TFPRUNE: PRUNING VISUAL REDUNDANCY WHILE PRESERVING TOKEN FUNCTIONS FOR EFFICIENT VLMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Large vision-language models (LVLMs) achieve strong multimodal reasoning by converting visual inputs into long token sequences processed by every subsequent LLM layer. High-resolution inputs further increase sequence length despite spatial redundancy, resulting in considerable inference costs. To reduce this overhead, numerous pruning methods have been proposed, many of which estimate redundancy from attention saliency or feature similarity. However, these scores do not assess whether the downstream effects of removed tokens can be reproduced by the retained set. To address this limitation, we propose TFPrune, a training-free, plug-and-play framework based on future functional substitutability. Specifically, TFPrune introduces an instruction-conditioned functional score that uses an exact softmax leave-one-out identity to quantify the counterfactual effect of removing each visual token on subsequent attention outputs without requiring an additional decoder-block forward pass. It selects a subset that minimizes the maximum functional discrepancy between each unselected token and its nearest retained representative, thereby preserving evidence that is difficult to substitute. Rather than simply discarding unselected tokens, TFPrune transfers their features to functionally matched representatives and uses accumulated transfer weights to correct post-compression attention. Experiments on LLaVA-1.5-7B, LLaVA-NeXT-7B, and Qwen2.5-VL-7B across nine benchmarks and multiple compression ratios demonstrate its effectiveness. Applied to LLaVA-1.5-7B, TFPrune retains only 11.1% of visual tokens while preserving 98.3% of full-model performance, achieving a favorable trade-off between compression and accuracy.

1 INTRODUCTION

Vision-language models (VLMs) process visual tokens jointly with textual instructions to support multimodal understanding and generation (Liu et al., 2023; 2024a; Bai et al., 2025; Chen et al., 2024b). High-resolution inputs often produce longer visual-token sequences, increasing decoder prefill computation and KV-cache storage during generation. Visual-token compression reduces this overhead by retaining a smaller set of informative representatives. Existing methods use attention-based importance, feature diversity, or multimodal coverage to guide selection (Chen et al., 2024a; Zhang et al., 2025b; Alvar et al., 2025; Dong et al., 2026; Wu et al., 2026b). These criteria identify relevant or representative content, but do not directly specify whether retained tokens can approximate the computational effects of those removed.

Attention rankings at shallow layers can be unreliable for persistent visual-token removal. In Figure 1, attention-only selection at layers 2, 4, and 8 retains scattered regions, while the selected tokens concentrate on the queried umbrella at layer 16. This pattern is consistent with prior findings that shallow decoder layers primarily propagate visual information, whereas active visual–textual fusion emerges in middle layers (Wu et al., 2026a). Figure 2(a) shows that top-K overlap with layer 2 declines across all three budgets as depth increases, with a partial recovery at layer 32. More directly, across four benchmarks, the top-K tokens selected at layer 4 receive 27.2–30.6 percentage points less visual attention at layer 24 than those selected at layer 24 (Figure 2(b)). Shallow attention rankings can therefore miss visual positions that become prominent later, motivating a compression criterion that considers tokens' subsequent computational effects.

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

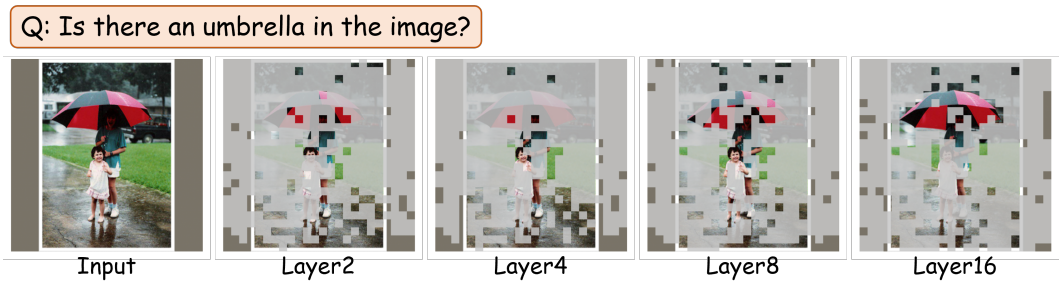


Figure 1: **Attention-only token selection across decoder layers.** The top-ranked visual tokens shift toward the queried umbrella at layer 16.

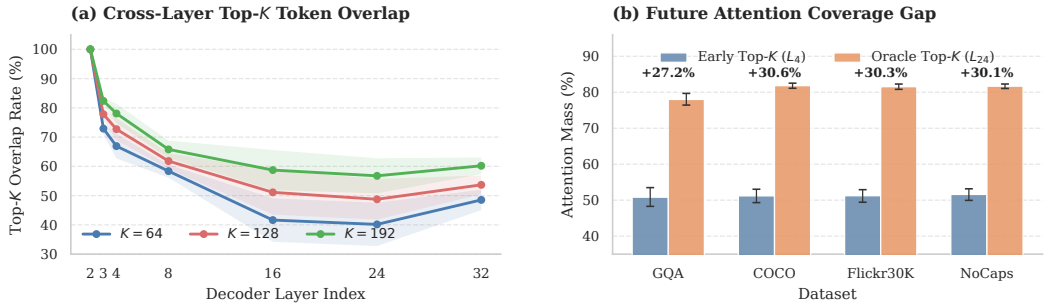


Figure 2: **Early token selection misses later visual attention.** (a) Top-K overlap with layer 2. (b) Layer-24 attention captured by layer-4 versus layer-24 selections.

These observations motivate *future functional substitutability*, which measures visual-token redundancy by how well retained tokens approximate a token's modeled effect on the following attention operation. This effect depends on attention weights, token values, and the remaining context, so an early attention ranking alone cannot characterize it. Tokens with similar functional effects may also carry different visual content, which can be lost through direct removal. Existing merging and recycling methods provide ways to preserve discarded content (Shang et al., 2025; Yang et al., 2024; Zhang et al., 2025b; 2026). A shared functional representation can connect the choice of retained tokens with the transfer of content from those removed.

TFPrune realizes this idea through three steps. It first represents each visual token by its instruction-conditioned functional response, which describes how removing the token changes the following attention output. It then selects a compact set of tokens that covers distinct responses and transfers content from unselected tokens to their retained representatives. Finally, accumulated transfer weights calibrate subsequent attention (Bolya et al., 2023). This training-free procedure uses the same functional representation to guide both token selection and content preservation.

Our contributions are threefold:

- We introduce **functional substitutability** to measure visual-token redundancy using instruction-conditioned removal responses and complementary value information.
- We connect **functional coverage and transport** through a shared representation and establish a conditional relation between directional coverage and response-reconstruction error.
- Experiments on nine benchmarks and three VLMs demonstrate strong performance under aggressive compression. Ablations and efficiency measurements examine component contributions and the performance–efficiency trade-off.

2 Related Work

Visual token pruning. Visual token pruning reduces the computational cost of VLMs by removing redundant visual tokens. A prominent line of work uses attention signals to identify informative tokens. Chen et al. (2024a) introduce FastV, which uses attention patterns in early decoder layers to prune visual tokens for subsequent computation. SparseVLM, proposed by Zhang et al. (2025b), uses visually relevant text tokens to guide progressive pruning. Beyond attention-based ranking, Alvar et al. (2025) formulate token selection as a max-min diversity problem in DivPrune, while DART (Wen et al., 2025) focuses on token redundancy by pruning visually redundant tokens based on their similarity to pivot tokens. Dong et al. (2026) jointly maximize textual and visual coverage through MMTok. PruneSID further combines semantic grouping with intra-group non-maximum suppression to balance importance and diversity (Fang et al., 2026). These methods provide effective criteria for retaining relevant and representative visual content. However, attention relevance and feature coverage do not directly establish whether retained tokens can approximate the downstream computational effects of removed tokens.

Visual token merging and recovery. To mitigate information loss during compression, several methods aggregate visual content that would otherwise be discarded. In vision transformers, Bolya et al. (2023) introduce ToMe, which merges similar tokens and adjusts attention according to their accumulated sizes. For VLMs, Shang et al. (2025) propose LLaVA-PruMerge, which supplements retained tokens through key-similarity-based clustering and attention-weighted aggregation. VisionZip constructs contextual tokens by merging non-dominant tokens with similar keys (Yang et al., 2024), while SparseVLM recycles a subset of pruned tokens into compact representations (Zhang et al., 2025b). These approaches demonstrate the value of recovering information alongside token pruning. They leave open whether the resulting representatives preserve the removed tokens' effects on subsequent attention.

Motivated by these observations, we propose TFPrune, a training-free framework that explicitly characterizes visual-token functional responses to guide both pruning and information recovery. By connecting redundancy estimation with functional preservation, TFPrune aims to retain useful visual evidence under limited token budgets.

3 Method

TFPrune is a training-free visual-token compression framework based on *future functional substitutability*. It characterizes tokens by their modeled effects on subsequent attention, selects representatives that cover distinct functional responses, and transfers unselected features to those representatives. A shared functional representation connects the criterion for identifying redundancy with the mechanism for recovering information.

3.1 Overview

Let $X = (x_1, \ldots, x_n)$ be the hidden-state sequence at a decoder boundary, and let $\mathcal{V}$ index its N visual tokens eligible for compression. Given a budget $1 \leq K \leq N$, TFPrune retains K representatives and aggregates unselected visual features into them. Functional estimation uses the current hidden states and the following block's attention operators, providing a local estimate of subsequent computational effects. Text and mandatory structural positions are retained. After compression, subsequent blocks process the shorter visual sequence. The insertion depth determines how much context is available for estimation and how many blocks benefit from compression.

3.2 Functional Responses

Defining token function. To characterize a visual token's local effect on subsequent computation, we measure how its removal changes the following block's attention outputs at text positions. Attention captures cross-position interactions, whereas the feed-forward sublayer processes each position independently (Vaswani et al., 2017). The resulting change reflects both the token's value contribution and the redistribution of attention over the remaining context.

