with a slight smile that some interpret as a hint of amusement or even a knowing look.

No Hallucination.



Q: What is the name of the character in the image?

A: The character in the image is Conan Edogawa, also known as Shinichi Kudo, from the anime series "Detective Conan" (also known as "Case Closed").

No Hallucination.

Figure 17: **More qualitative results on retain 22.2% visual tokens.** Using LLaVA-bench, we have demonstrated the effectiveness of our method on OpenQA, where the model's hallucination-free responses fully showcase the validity of our method.

Layer Index	MME	MMStar
9	1786	41.9
11	1789	41.8
13	1800	41.6
15	2006	44.3
17	2053	45.7
19 (Ours)	**2093**	**46.3**
21	2064	43.4

Table 16: **Ablation study on layer selection.** Performance on MME and MMStar benchmarks when applying our extraction strategy at different intermediate layers.

p_2	MME	SQA_I	MMStar
1/5	2054	80.9	44.9
1/4	2069	81.4	43.3
1/3	**2093**	**82.3**	**46.3**
1/2	2050	82.2	43.4

Table 17: **Ablation study on the intermediate layer pruning p_2.** The total token retention budget is strictly fixed at 11.1%. Setting the coefficient p_2 to 1/3 yields the optimal balance across multiple benchmarks.

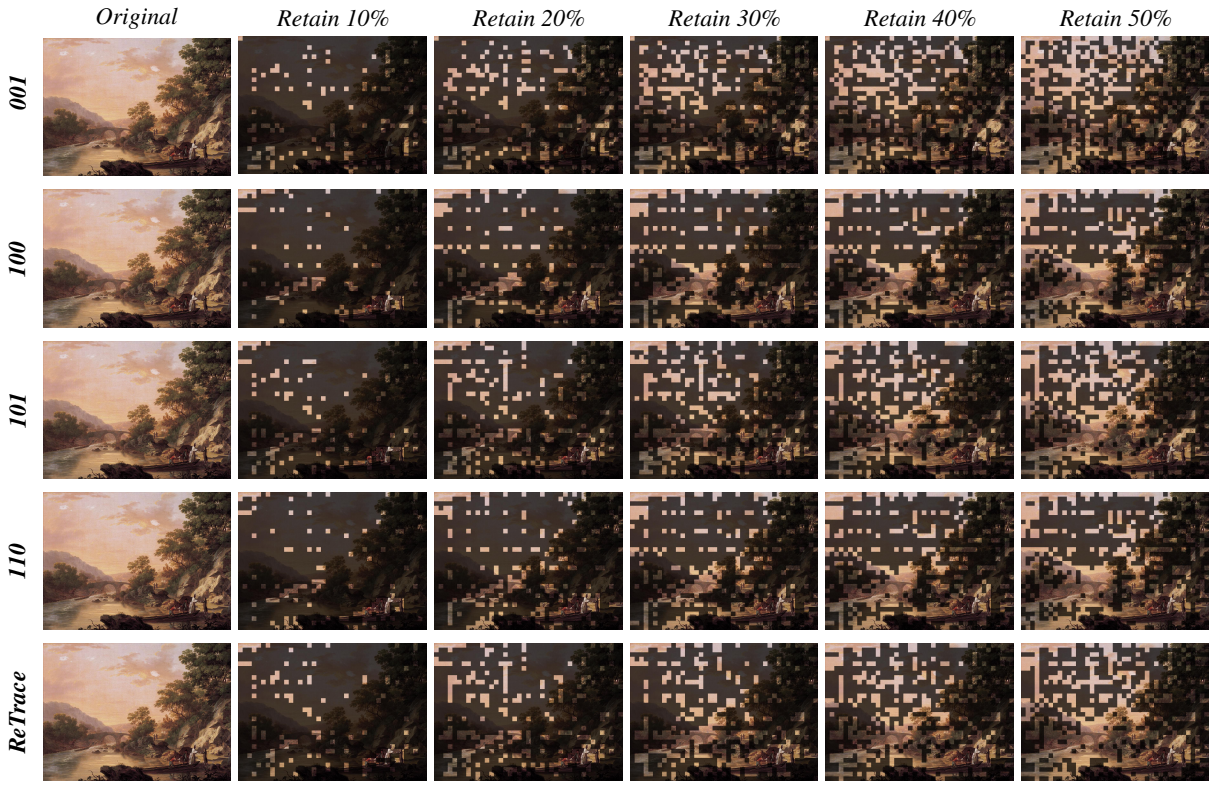


Figure 18: **Qualitative comparison.**

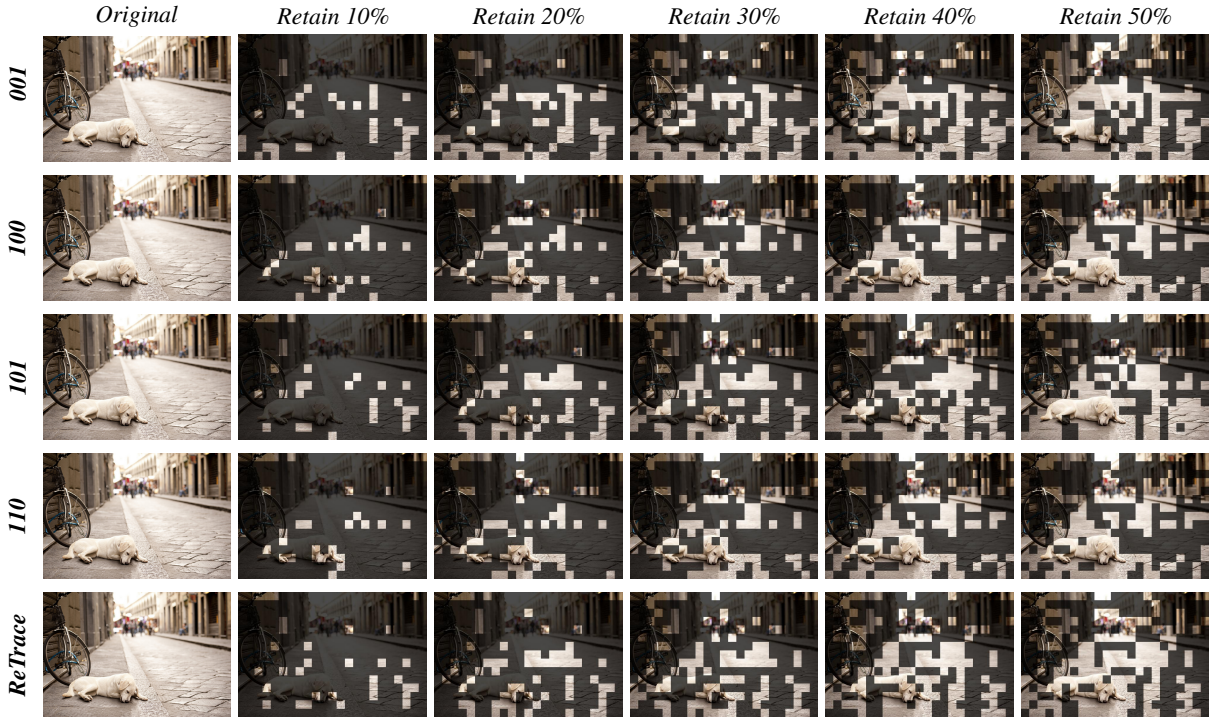


Figure 19: **Qualitative comparison.**

Method	Tokens ↓	Total Time ↓ (Sec)	Prefilling Time ↓ (Sec)	FLOPs ↓	KV Cache ↓ (MB)	MME ↑ (Score)	Speedup ↑ Total	Speedup ↑ Prefilling
Qwen2.5VL-7B	100%	557.443	239.833	1043.83	211.4	**2347**	1.00×	1.00×
+ Ours	22.2%	401.919	105.234	279.82	37.0	2160	1.38×	2.28×
+ Ours	**11.1%**	**373.961**	**84.809**	**148.06**	**26.7**	2093	**1.49**×	**2.83**×

Table 18: **Efficiency results on Qwen2.5VL-7B.** Performance, runtime, FLOPs, and memory overhead are evaluated on a single NVIDIA A100-80GB GPU with FlashAttention.