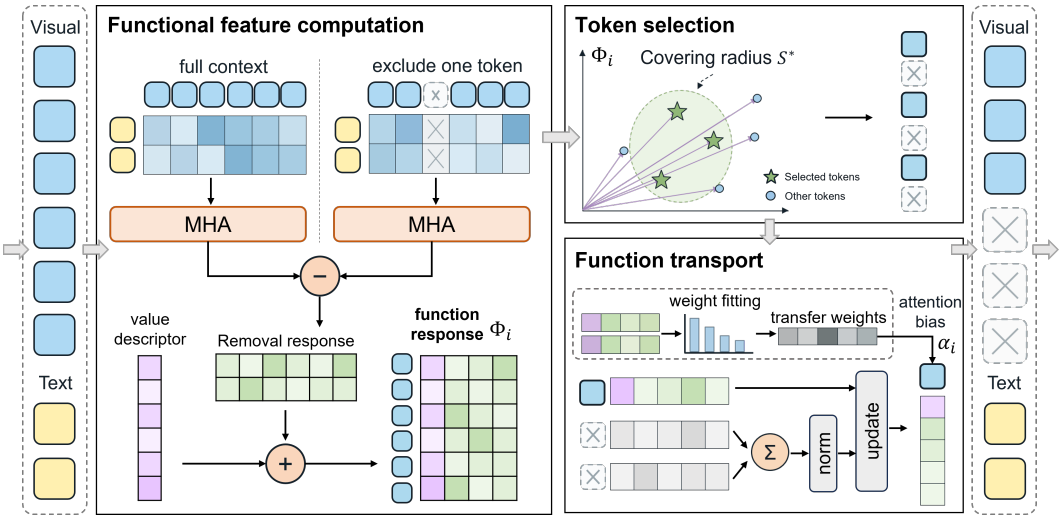


Figure 3: **Overview of TFPrune.** Functional responses guide representative selection and the transfer of removed-token features. Transfer weights calibrate subsequent attention.

For efficiency, we evaluate a small set $\mathcal{T}$ of text positions and a subset of attention heads. Let $\mathcal{A}_t(X)$ denote their combined attention output at text position t, mapped into a fixed low-dimensional space. Let $X_{\setminus i}$ exclude visual token i from the attention context while fixing all other hidden states, positions, selected heads, and projections. Its removal response is

$$f_i = \mathrm{vec}_{t\in\mathcal{T}}\big(\mathcal{A}_t(X) - \mathcal{A}_t(X_{\setminus i})\big)\,. \tag{1}$$

Here, vec concatenates differences across text positions. A softmax leave-one-out identity evaluates all responses using shared attention probabilities and projected values, without a separate forward pass per removal or an additional full decoder-block evaluation. The identity is exact for the fixed, projected attention operation when numerical safeguards are inactive. Appendix C.1 gives the derivation, query/head selection, and stabilized implementation.

Complementary value information. Because f_i is computed from a small set of text queries, tokens receiving little attention from those queries may have weak removal responses even if they contain useful visual content. We therefore supplement f_i with a descriptor p_i that sums token i's projected value contributions across the selected heads before attention weighting. The combined functional representation is

$$\phi_i = \mathrm{concat}(f_i/\sigma_f,\ p_i/\sigma_p)\,. \tag{2}$$

The scalars σ_f and σ_p are the root-mean-square token norms of the respective components over the visual span, with numerical floors specified in Appendix C.1. This shared scaling gives the two components comparable average squared norms while preserving magnitude differences between tokens within each component.

3.3 Selecting a Functional Cover

To represent a range of token functions with K visual tokens, we compare the directions of their functional representations while retaining response magnitudes for subsequent transfer. We apply a fixed random projection (Achlioptas, 2003) to obtain compact selection vectors and define $d_f(i,j)$ as their Euclidean distance after unit normalization. For a retained set S, let $R_f(S) = \max_{i\in\mathcal{V}} \min_{j\in S} d_f(i,j)$ denote its covering radius: the largest discrepancy between any visual token and its closest representative. We seek

$$S^* = \arg\min_{S\subseteq\mathcal{V},\,|S|=K} R_f(S). \tag{3}$$

Minimizing this worst-case distance helps retain distinctive functional responses that an average-distance objective could overlook.

We approximately solve Eq. 3 by farthest-first traversal (Gonzalez, 1985). Starting with the token of maximum squared response norm $\|\phi_i\|_2^2$, each step adds the token farthest from its nearest selected representative. After selecting K tokens, the remaining tokens are assigned to their nearest representatives. When the normalized selection vectors are nonzero, the resulting covering radius is at most twice the smallest achievable radius for any K-token set in the projected space. We retain the original functional vectors for transfer. Further theoretical and implementation details are provided in the Appendix.

3.4 Functional Transport

Functional similarity identifies candidate substitutes, but directly discarding unselected tokens may still lose complementary visual content. TFPrune uses the same functional representation to determine how their features contribute to retained representatives.

Fitting functional contributions. Let S be the selected set and $\pi(i)$ the nearest representative of token i under d_f, with $\pi(j) = j$ for $j \in S$. For each unselected token, we fit its original functional vector using a nonnegative multiple of its representative's vector:

$$\alpha_i = \arg \min_{0 \le \alpha \le \alpha_{\max}} \|\phi_i - \alpha \phi_{\pi(i)}\|_2^2. \tag{4}$$

The coefficient α_i fits token i's functional response using its assigned representative and sets the weight of x_i in feature aggregation. For every retained representative $j \in S$, we set $\alpha_j = 1$.

Directional coverage and functional fitting are mathematically connected. For nonzero original functional vectors, when coefficient clipping and numerical floors are inactive, the optimal relative fitting error is bounded by the distance between their unit-normalized directions. Because selection uses projected vectors, applying this relation to the selected cover also requires accounting for projection distortion. This relation supports the compatibility of coverage and transfer in functional space, without assuming that pooled features exactly preserve nonlinear decoder outputs.

Aggregating visual evidence. For representative j, define its assignment group $C_j = \{i \in \mathcal{V} : \pi(i) = j\}$ and accumulated transfer weight $m_j = \sum_{i \in C_j} \alpha_i$. Its compressed feature is

$$\widetilde{x}_j = \frac{1}{m_j} \sum_{i \in C_j} \alpha_i x_i. \tag{5}$$

The retained position incorporates content from its assigned tokens before their positions are removed. Since each representative has self-weight one, $m_j \ge 1$. For fixed assignments and nonnegative weights, this weighted average minimizes the group's weighted squared feature-reconstruction error. To account for accumulated transfer weights in subsequent attention, we add $\log m_j$ to the attention logits of representative j, following proportional attention (Bolya et al., 2023). This adjustment calibrates attention to each retained position according to its total transfer weight.

Budget and deployment. TFPrune supports fixed and proportional budgets. For a prescribed ratio $0 < \rho \le 1$, the proportional budget is $K = \max(1, \mathrm{round}(\rho N))$. The compressed sequence and attention correction persist during subsequent prefill and autoregressive decoding. Original representative positions are retained, and populated visual KV caches are aggregated using the transfer weights. Appendix C.5 details positional handling, structural-token accounting, and model integration. Smaller budgets reduce subsequent sequence-dependent computation, while realized latency also depends on response, selection, and aggregation overhead.

4 Experiments

4.1 Experimental Setup

We evaluate LLaVA-1.5-7B, LLaVA-NeXT-7B, and Qwen2.5-VL-7B-Instruct (Liu et al., 2023; 2024a; Bai et al., 2025) with lmms-eval (Zhang et al., 2025a). Nine benchmarks span visual question answering, multimodal understanding, hallucination, and captioning. They are GQA (Hudson & Manning, 2019), MMBench (Liu et al., 2024b), MME (Fu et al., 2025), OK-VQA (Marino et al.,

Table 1: Main results on LLaVA-1.5-7B.

Method	GQA EM	MMB Acc	MME P-score	OKVQA EM	SQA EM	POPE F1	COCO CIDEr	Flickr CIDEr	NoCaps CIDEr	Avg.
All 576 Tokens (100%)										
LLaVA-1.5-7B	62.0	64.1	1511	53.4	69.6	85.9	110.3	74.9	105.4	100%
Retain 192 Tokens (↓ 66.7%)										
FastV	59.4	63.8	1482	53.3	69.1	84.7	107.6	73.9	103.3	98.4%
DivPrune	60.0	62.5	1429	51.6	68.7	87.0	106.9	71.5	100.9	97.1%
MMTok	60.2	63.7	1439	51.7	68.8	86.8	103.3	70.9	99.6	96.8%
PruneSID	59.3	63.0	1438	52.3	68.8	86.1	106.3	71.6	101.0	97.1%
HiPrune	59.2	62.9	1463	52.3	68.7	86.2	107.4	73.4	101.5	97.7%
AnchorPrune	60.4	63.7	1462	52.3	68.4	86.9	106.6	73.4	102.0	98.0%
TFPrune (Ours)	61.7	64.2	1500	53.2	69.2	85.9	111.6	75.2	105.2	**99.9%**
Retain 128 Tokens (↓ 77.8%)										
FastV	56.9	62.5	1456	53.5	68.9	76.2	106.5	72.3	101.3	95.9%
DivPrune	59.3	62.1	1400	50.0	68.6	86.7	104.1	69.1	98.3	95.4%
MMTok	59.5	62.5	1432	50.1	69.3	86.3	100.6	69.2	96.8	95.3%
PruneSID	58.3	61.9	1423	50.1	68.2	84.7	103.2	69.7	98.1	95.0%
HiPrune	57.2	62.1	1425	50.4	68.4	83.0	104.9	70.8	98.4	95.1%
AnchorPrune	59.8	63.1	1452	50.8	68.7	86.8	103.9	71.9	99.3	96.7%
TFPrune (Ours)	60.9	64.1	1487	52.8	68.6	85.3	110.8	75.0	104.9	**99.3%**
Retain 64 Tokens (↓ 88.9%)										
FastV	53.6	60.1	1358	53.5	69.3	68.1	100.5	70.3	95.4	91.6%
DivPrune	57.8	59.5	1346	48.0	67.9	85.5	99.8	64.8	94.1	92.1%
MMTok	58.3	61.6	1392	48.6	69.3	85.8	96.8	66.2	92.5	93.0%
PruneSID	57.3	59.4	1384	47.9	68.1	84.7	96.7	65.0	93.8	91.8%
HiPrune	53.4	59.5	1321	46.4	69.0	73.1	90.8	60.3	84.2	86.7%
AnchorPrune	58.6	61.1	1419	49.0	68.9	85.8	99.0	67.0	93.6	93.6%
TFPrune (Ours)	59.8	63.7	1482	51.5	68.5	84.3	109.6	74.4	104.3	**98.3%**

2019), ScienceQA-IMG (SQA) (Lu et al., 2022), POPE (Li et al., 2023), COCO 2017 (Lin et al., 2014), Flickr30k (Plummer et al., 2015), and NoCaps (Agrawal et al., 2019). We use exact match for GQA, OK-VQA, and SQA, accuracy for MMBench, the perception score for MME, F1 for POPE, and CIDEr (Vedantam et al., 2015) for captioning. The Avg. column averages each task score relative to its uncompressed score for the same model. For efficiency, we measure end-to-end latency, samples per second, pruning overhead, and KV-cache size. Fixed-length efficiency and cache measurements use three independent runs.