Tokens ↓	Total Time ↓ (Sec)	T_d Extraction Total (s)	T_d Extraction Avg. (ms)	Mid-stage Injection Total (s)	Mid-stage Injection Avg. (ms)	Overhead Proportion ↓ T_d Ext.	Overhead Proportion ↓ Inject
33.3%	426.85	1.78	0.8	0.20	0.1	0.42%	0.05%
22.2%	401.92	1.66	0.7	0.21	0.1	0.41%	0.05%
11.1%	373.96	1.58	0.7	0.23	0.1	0.42%	0.06%

Table 19: Algorithmic overhead analysis under different token retention settings on Qwen2.5-VL-7B. The absolute execution times and their relative proportions explicitly demonstrate that the spatial diversity extraction (T_d) and mid-stage injection introduce negligible latency to the overall inference.

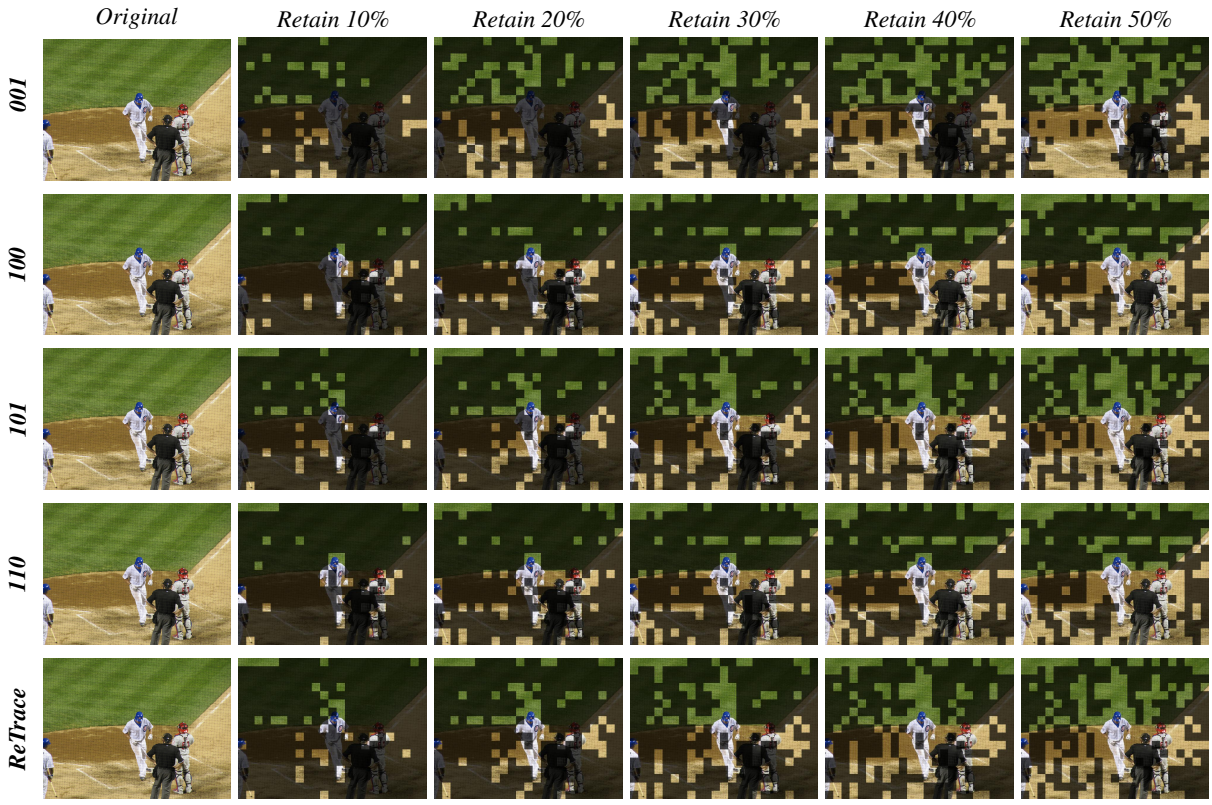


Figure 20: **Qualitative comparison.**

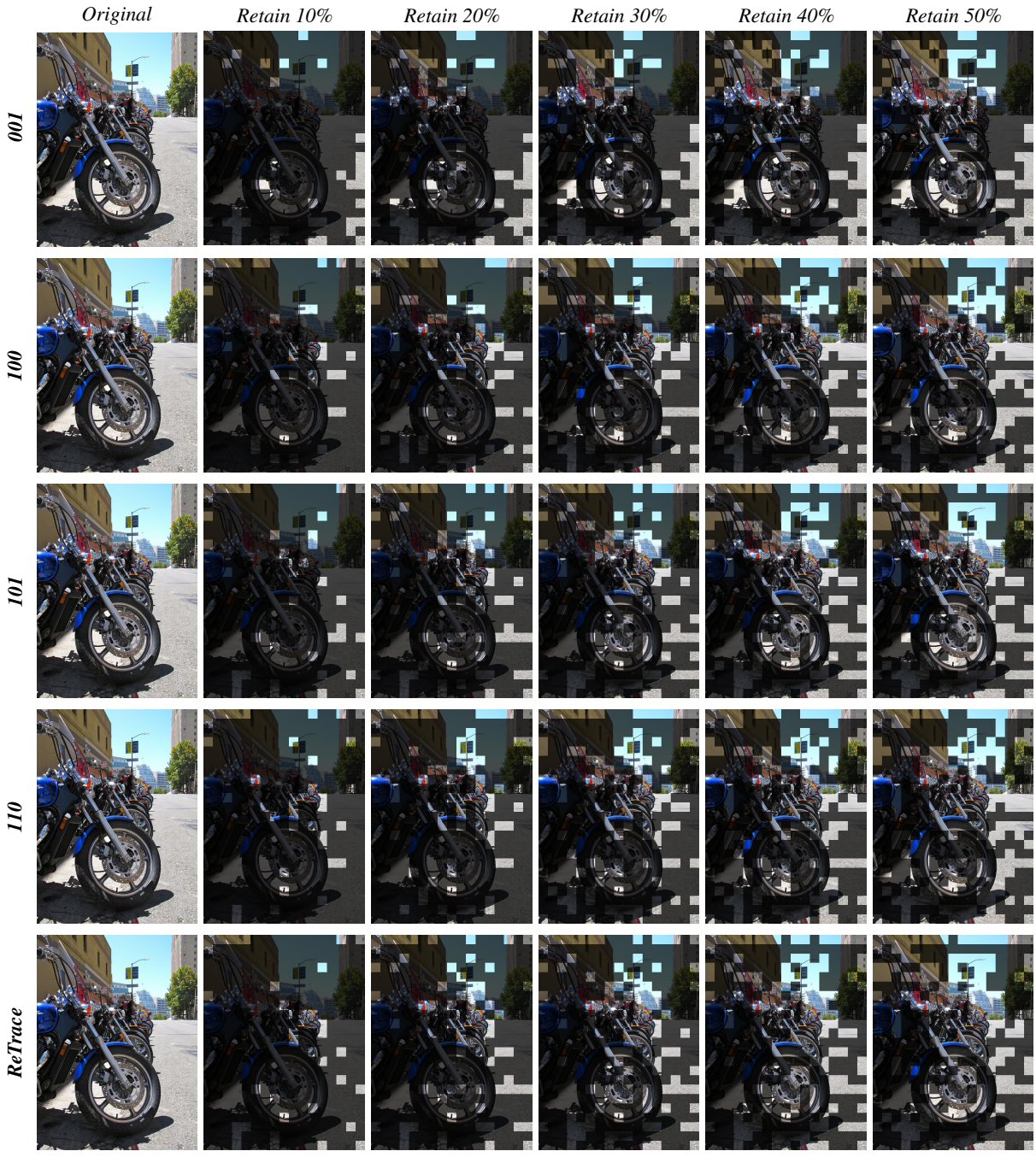


Figure 21: **Qualitative comparison.**

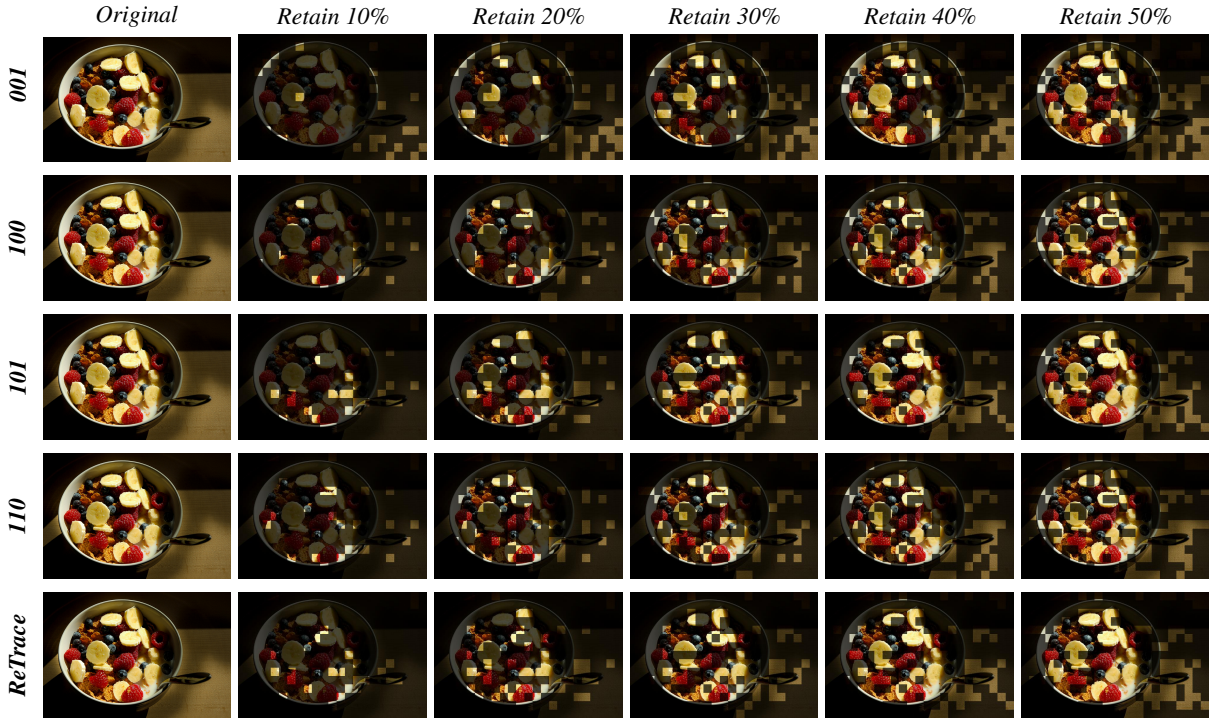