We consider FastV (Chen et al., 2024a), DivPrune (Alvar et al., 2025), MMTok (Dong et al., 2026), PruneSID (Fang et al., 2026), ZOO-Prune (Kim et al., 2026), HiPrune (Liu et al., 2026), and AnchorPrune (Oh & Kim, 2026). These methods select tokens using signals available at the compression boundary, whereas TFPrune models each token's effect on the following attention operation to guide both selection and the transfer of removed content. We test $K \in \{192, 128, 64\}$ for LLaVA-1.5, $K \in \{640, 320, 160\}$ for LLaVA-NeXT, and retention ratios of 33.4%, 22.3%, and 11.2% for Qwen2.5-VL. More details are in Appendix A.

4.2 Main Results

Performance across token budgets. TFPrune achieves the highest normalized average at every LLaVA-1.5 budget in Table 1. Its margin over the strongest baseline widens from 1.5 to 2.6 and 4.7 percentage points as K decreases from 192 to 128 and 64. At $K = 64$, TFPrune retains 98.3% of the full-model average with 11.1% of the visual tokens. Captioning performance is especially well preserved: COCO, Flickr30k, and NoCaps retain approximately 99.4%, 99.3%, and 99.0% of their full-model CIDEr scores. Generating a description can require details from several image regions. Covering distinct functional responses helps keep those details represented, while transferring content from removed tokens gives the retained set more information to draw on during generation.

Effective compression of long visual sequences. LLaVA-NeXT supports higher-resolution images and can yield up to 2,880 visual tokens in the evaluated reference configuration, providing a test of compression on much longer inputs. In Table 2, keeping 640 tokens gives TFPrune 98.4% of the full-model normalized average. With only 160 tokens, it still achieves 93.7%, exceeding the strongest competing method by 4.4 percentage points. At this smallest budget, its GQA score is

Table 2: Main results on LLaVA-NeXT-7B.

Method	GQA EM	MMB Acc	MME P-score	OKVQA EM	SQA EM	POPE F1	COCO CIDEr	Flickr CIDEr	NoCaps CIDEr	Avg.
All 2,880 Tokens (100%)										
LLaVA-NeXT-7B	64.2	67.0	1529	44.2	70.3	86.4	100.1	68.8	88.3	100%
Retain 640 Tokens (↓ 77.8%)										
FastV	60.5	62.5	1514	43.2	67.9	82.6	99.9	66.4	84.0	96.4%
DivPrune	58.1	62.8	1315	45.5	67.4	79.6	95.9	67.1	84.5	94.5%
ZOO-Prune	61.3	64.5	1485	44.8	67.9	85.5	94.1	65.2	81.6	96.3%
HiPrune	60.6	66.7	1490	48.7	68.0	85.1	96.5	65.3	82.7	98.0%
TFPrune (Ours)	63.9	66.5	1497	43.8	69.5	86.5	97.7	66.7	85.0	**98.4%**
Retain 320 Tokens (↓ 88.9%)										
FastV	55.7	60.2	1327	41.7	67.0	73.9	95.1	63.4	80.1	90.7%
DivPrune	56.1	60.4	1278	44.9	67.3	74.4	94.2	65.2	81.7	91.8%
ZOO-Prune	59.8	63.9	1417	42.2	67.9	83.3	90.5	61.9	77.4	93.1%
HiPrune	57.2	64.9	1384	45.9	67.3	78.8	89.7	64.2	78.5	93.2%
TFPrune (Ours)	62.7	65.5	1506	42.3	68.4	86.3	95.4	64.7	82.4	**96.6%**
Retain 160 Tokens (↓ 94.4%)										
FastV	52.2	57.0	1215	40.3	66.1	68.5	89.1	61.5	77.1	86.2%
DivPrune	53.2	57.4	1212	42.2	67.3	67.9	90.4	61.8	77.4	87.3%
ZOO-Prune	58.2	61.9	1372	39.3	68.5	79.3	86.0	57.5	73.4	89.3%
HiPrune	52.4	60.3	1260	39.0	69.0	68.0	69.6	50.4	60.9	81.2%
TFPrune (Ours)	61.4	65.3	1449	41.3	68.3	85.0	91.2	60.8	76.5	**93.7%**

Table 3: Main results on Qwen2.5-VL-7B.

Method	GQA EM	MMB Acc	MME P-score	OKVQA EM	SQA EM	POPE F1	COCO CIDEr	Flickr CIDEr	NoCaps CIDEr	Avg.
All Tokens (100%)										
Qwen2.5-VL-7B	60.9	84.1	1684	43.7	88.3	86.3	32.5	33.2	37.3	100%
Retain 33.4% Tokens										
FastV	56.1	79.7	1627	38.2	84.6	84.2	36.2	34.4	39.2	98.3%
DivPrune	57.9	79.6	1691	37.5	82.9	81.8	31.1	31.7	35.2	94.5%
MMTok	59.4	81.9	1694	39.3	85.4	83.9	32.7	33.5	37.5	97.9%
TFPrune (Ours)	60.2	81.4	1682	39.9	85.9	85.7	33.9	33.2	38.5	**99.0%**
Retain 22.3% Tokens										
FastV	53.3	77.7	1582	35.3	81.6	81.0	35.9	32.7	38.8	94.9%
DivPrune	57.2	77.3	1655	35.7	80.9	81.1	30.8	31.3	36.9	93.3%
MMTok	57.9	79.9	1686	37.1	83.7	82.3	32.2	33.0	36.6	95.8%
TFPrune (Ours)	59.9	81.1	1665	39.0	85.2	85.0	33.9	33.1	38.4	**98.3%**
Retain 11.2% Tokens										
FastV	47.4	66.7	1400	28.4	79.2	71.1	31.7	26.0	35.9	83.3%
DivPrune	54.1	73.5	1557	30.3	80.4	76.4	30.1	28.3	34.7	87.7%
MMTok	54.8	75.7	1524	31.7	81.8	78.0	30.9	30.2	35.4	89.7%
TFPrune (Ours)	58.3	79.3	1640	35.7	82.7	83.8	33.6	32.3	38.2	**95.9%**

3.2 points above the strongest competing result, while POPE F1 is higher by 5.7 points. The retained functional responses remain useful in the higher-resolution setting even when 94.4% of the reference visual positions are removed. The modest aggregate loss at that budget also indicates substantial redundancy in these evaluated long visual sequences.

Effectiveness across model architectures. TFPrune requires no additional training and can be integrated into different decoders. To assess this flexibility, we evaluate it on Qwen2.5-VL, which uses dynamic-resolution image processing and merges adjacent visual features before passing them to the language model (Bai et al., 2025). We compare against FastV, DivPrune, and MMTok, which have Qwen2.5-VL implementations. As shown in Table 3, TFPrune retains 95.9% of the full-model average score at 11.2% visual-token retention and leads all three baselines on every reported task. The strongest baseline reaches 89.7%, leaving a 6.2-percentage-point gap. These results suggest that visual redundancy remains after Qwen2.5-VL's native feature merging. TFPrune removes much of this redundancy while preserving most of the model's task performance. Together with the LLaVA results, this supports its effectiveness across model architectures and token budgets.

4.3 Ablation Studies

Table 4 asks whether the representation captures useful token contributions, whether selection covers distinct responses, and whether those contributions survive compression. At $K = 64$ on LLaVA-1.5, hard pruning of the functionally selected tokens reduces COCO CIDEr from 109.56 to 103.44 and SQA accuracy from 68.52 to 64.61. This comparison isolates the value of preserving content after representative selection rather than changing the selection criterion. The auxiliary value component encodes projected visual values before text-query attention weights are applied, helping retain content that a limited set of text queries may attend to weakly.

Table 4: Ablation study.

Setting	COCO CIDEr	GQA EM	SQA EM
Full	**109.56**	**59.83**	**68.52**
w/o value component	106.70	58.87	67.37
w/o functional transport	103.16	58.73	65.72
w/o attention bias	106.73	59.12	67.46
Hard pruning	103.44	58.03	64.61
Average coverage	105.11	57.64	66.92

The choice of cover matters as well. Replacing minimax coverage with average coverage reduces GQA accuracy by 2.19 points and COCO CIDEr by 4.45 points. Protecting the least well represented response appears valuable when only 64 tokens remain. Disabling functional transport lowers COCO CIDEr by 6.40 points and SQA accuracy by 2.80 points. Disabling only the accumulated-mass attention bias also lowers all three scores. Together, these comparisons indicate that a representative set is most effective when omitted visual content contributes to its features and subsequent attention.

4.4 Efficiency Analysis

End-to-end acceleration. Table 5 includes compression overhead in the measured COCO captioning latency. Reducing the retained set from 192 to 64 tokens lowers latency from 896.67 to 723.47 ms per sample and pruning time from 129.82 to 44.76 ms. Relative to the native full-token model, $K = 64$ gives a 1.268× end-to-end speedup. At $K = 192$, speedup is only 1.023×. This makes the performance-efficiency choice budget dependent. The corresponding LLaVA-1.5 normalized task averages are 99.9%, 99.3%, and 98.3% at $K = 192, 128, 64$, respectively. Selecting a smaller budget gives larger latency savings at the cost of a modest decrease in task performance.

We also compared the end-to-end latency of TFPrune and FastV. Appendix B.5.4 reports the measured latency at matched token budgets and specifies the distinct pruning layers used in this comparison.

KV-cache reduction. FastV is a useful efficiency comparator because it remains competitive across budgets in Tables 1 and 2 and represents early-attention-based pruning (Chen et al., 2024a). We compare KV-cache storage at matched visual-token budgets. TFPrune shortens the visual prefix and aggregates existing cache entries, so removed positions no longer require individual KV entries during decoding. Table 6 shows that retaining 64 visual tokens reduces prefill cache storage by

Table 5: End-to-end COCO efficiency on LLaVA-1.5-7B.

Method	tokens	E2E (ms/sample) ↓	Samples/s ↑	Speedup vs. native	Pruning (ms) ↓
LLaVA-1.5-7B	576 (100%)	917.60	1.090	1.000×	—
TFPrune	192 (33.3%)	896.67	1.115	1.023×	129.82
TFPrune	128 (22.2%)	791.20	1.264	1.160×	80.73
TFPrune	**64 (11.1%)**	**723.47**	**1.382**	**1.268×**	**44.76**

Table 6: KV-cache size (MiB) on LLaVA-1.5-7B.

		COCO		GQA		POPE		OK-VQA	
Method	tokens	prefill	final	prefill	final	prefill	final	prefill	final
LLaVA-1.5-7B	576 (100%)	316.5	348.0	319.8	351.3	318.8	350.3	328.2	359.7
FastV	128 (22.2%)	106.5	138.0	109.8	141.3	108.8	140.3	118.2	149.7
	64 (11.1%)	76.5	108.0	79.8	111.3	78.8	110.3	88.2	119.7
	32 (5.6%)	61.5	93.0	64.8	96.3	63.8	95.3	73.2	104.7
TFPrune (Ours)	128 (22.2%)	92.5	124.0	95.8	127.3	94.8	126.3	104.2	135.7
	64 (11.1%)	60.5	92.0	63.8	95.3	62.8	94.3	72.2	103.7
	32 (5.6%)	44.5	76.0	47.8	79.3	46.8	78.3	56.2	87.7

78.0–80.9% and final cache storage by 71.2–73.6% across four tasks. Storage is also lower than FastV's at matched budgets. At $K = 32$, prefill savings reach 82.9–85.9% relative to the full-token model.

4.5 Case Study: Retaining Evidence Used by Later Layers

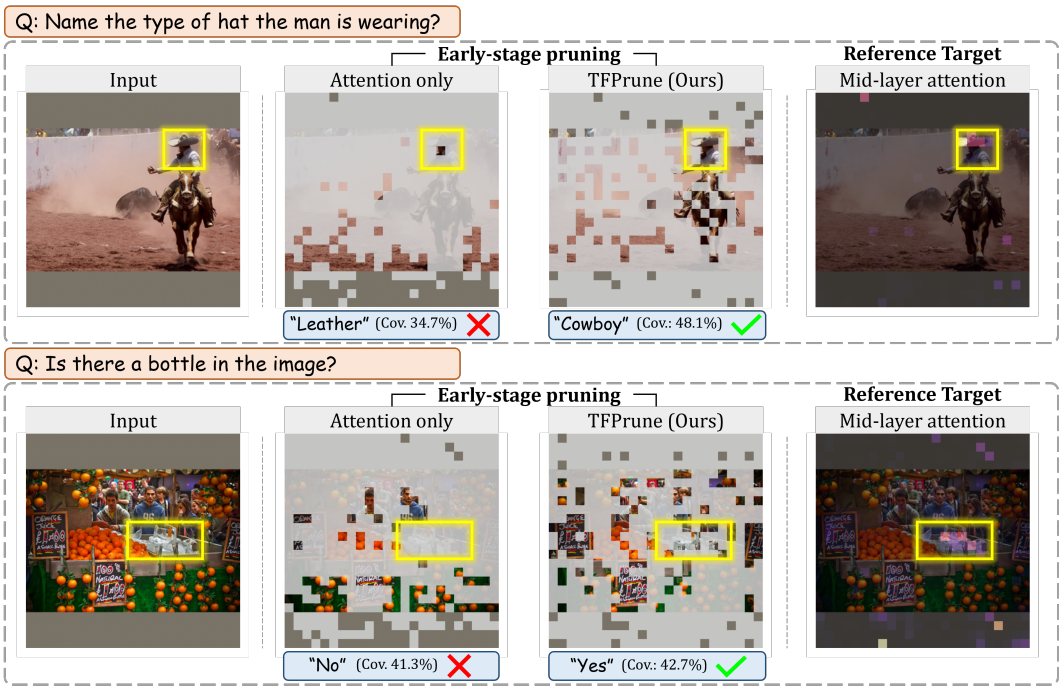


Figure 4: **Case comparison under attention shift.** Early attention-only selection and TFPrune are shown for two visual questions. Cov. denotes coverage of the reference middle-layer attention.

Figure 4 revisits the mismatch between early token importance and the visual regions emphasized by subsequent layers. For the question about the rider's hat, TFPrune preserves more of the queried region than early attention-only selection. Its middle-layer reference attention coverage is 48.1%, versus 34.7%, and its answer is "Cowboy" rather than "Leather". In the bottle example, TFPrune answers "Yes" instead of "No" and increases the displayed coverage from 41.3% to 42.7%.

TFPrune still starts from early hidden states, but it also evaluates how each token would affect the next attention operation and transfers contributions from omitted tokens. The examples suggest that these functional responses can help retain positions useful during later multimodal processing. An intervention analysis further shows that shallow functional-response dissimilarity is more strongly associated with later hidden-state changes than shallow feature dissimilarity across nine benchmarks (Appendix B.1). Selection and transport together offer a way to cover complementary responses and preserve their contributions without accessing future attention maps during inference.

5 Conclusion

TFPrune frames visual-token compression as functional substitutability: retained tokens cover the modeled attention responses of removed tokens and receive their visual contributions. This training-free combination of functional coverage and transport addresses the cross-layer shift in attention observed in our motivation study. Across three VLMs and nine benchmarks, TFPrune maintains strong performance under aggressive compression. Ablations support the roles of coverage and transport, while efficiency results show substantial KV-cache savings and workload-dependent end-to-end acceleration. These results suggest a practical route to efficient multimodal inference through a shared representation for deciding which tokens to retain and how to preserve the contributions of those removed.

AI USE STATEMENT

We used generative AI tools to search for and survey related papers and to polish the language of this manuscript. The authors take responsibility for the cited literature and for the final content and claims of this work.

ETHICS STATEMENT

To the best of our knowledge, this work has no potential ethical issues to disclose.

REPRODUCIBILITY STATEMENT

The method and experimental protocols are described in the main text and appendix. Code and evaluation scripts will be available in an anonymous repository at `https://anonymous.4open.science/r/TFPrune-iclr2027`.

REFERENCES

Dimitris Achlioptas. Database-friendly random projections: Johnson–Lindenstrauss with binary coins. *Journal of Computer and System Sciences*, 66(4):671–687, 2003. doi: 10.1016/S0022-0000(03)00025-4. URL https://doi.org/10.1016/S0022-0000(03)00025-4.

Harsh Agrawal, Karan Desai, Yufei Wang, Xinlei Chen, Rishabh Jain, Mark Johnson, Dhruv Batra, Devi Parikh, Stefan Lee, and Peter Anderson. Nocaps: Novel object captioning at scale. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8947–8956. IEEE, 2019.

Saeed Ranjbar Alvar, Gursimran Singh, Mohammad Akbari, and Yong Zhang. Divprune: Diversity-based visual token pruning for large multimodal models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 9392–9401, 2025.

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, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your vit but faster. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=JroZRaRw7Eu.

Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang. 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*, pp. 19–35. Springer, 2024a.

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. 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*, pp. 24185–24198, 2024b.

Sixun Dong, Juhua Hu, Mian Zhang, Ming Yin, Yanjie Fu, and Qi Qian. MMTok: Multimodal coverage maximization for efficient inference of VLMs. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=GvPdSWZT31.

Zhengyao Fang, Pengyuan Lyu, Chengquan Zhang, Guangming Lu, Jun Yu, and Wenjie Pei. Prune redundancy, preserve essence: Vision token compression in vlms via synergistic importance-diversity. *CoRR*, abs/2603.09480, 2026. doi: 10.48550/ARXIV.2603.09480. URL https://doi.org/10.48550/arXiv.2603.09480.

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. Mme: A comprehensive evaluation benchmark for multimodal large language models. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference. Curran Associates, Inc., 2025. doi: 10.52202/085713-4899. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/d79a27cf2772fe00be7f341efc0eb517-Paper-Datasets_and_Benchmarks_Track.pdf.

Teofilo F Gonzalez. Clustering to minimize the maximum intercluster distance. *Theoretical computer science*, 38:293–306, 1985.

Drew A Hudson and Christopher D Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6693–6702. IEEE, 2019.

Youngeun Kim, Youjia Zhang, Huiling Liu, Aecheon Jung, Sunwoo Lee, and Sungeun Hong. Zooprune: training-free token pruning via zeroth-order gradient estimation in vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 39572–39582, 2026.

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

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pp. 740–755. Springer, 2014.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning, 2023.

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

Jizhihui Liu, Guangdao Zhu, Feiyi Du, Niu Lian, Jun Li, Bin Chen, Weili Guan, and Yaowei Wang. Hiprune: Hierarchical attention for efficient token pruning in vision-language models. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 3274–3291, 2026.

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

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

Kenneth Marino, Mohammad Rastegari, Ali Farhadi, and Roozbeh Mottaghi. Ok-vqa: A visual question answering benchmark requiring external knowledge. In *2019 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 3190–3199. IEEE, 2019.

Kyuan Oh and Bumsoo Kim. Anchorprune: Relevance-anchored contextual expansion for visual token pruning. In *European Conference on Computer Vision*, pp. 595–612. Springer, 2026.

Bryan A Plummer, Liwei Wang, Chris M Cervantes, Juan C Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *Proceedings of the IEEE international conference on computer vision*, pp. 2641–2649, 2015.

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

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4566–4575, 2015.

Zichen Wen, Yifeng Gao, Shaobo Wang, Junyuan Zhang, Qintong Zhang, Weijia Li, Conghui He, and Linfeng Zhang. Stop looking for important tokens in multimodal language models: Duplication matters more. *arXiv preprint arXiv:2502.11494*, 2025.