Figure 22: **Qualitative comparison.**



Figure 23: **Qualitative comparison.**

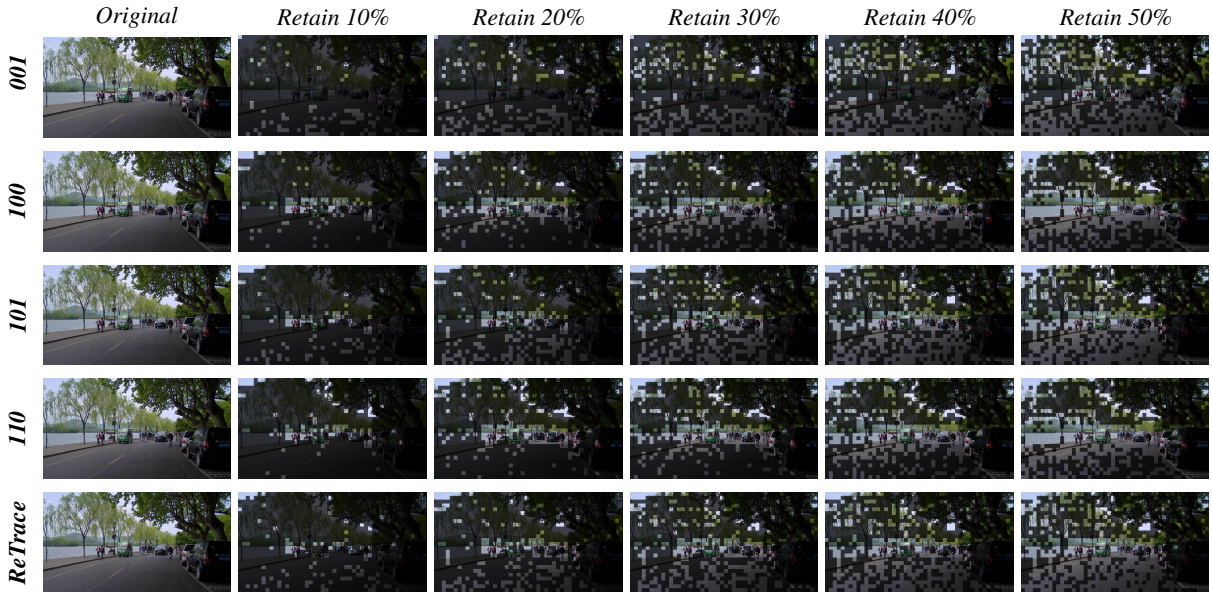


Figure 24: **Qualitative comparison.**

Algorithm 1 ReTrace

Require: Visual tokens $\mathcal{V} = \{v_1, v_2, \ldots, v_N\}$; grid windows $\mathcal{W}$; retention ratios ρ_1, ρ_2; injection layer b; pruning layer p; last text token t_{last}.

Ensure: Compact visual token set $\mathcal{P}_{final}$.

% Stage 1: Cache – Pivot Buffer Construction

$K_1 \leftarrow \lfloor \rho_1 N \rfloor, \quad K_2 \leftarrow \lfloor \rho_2 K_1 \rfloor$

Compute global saliency scores $s_g(v_i)$ from a selected encoder attention layer

Compute local saliency scores $s_l(v_i)$ from a selected encoder attention layer

$T_g \leftarrow \text{Top-}K_g(\{s_g(v_i) \mid v_i \in \mathcal{V}\})$ ▷ Global saliency