Hao Wu, Yingqi Fan, Dai Jinyang, Junlong Tong, Yunpu Ma, and Xiaoyu Shen. Hidrop: Hierarchical vision token reduction in mllms via late injection, concave pyramid pruning, and early exit. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 97296–97316, 2026a. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/9e0d77d3ce77fb7369663c27d7bcc838-Paper-Conference.pdf.

Zhenkai Wu, Xiaowen Ma, Zhenliang Ni, Dengming Zhang, Han Shu, Xin Jiang, and Xinghao Chen. 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)*, pp. 31952–31961, June 2026b.

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

Kaichen Zhang, Bo Li, Peiyuan Zhang, Fanyi Pu, Joshua Adrian Cahyono, Kairui Hu, Shuai Liu, Yuanhan Zhang, Jingkang Yang, Chunyuan Li, et al. Lmms-eval: Reality check on the evaluation of large multimodal models. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 881–916, 2025a.

Yuan Zhang, Chun-Kai Fan, Junpeng Ma, Wenzhao Zheng, Tao Huang, Kuan Cheng, Denis Gudovskiy, Tomoyuki Okuno, Yohei Nakata, Kurt Keutzer, et al. Sparsevlm: Visual token sparsification for efficient vision-language model inference. In *International Conference on Machine Learning*, 2025b.

Yuan Zhang, Junpeng Ma, Qizhe Zhang, Chun-Kai Fan, Wenzhao Zheng, Kuan Cheng, Jiwen Lu, and Shanghang Zhang. Sparsevlm+: Visual token sparsification with improved text-visual attention pattern. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2026.

A Experimental Setup

A.1 VLM Backbones

A.1.1 LLaVA-1.5-7B

LLaVA-1.5 encodes a 336×336 image with a CLIP ViT-L/14 vision tower and maps the resulting features to the Vicuna decoder through a visual projector (Liu et al., 2023). Its 24×24 patch grid supplies 576 visual tokens before compression. We evaluate fixed budgets of 192, 128, and 64 tokens.

A.1.2 LLaVA-NeXT-7B

LLaVA-NeXT extends the LLaVA architecture with higher-resolution image processing (Liu et al., 2024a). The evaluated Vicuna-based checkpoint uses image crops and a base image rather than a single fixed-resolution patch grid. The reference configuration in the main results is reported with up to 2,880 visual tokens, and its fixed budgets are 640, 320, and 160. The exact visual-token count can vary with the image and its layout.

A.1.3 Qwen2.5-VL-7B-Instruct

Qwen2.5-VL processes images at dynamic resolution and merges adjacent vision features before they enter its language decoder (Bai et al., 2025). The supplied evaluation launcher sets `min_pixels=200704` and `max_pixels=1605632`. Thus neither the processed image dimensions nor the number of decoder visual tokens is fixed across the evaluation set. We request proportional budgets corresponding to the three retention levels reported in Table 3. TFPrune selects and aggregates tokens after Qwen's native feature merger, which allows the experiment to test for redundancy remaining at the decoder input.

A.2 Datasets and Evaluation

The main tables cover nine benchmarks spanning image captioning, visual question answering, multimodal understanding, and hallucination detection. The task identifiers below match those used in our evaluation scripts.

COCO 2017. Task `coco2017_cap_val`. This task evaluates caption generation for images from the COCO 2017 validation set (Lin et al., 2014). Its everyday scenes test whether a model can identify salient objects and describe their actions and relationships in a coherent sentence.

Flickr30k. Task `flickr30k_test`. This image-captioning task uses the Flickr30k test split (Plummer et al., 2015). It tests whether generated descriptions capture the people, objects, activities, and relationships depicted in an image.

NoCaps. Task `nocaps_val`. NoCaps evaluates caption generation on images containing objects that are uncommon or absent in standard caption-training data (Agrawal et al., 2019). It tests whether a model can recognize these objects and incorporate them into an accurate description, rather than relying only on familiar caption patterns.

POPE. Task `pope`. POPE evaluates object hallucination through yes/no questions about whether a named object appears in an image (Li et al., 2023). It tests whether a model grounds its answer in visible evidence instead of asserting the presence of plausible but absent objects.

MME. Task `mme`. MME assesses a range of multimodal perception and cognition abilities (Fu et al., 2025). We report its perception score, which tests capabilities such as identifying objects and attributes, counting, understanding positions, and reading text in images.

GQA. Task `gqa`. GQA uses questions about real images that involve objects, attributes, and their relationships (Hudson & Manning, 2019). Its compositional questions test scene understanding and visual reasoning beyond recognizing an individual object.

OK-VQA. Task `ok_vqa_val2014`. OK-VQA contains image questions that cannot always be answered from visual content alone (Marino et al., 2019). It tests whether a model can connect what it sees with relevant external knowledge to produce an answer.

ScienceQA-IMG. Task `scienceqa_img`. This subset of ScienceQA contains science questions accompanied by images (Lu et al., 2022). It tests whether a model can combine visual evidence with scientific knowledge and reasoning when selecting an answer.

MMBench. Task `mmbench_en_dev`. MMBench evaluates multimodal understanding using questions across fine-grained ability dimensions (Liu et al., 2024b). Its English development split tests visual perception alongside capabilities such as relational understanding, knowledge use, and reasoning.

All tasks are evaluated using lmms-eval v0.7.1 (Zhang et al., 2025a) with batch size one. GQA, OK-VQA, and ScienceQA-IMG use exact match, MMBench uses accuracy, MME uses the perception score, and POPE uses F1. COCO, Flickr30k, and NoCaps use CIDEr (Vedantam et al., 2015). Let $s_{m,t}$ denote the score of a method on task t under model m, and $s_{m,t}^{\text{full}}$ the uncompressed score of the same model. We report

$$\text{Avg.}(m) = \frac{100}{9} \sum_{t=1}^{9} \frac{s_{m,t}}{s_{m,t}^{\text{full}}}. \tag{6}$$

Ratios exceeding 100% are retained without clipping. The three captioning scores and the MME score have different scales from accuracy and F1, so averaging the original scores would have no meaningful interpretation.

A.3 Baselines and Budget Matching

We compare with seven visual-token compression methods. FastV (Chen et al., 2024a) uses decoder attention to identify visual tokens for subsequent pruning. DivPrune (Alvar et al., 2025) selects tokens for diversity in visual feature space. MMTok (Dong et al., 2026) maximizes coverage of both visual and textual representations. PruneSID (Fang et al., 2026) combines semantic grouping with within-group suppression to balance relevance and diversity. ZOO-Prune (Kim et al., 2026) estimates token sensitivity using zeroth-order perturbations. HiPrune (Liu et al., 2026) exploits attention at different levels of the vision encoder and retains complementary anchor, buffer, and register tokens. AnchorPrune (Oh & Kim, 2026) protects relevance-based anchors before expanding the retained set with complementary context.

We use the baseline implementations' default settings and change the visual-token budget to match each comparison. Baselines may compress at different decoder boundaries. The LLaVA-1.5 results contain FastV, DivPrune, MMTok, PruneSID, HiPrune, and AnchorPrune. The LLaVA-NeXT results contain FastV, DivPrune, ZOO-Prune, and HiPrune. The Qwen2.5-VL results contain FastV, DivPrune, and MMTok. All evaluations within a model use the same lmms-eval tasks. For Qwen2.5-VL, the launcher requests proportional ratios 0.333, 0.222, and 0.111, while the tables report rounded effective retention percentages of 33.4%, 22.3%, and 11.2%.

B Extended Experimental Results

B.1 Single-Token Intervention Analysis

We examine whether scores available after decoder block 4 are associated with the effect of a visual-token intervention at a later layer. For each preselected token pair, we merge one visual token into another and remove its original position. We then measure the relative change in the last-prompt hidden state at block 24. Table 7 reports Spearman correlations between this change and three shallow-layer scores: next-block functional-response dissimilarity (R), feature dissimilarity (F), and attention assigned to the removed token (A).

Functional-response dissimilarity has a stronger association with the block-24 hidden-state change than feature dissimilarity on all nine datasets. Every confidence interval for Δ_{R-F} lies above zero, and the equal-dataset mean correlations are 0.530 for R and 0.349 for F. The comparison with early attention is less uniform. These findings support functional responses as a measure of token-pair differences that are relevant to later representations.

Table 7: Association between shallow-layer scores and the block-24 hidden-state change under a single-token merge-and-delete intervention on LLaVA-1.5-7B. Intervals are 95% image-bootstrap confidence intervals for paired differences in Spearman correlation.

Dataset	ρ_R	ρ_F	ρ_A	Δ_{R-F} [95% CI]	Δ_{R-A} [95% CI]
COCO	0.600	0.488	0.217	+0.112 [+0.077, +0.151]	+0.383 [+0.274, +0.484]
Flickr30k	0.548	0.423	0.246	+0.125 [+0.093, +0.163]	+0.302 [+0.190, +0.416]
GQA	0.526	0.369	0.125	+0.157 [+0.119, +0.198]	+0.401 [+0.312, +0.486]
MMBench	0.469	0.225	0.403	+0.244 [+0.208, +0.282]	+0.066 [-0.018, +0.141]
MME	0.587	0.394	0.407	+0.194 [+0.153, +0.234]	+0.180 [+0.084, +0.272]
NoCaps	0.629	0.472	0.308	+0.157 [+0.115, +0.195]	+0.321 [+0.217, +0.425]
OK-VQA	0.527	0.336	0.316	+0.191 [+0.150, +0.233]	+0.210 [+0.109, +0.314]
POPE	0.507	0.374	0.180	+0.133 [+0.092, +0.177]	+0.326 [+0.228, +0.428]
ScienceQA-IMG	0.373	0.057	0.477	+0.316 [+0.275, +0.354]	-0.103 [-0.198, -0.004]

B.2 Detailed Results on LLaVA-1.5-7B

Table 8 expands Table 1. Each method occupies two lines. *Raw* gives the metric score, and *Rel.* divides that score by the full-token result on the same task. The Avg. column appears only in the relative line because the raw metrics are on incompatible scales. Relative values are calculated from the displayed scores and rounded to one decimal place, while Avg. reproduces the main table's value computed from the underlying task scores.

B.3 Detailed Results on LLaVA-NeXT-7B

Table 9 reports raw and full-model-relative task scores for the longer visual sequences in Table 2. The full-token row uses the 2,880-token reference configuration in the reported table.