$T_l \leftarrow \{\arg\max_{v_i \in W_m} s_l(v_i) \mid W_m \in \mathcal{W}\}$ ▷ Local saliency

$T_d \leftarrow \text{FarthestPointSample}(\mathcal{V} \setminus (T_g \cup T_l), \text{metric} = \text{cosine})$ ▷ Spatial diversity

$\mathcal{P} \leftarrow T_g \cup T_l \cup T_d$ ▷ Active pivot buffer, $|\mathcal{P}| = K_1$

$\mathcal{D} \leftarrow \mathcal{V} \setminus \mathcal{P}$ ▷ Dormant context

$\hat{\mathcal{P}}, \hat{\mathcal{D}} \leftarrow \text{Proj}(\mathcal{P}), \text{Proj}(\mathcal{D})$ ▷ Map visual features to the LLM hidden space

$\mathcal{P} \leftarrow \text{LLM}_{1 \to b-1}(\hat{\mathcal{P}})$ ▷ Forward projected active pivots

% Stage 2: Inject – Backtracking-Aware Context Restoration

for each $\hat{v}_d \in \hat{\mathcal{D}}$ **do**

  $S_{d,k} \leftarrow \frac{\hat{v}_d}{\|\hat{v}_d\|_2} \cdot \left(\frac{h_{b,p_k}}{\|h_{b,p_k}\|_2}\right)^\top \quad \forall p_k \in \mathcal{P}$