B.4 Detailed Results on Qwen2.5-VL-7B-Instruct

Table 10 expands Table 3 at the three displayed proportional budgets. Scores are normalized to the same model without TFPrune. Relative scores greater than 100% indicate a task improvement over that model-specific reference.

Table 8: Detailed task results on LLaVA-1.5-7B.

Method	Type	GQA	MMB	MME	OK-VQA	SQA	POPE	COCO	Flickr	NoCaps	Avg.
All 576 Tokens (100%)											
LLaVA-1.5-7B	Raw	62.0	64.1	1511	53.4	69.6	85.9	110.3	74.9	105.4	—
	Rel.	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%
Retain 192 Tokens (removed 66.7%)											
FastV	Raw	59.4	63.8	1482	53.3	69.1	84.7	107.6	73.9	103.3	—
	Rel.	95.8%	99.5%	98.1%	99.8%	99.3%	98.6%	97.6%	98.7%	98.0%	98.4%
DivPrune	Raw	60.0	62.5	1429	51.6	68.7	87.0	106.9	71.5	100.9	—
	Rel.	96.8%	97.5%	94.6%	96.6%	98.7%	101.3%	96.9%	95.5%	95.7%	97.1%
MMTok	Raw	60.2	63.7	1439	51.7	68.8	86.8	103.3	70.9	99.6	—
	Rel.	97.1%	99.4%	95.2%	96.8%	98.9%	101.0%	93.7%	94.7%	94.5%	96.8%
PruneSID	Raw	59.3	63.0	1438	52.3	68.8	86.1	106.3	71.6	101.0	—
	Rel.	95.6%	98.3%	95.2%	97.9%	98.9%	100.2%	96.4%	95.6%	95.8%	97.1%
HiPrune	Raw	59.2	62.9	1463	52.3	68.7	86.2	107.4	73.4	101.5	—
	Rel.	95.5%	98.1%	96.8%	97.9%	98.7%	100.3%	97.4%	98.0%	96.3%	97.7%
AnchorPrune	Raw	60.4	63.7	1462	52.3	68.4	86.9	106.6	73.4	102.0	—
	Rel.	97.4%	99.4%	96.8%	97.9%	98.3%	101.2%	96.6%	98.0%	96.8%	98.0%
TFPrune	Raw	61.7	64.2	1500	53.2	69.2	85.9	111.6	75.2	105.2	—
	Rel.	**99.5%**	**100.2%**	**99.3%**	**99.6%**	**99.4%**	**100.0%**	**101.2%**	**100.4%**	**99.8%**	**99.9%**
Retain 128 Tokens (removed 77.8%)											
FastV	Raw	56.9	62.5	1456	53.5	68.9	76.2	106.5	72.3	101.3	—
	Rel.	91.8%	97.5%	96.4%	100.2%	99.0%	88.7%	96.6%	96.5%	96.1%	95.9%
DivPrune	Raw	59.3	62.1	1400	50.0	68.6	86.7	104.1	69.1	98.3	—
	Rel.	95.6%	96.9%	92.7%	93.6%	98.6%	100.9%	94.4%	92.3%	93.3%	95.4%
MMTok	Raw	59.5	62.5	1432	50.1	69.3	86.3	100.6	69.2	96.8	—
	Rel.	96.0%	97.5%	94.8%	93.8%	99.6%	100.5%	91.2%	92.4%	91.8%	95.3%
PruneSID	Raw	58.3	61.9	1423	50.1	68.2	84.7	103.2	69.7	98.1	—
	Rel.	94.0%	96.6%	94.2%	93.8%	98.0%	98.6%	93.6%	93.1%	93.1%	95.0%
HiPrune	Raw	57.2	62.1	1425	50.4	68.4	83.0	104.9	70.8	98.4	—
	Rel.	92.3%	96.9%	94.3%	94.4%	98.3%	96.6%	95.1%	94.5%	93.4%	95.1%
AnchorPrune	Raw	59.8	63.1	1452	50.8	68.7	86.8	103.9	71.9	99.3	—
	Rel.	96.5%	98.4%	96.1%	95.1%	98.7%	101.0%	94.2%	96.0%	94.2%	96.7%
TFPrune	Raw	60.9	64.1	1487	52.8	68.6	85.3	110.8	75.0	104.9	—
	Rel.	**98.2%**	**100.0%**	**98.4%**	**98.9%**	**98.6%**	**99.3%**	**100.5%**	**100.1%**	**99.5%**	**99.3%**
Retain 64 Tokens (removed 88.9%)											
FastV	Raw	53.6	60.1	1358	53.5	69.3	68.1	100.5	70.3	95.4	—
	Rel.	86.5%	93.8%	89.9%	100.2%	99.6%	79.3%	91.1%	93.9%	90.5%	91.6%
DivPrune	Raw	57.8	59.5	1346	48.0	67.9	85.5	99.8	64.8	94.1	—
	Rel.	93.2%	92.8%	89.1%	89.9%	97.6%	99.5%	90.5%	86.5%	89.3%	92.1%
MMTok	Raw	58.3	61.6	1392	48.6	69.3	85.8	96.8	66.2	92.5	—
	Rel.	94.0%	96.1%	92.1%	91.0%	99.6%	99.9%	87.8%	88.4%	87.8%	93.0%
PruneSID	Raw	57.3	59.4	1384	47.9	68.1	84.7	96.7	65.0	93.8	—
	Rel.	92.4%	92.7%	91.6%	89.7%	97.8%	98.6%	87.7%	86.8%	89.0%	91.8%
HiPrune	Raw	53.4	59.5	1321	46.4	69.0	73.1	90.8	60.3	84.2	—
	Rel.	86.1%	92.8%	87.4%	86.9%	99.1%	85.1%	82.3%	80.5%	79.9%	86.7%
AnchorPrune	Raw	58.6	61.1	1419	49.0	68.9	85.8	99.0	67.0	93.6	—
	Rel.	94.5%	95.3%	93.9%	91.8%	99.0%	99.9%	89.8%	89.5%	88.8%	93.6%
TFPrune	Raw	59.8	63.7	1482	51.5	68.5	84.3	109.6	74.4	104.3	—
	Rel.	**96.5%**	**99.4%**	**98.1%**	**96.4%**	**98.4%**	**98.1%**	**99.4%**	**99.3%**	**99.0%**	**98.3%**

Table 9: Detailed task results on LLaVA-NeXT-7B.

Method	Type	GQA	MMB	MME	OK-VQA	SQA	POPE	COCO	Flickr	NoCaps	Avg.
All 2,880 Tokens (100%)											
LLaVA-NeXT-7B	Raw	64.2	67.0	1529	44.2	70.3	86.4	100.1	68.8	88.3	—
	Rel.	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%
Retain 640 Tokens (removed 77.8%)											
FastV	Raw	60.5	62.5	1514	43.2	67.9	82.6	99.9	66.4	84.0	—
	Rel.	94.2%	93.3%	99.0%	97.7%	96.6%	95.6%	99.8%	96.5%	95.1%	96.4%
DivPrune	Raw	58.1	62.8	1315	45.5	67.4	79.6	95.9	67.1	84.5	—
	Rel.	90.5%	93.7%	86.0%	102.9%	95.9%	92.1%	95.8%	97.5%	95.7%	94.5%
ZOO-Prune	Raw	61.3	64.5	1485	44.8	67.9	85.5	94.1	65.2	81.6	—
	Rel.	95.5%	96.3%	97.1%	101.4%	96.6%	99.0%	94.0%	94.8%	92.4%	96.3%
HiPrune	Raw	60.6	66.7	1490	48.7	68.0	85.1	96.5	65.3	82.7	—
	Rel.	94.4%	99.6%	97.4%	110.2%	96.7%	98.5%	96.4%	94.9%	93.7%	98.0%
TFPrune	Raw	63.9	66.5	1497	43.8	69.5	86.5	97.7	66.7	85.0	—
	Rel.	**99.5%**	**99.3%**	**97.9%**	**99.1%**	**98.9%**	**100.1%**	**97.6%**	**96.9%**	**96.3%**	**98.4%**
Retain 320 Tokens (removed 88.9%)											
FastV	Raw	55.7	60.2	1327	41.7	67.0	73.9	95.1	63.4	80.1	—
	Rel.	86.8%	89.9%	86.8%	94.3%	95.3%	85.5%	95.0%	92.2%	90.7%	90.7%
DivPrune	Raw	56.1	60.4	1278	44.9	67.3	74.4	94.2	65.2	81.7	—
	Rel.	87.4%	90.1%	83.6%	101.6%	95.7%	86.1%	94.1%	94.8%	92.5%	91.8%
ZOO-Prune	Raw	59.8	63.9	1417	42.2	67.9	83.3	90.5	61.9	77.4	—
	Rel.	93.1%	95.4%	92.7%	95.5%	96.6%	96.4%	90.4%	90.0%	87.7%	93.1%
HiPrune	Raw	57.2	64.9	1384	45.9	67.3	78.8	89.7	64.2	78.5	—
	Rel.	89.1%	96.9%	90.5%	103.8%	95.7%	91.2%	89.6%	93.3%	88.9%	93.2%
TFPrune	Raw	62.7	65.5	1506	42.3	68.4	86.3	95.4	64.7	82.4	—
	Rel.	**97.7%**	**97.8%**	**98.5%**	**95.7%**	**97.3%**	**99.9%**	**95.3%**	**94.0%**	**93.3%**	**96.6%**
Retain 160 Tokens (removed 94.4%)											
FastV	Raw	52.2	57.0	1215	40.3	66.1	68.5	89.1	61.5	77.1	—
	Rel.	81.3%	85.1%	79.5%	91.2%	94.0%	79.3%	89.0%	89.4%	87.3%	86.2%
DivPrune	Raw	53.2	57.4	1212	42.2	67.3	67.9	90.4	61.8	77.4	—
	Rel.	82.9%	85.7%	79.3%	95.5%	95.7%	78.6%	90.3%	89.8%	87.7%	87.3%
ZOO-Prune	Raw	58.2	61.9	1372	39.3	68.5	79.3	86.0	57.5	73.4	—
	Rel.	90.7%	92.4%	89.7%	88.9%	97.4%	91.8%	85.9%	83.6%	83.1%	89.3%
HiPrune	Raw	52.4	60.3	1260	39.0	69.0	68.0	69.6	50.4	60.9	—
	Rel.	81.6%	90.0%	82.4%	88.2%	98.2%	78.7%	69.5%	73.3%	69.0%	81.2%
TFPrune	Raw	61.4	65.3	1449	41.3	68.3	85.0	91.2	60.8	76.5	—
	Rel.	**95.6%**	**97.5%**	**94.8%**	**93.4%**	**97.2%**	**98.4%**	**91.1%**	**88.4%**	**86.6%**	**93.7%**

Table 10: Detailed task results on Qwen2.5-VL-7B-Instruct.

Method	Type	GQA	MMB	MME	OK-VQA	SQA	POPE	COCO	Flickr	NoCaps	Avg.
All Tokens (100%)											
Qwen2.5-VL-7B	Raw	60.9	84.1	1684	43.7	88.3	86.3	32.5	33.2	37.3	—
	Rel.	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%
Retain 33.4% Tokens											
FastV	Raw	56.1	79.7	1627	38.2	84.6	84.2	36.2	34.4	39.2	—
	Rel.	92.1%	94.8%	96.6%	87.4%	95.8%	97.6%	111.4%	103.6%	105.1%	98.3%
DivPrune	Raw	57.9	79.6	1691	37.5	82.9	81.8	31.1	31.7	35.2	—
	Rel.	95.1%	94.6%	100.4%	85.8%	93.9%	94.8%	95.7%	95.5%	94.4%	94.5%
MMTok	Raw	59.4	81.9	1694	39.3	85.4	83.9	32.7	33.5	37.5	—
	Rel.	97.5%	97.4%	100.6%	89.9%	96.7%	97.2%	100.6%	100.9%	100.5%	97.9%
TFPrune	Raw	60.2	81.4	1682	39.9	85.9	85.7	33.9	33.2	38.5	—
	Rel.	**98.9%**	**96.8%**	**99.9%**	**91.3%**	**97.3%**	**99.3%**	**104.3%**	**100.0%**	**103.2%**	**99.0%**
Retain 22.3% Tokens											
FastV	Raw	53.3	77.7	1582	35.3	81.6	81.0	35.9	32.7	38.8	—
	Rel.	87.5%	92.4%	93.9%	80.8%	92.4%	93.9%	110.5%	98.5%	104.0%	94.9%
DivPrune	Raw	57.2	77.3	1655	35.7	80.9	81.1	30.8	31.3	36.9	—
	Rel.	93.9%	91.9%	98.3%	81.7%	91.6%	94.0%	94.8%	94.3%	98.9%	93.3%
MMTok	Raw	57.9	79.9	1686	37.1	83.7	82.3	32.2	33.0	36.6	—
	Rel.	95.1%	95.0%	100.1%	84.9%	94.8%	95.4%	99.1%	99.4%	98.1%	95.8%
TFPrune	Raw	59.9	81.1	1665	39.0	85.2	85.0	33.9	33.1	38.4	—
	Rel.	**98.4%**	**96.4%**	**98.9%**	**89.2%**	**96.5%**	**98.5%**	**104.3%**	**99.7%**	**102.9%**	**98.3%**
Retain 11.2% Tokens											
FastV	Raw	47.4	66.7	1400	28.4	79.2	71.1	31.7	26.0	35.9	—
	Rel.	77.8%	79.3%	83.1%	65.0%	89.7%	82.4%	97.5%	78.3%	96.2%	83.3%
DivPrune	Raw	54.1	73.5	1557	30.3	80.4	76.4	30.1	28.3	34.7	—
	Rel.	88.8%	87.4%	92.5%	69.3%	91.1%	88.5%	92.6%	85.2%	93.0%	87.7%
MMTok	Raw	54.8	75.7	1524	31.7	81.8	78.0	30.9	30.2	35.4	—
	Rel.	90.0%	90.0%	90.5%	72.5%	92.6%	90.4%	95.1%	91.0%	94.9%	89.7%
TFPrune	Raw	58.3	79.3	1640	35.7	82.7	83.8	33.6	32.3	38.2	—
	Rel.	**95.7%**	**94.3%**	**97.4%**	**81.7%**	**93.7%**	**97.1%**	**103.4%**	**97.3%**	**102.4%**	**95.9%**

Table 11: End-to-end request latencies (s) at $K = 64$. Each value is the median of three repetition-level means. Lower is better.

Dataset	Native ($K = 576$)	FastV	TFPrune
COCO	2.271 ± 0.052	2.479 ± 0.129	$\mathbf{2.214 \pm 0.073}$
Flickr30k	3.490 ± 0.194	2.278 ± 0.063	$\mathbf{2.175 \pm 0.227}$
GQA	2.206 ± 0.043	2.304 ± 0.008	$\mathbf{2.139 \pm 0.012}$
MMBench	$\mathbf{2.225 \pm 0.043}$	3.612 ± 2.218	2.475 ± 0.389
MME	2.195 ± 0.009	2.588 ± 0.324	$\mathbf{2.133 \pm 0.016}$
NoCaps	2.530 ± 0.546	2.305 ± 0.045	$\mathbf{2.208 \pm 0.532}$
OK-VQA	2.451 ± 0.226	2.324 ± 0.111	$\mathbf{2.142 \pm 0.017}$
POPE	2.177 ± 0.033	2.279 ± 0.030	$\mathbf{2.147 \pm 0.021}$
ScienceQA-IMG	3.099 ± 0.532	2.380 ± 0.120	$\mathbf{2.208 \pm 0.664}$

B.5 Efficiency Measurement

B.5.1 Experimental environment

Experiments are conducted on NVIDIA GeForce RTX 3090 GPUs with CUDA 12.4. The LLaVA evaluation launcher supports multiple independent GPU processes, while the Qwen launcher uses a single process with `device_map=auto`. All efficiency measurements are performed on a single RTX 3090.

B.5.2 End-to-end timing

Table 5 reports end-to-end latency per sample for LLaVA-1.5 on COCO captioning. All compressed configurations prune after the first decoder block. Each run uses 20 warm-up samples followed by 500 measured samples on a single NVIDIA GeForce RTX 3090. We repeat the experiment independently three times and report the mean. Timing includes response construction, token selection, feature and KV-cache aggregation, the remaining prefill, and caption generation. Because generated captions vary in length, the reported latency reflects the evaluated workload. Throughput is computed from mean per-sample latency, and pruning time is included in end-to-end latency. Speedup is relative to the native, uncompressed model.

B.5.3 Fixed-length decoding and KV-cache measurement

Table 6 reports LLaVA-1.5 KV-cache storage after prefill and after generating 64 text tokens on COCO, GQA, POPE, and OK-VQA. Following the efficiency-test configuration, compression occurs after the first decoder block, and each run uses a single NVIDIA GeForce RTX 3090. We repeat each task and budget configuration independently three times and report the mean. At the compression boundary, KV entries for removed visual tokens are aggregated into those of the retained tokens. Cache size is reported as KV payload in MiB, rather than peak allocated GPU memory.

B.5.4 Fixed-length end-to-end comparison with FastV

We further compare end-to-end latency under fixed 64-token generation on LLaVA-1.5-7B. Each of three repetitions contains 5 warm-up requests and 100 measured requests. Table 11 reports end-to-end latency across all nine tasks, with FastV included for comparison.

Across the nine tasks, TFPrune achieves lower end-to-end latency than FastV. FastV does not consistently translate token reduction into lower end-to-end latency. The gain is clearest on COCO, GQA, and MME. These fixed-length results illustrate that token reduction alone need not translate into a wall-clock speedup, while TFPrune can offset its compression overhead in this low-budget setting.

B.6 Hyperparameter Sensitivity

We study four hyperparameters on LLaVA-1.5-7B at $K = 128$: the maximum transfer coefficient $\alpha_{\max}$, functional-response dimension, number of text probes, and selection seed. Following the sen-

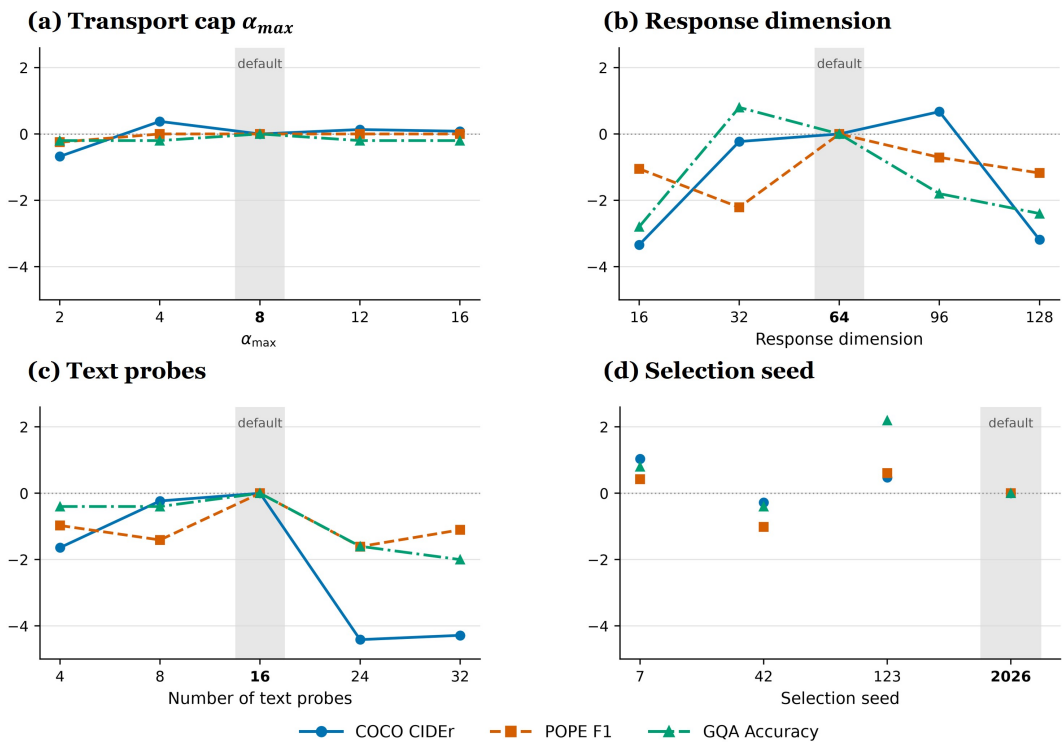


Figure 5: Sensitivity to (a) the transfer cap, (b) response dimension, (c) number of text probes, and (d) selection seed. Scores are shown relative to the highlighted reference setting.

sitivity script, we evaluate COCO, POPE and GQA while varying one parameter at a time. Figure 5 shows the change in each task score relative to the reference setting marked in gray.