  $a(d) \leftarrow \arg\max_k S_{d,k}$

  $w_d \leftarrow S_{d,a(d)}$

end for

for each $p_k \in \mathcal{P}$ **do**

  $\tilde{\Delta}_k \leftarrow \frac{\sum_{\hat{v}_d \in \hat{\mathcal{D}}} \mathbf{1}[a(d)=k]\, w_d \hat{v}_d}{\sum_{\hat{v}_d \in \hat{\mathcal{D}}} \mathbf{1}[a(d)=k]\, w_d + \epsilon}$

  $h_{b,p_k} \leftarrow h_{b,p_k} + \tilde{\Delta}_k$

end for

$\mathcal{P} \leftarrow \text{LLM}_{b \to p-1}(\mathcal{P})$ ▷ Forward enriched pivots

% Stage 3: Prune – Post-Absorption Task Pruning

$A_i \leftarrow \text{Attn}(t_{last}, p_i) \quad \forall p_i \in \mathcal{P}$

$T_p \leftarrow \text{Bottom-}(K_1 - K_2)(\{p_i \mid A_i\})$

$\mathcal{P}_{final} \leftarrow \mathcal{P} \setminus T_p$

$\mathcal{P}_{final} \leftarrow \text{LLM}_{p \to L}(\mathcal{P}_{final})$ ▷ Forward to final layer

return $\mathcal{P}_{final}$