Performance changes little as $\alpha_{\max}$ varies from 2 to 16, suggesting that transfer is not sensitive to its coefficient cap in the tested range. Increasing the response dimension or the number of text probes does not consistently improve performance. Larger settings can reduce scores on some tasks, potentially because they add redundant or noisy functional directions or text-conditioned information. The seed sweep likewise shows no setting that improves all three tasks. Overall, the reference response dimension of 64 and 16 text probes provide a practical balance without requiring larger representations.

B.7 Additional Case Studies

Figure 6 provides further examples of the visual regions selected by the two methods and emphasized by later reference attention.

C Method Details

The following derivations and implementation details expand the definitions in Section 3. They describe a local attention-response criterion and its use for coverage and aggregation.

C.1 Functional Response Construction

C.1.1 Evaluation Context

Let $X \in \mathbb{R}^{n \times d}$ be the hidden sequence at the selected insertion point, immediately before an available decoder block. Its token-wise input normalization gives $\bar{x}_j$. The following attention operators are evaluated on these current states. They do not require previously generated outputs from that

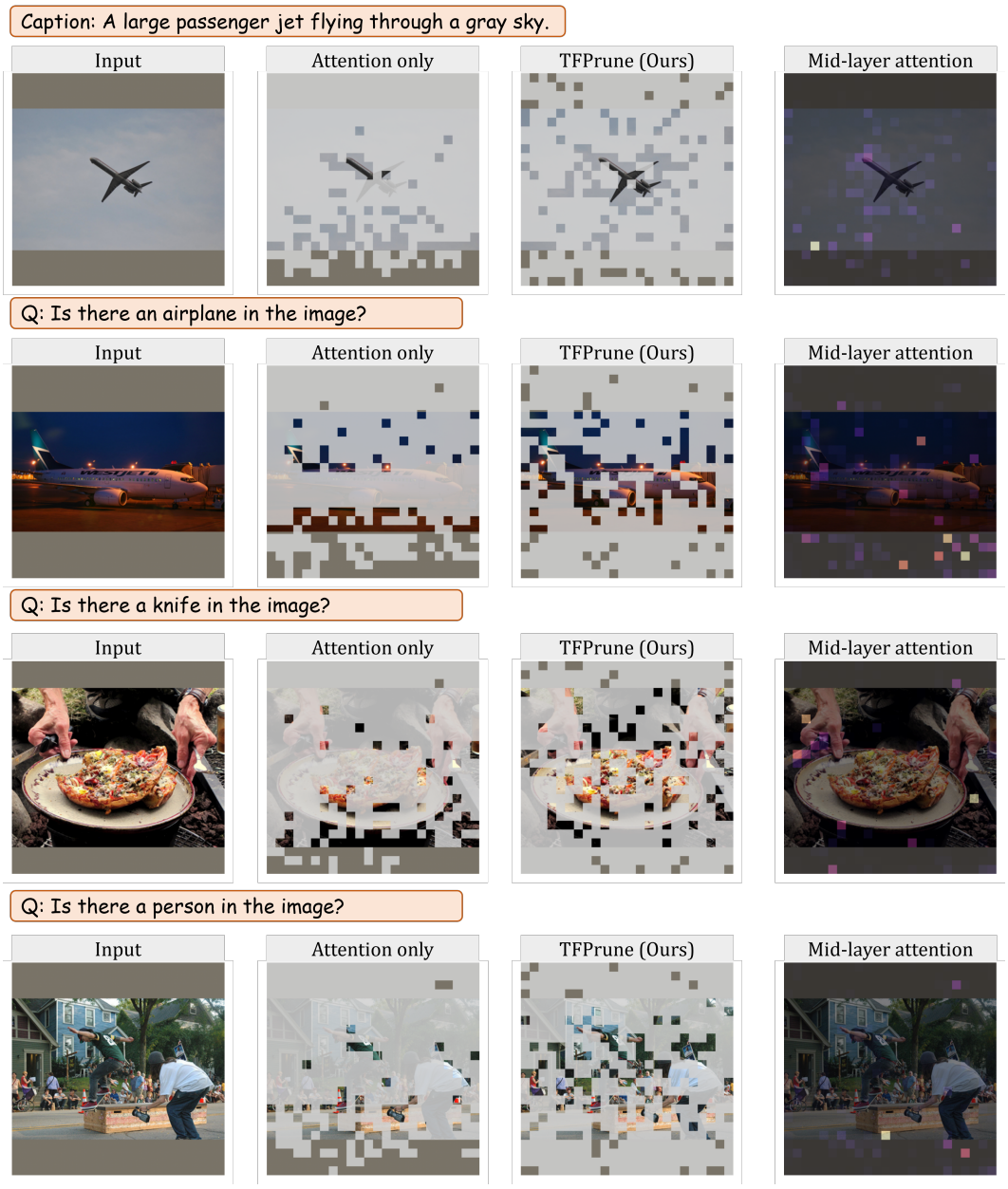


Figure 6: Additional visual-token selection examples comparing early attention-only selection, TF-Prune, and reference attention.

block. During single-token removal, other normalized states, original positions, query positions, selected heads, and projection matrices remain fixed.

The response is local to attention. At a surviving text position, the residual input is unchanged by removing a visual key-value entry. Its post-attention hidden difference is therefore the full attention-output difference before the subsequent FFN. The FFN is position-wise but nonlinear, so projecting or reducing this attention difference does not by itself bound the final block output or task loss. The functional descriptor is used as a compression criterion rather than a certified full-network influence estimate.

C.1.2 TEXT POSITIONS AND HEAD SELECTION

The implementation selects the last P available text positions following the visual span to form $\mathcal{T}$. If none are available, it uses preceding text positions. Preceding queries cannot attend to later visual keys under a causal mask, so their visual removal responses vanish and only the auxiliary value component can remain informative. The primary setting contains instruction positions after the image.

Let H be the number of query heads, H_{kv} the number of KV heads, and d_h the head dimension. Let $q^h_{t,0}$ denote the unrotated query for position t and head h. The head-selection statistic is

$$e_h = \frac{1}{|\mathcal{T}| d_h} \sum_{t \in \mathcal{T}} \|q^h_{t,0}\|_2^2. \tag{7}$$

When the head budget H_s is smaller than H, the set $\mathcal{H}$ contains the H_s heads with the largest e_h. The energy score measures query activation magnitude rather than established causal importance. A zero head-budget setting evaluates all heads, in which case $\mathcal{H}$ contains every query head and ranking does not affect the response. Grouped-query attention reuses each selected query head's native KV group. Queries are evaluated at $\mathcal{T}$, whereas keys and values include the full accessible context.

With native Q/K projections, rotary positions, and accessibility mask M, attention probabilities are

$$a^h_{tj} = \mathrm{softmax}_j\left((q^h_t)^\top k^h_j / \sqrt{d_h} + M_{tj}\right). \tag{8}$$

The denominator includes accessible textual and visual positions. RoPE and multimodal RoPE retain their original coordinates.

C.1.3 COMPACT MULTI-HEAD OUTPUT

Let O_h denote head h's block of the attention output projection, and let $W_V^{g(h)}$ denote the value projection of its associated KV group. Let $G \in \mathbb{R}^{d \times r}$ have fixed Gaussian entries drawn from $\mathcal{N}(0, 1/r)$. The cached composite operator is

$$C_h = G^\top O_h W_V^{g(h)}. \tag{9}$$

The compact value at any context position is $u^h_j = C_h \bar{x}_j$. This directly evaluates the value and output projections in the response space. The per-head attention output is

$$y^h_t = \sum_{j=1}^{n} a^h_{tj} u^h_j. \tag{10}$$

Writing $c_H = H/H_s$, the operator used in Eq. 1 is

$$\mathcal{A}_t(X) = c_H \sum_{h \in \mathcal{H}} y^h_t. \tag{11}$$

Thus a single operator in the main text combines all evaluated heads after their output projections. It is a restricted, projected MHA output. Deterministic head selection does not make c_H an unbiased correction for excluded heads.

The composite in Eq. 9 omits an affine value bias. A constant projected value bias appears in both u^h_i and y^h_t and cancels in their difference, leaving the deletion response unchanged. The auxiliary value descriptor is bias-free, as in the implementation. A constant output bias also cancels in output differences.

C.1.4 EXACT SINGLE-TOKEN REMOVAL IDENTITY

Fix t, h, and a visual token i. Assume the remaining context has nonzero softmax weight, so $a^h_{ti} < 1$. Removing only token i renormalizes every remaining probability by $1/(1 - a^h_{ti})$. The resulting output is

$$y^{h,\setminus i}_t = \frac{y^h_t - a^h_{ti} u^h_i}{1 - a^h_{ti}}. \tag{12}$$

Subtracting this expression from y_t^h gives the exact difference

$$\Delta_{ti}^h = \frac{a_{ti}^h}{1 - a_{ti}^h}(u_i^h - y_t^h). \tag{13}$$

Combining Eq. 13 across the fixed selected heads yields

$$f_i = \mathrm{vec}_{t\in\mathcal{T}}\left(c_H \sum_{h\in\mathcal{H}} \Delta_{ti}^h\right). \tag{14}$$

This is exactly Eq. 1 before numerical safeguards. It requires shared probabilities and projected values rather than a separate block evaluation for every i. A large attention probability alone does not imply a large removal response because the value difference $u_i^h - y_t^h$ also matters. The identity concerns individual removals with fixed inputs and does not assert additivity of simultaneous deletions.

C.1.5 Stabilization and Component Normalization

The implementation replaces the prefactor in Eq. 13 by

$$\gamma_{ti}^h = \min\left(\frac{a_{ti}^h}{\max(1 - a_{ti}^h, \epsilon)}, \gamma_{\max}\right). \tag{15}$$

The resulting response uses $\widehat{\Delta}_{ti}^h = \gamma_{ti}^h(u_i^h - y_t^h)$ in Eq. 14. It equals the exact identity when the safeguards are inactive and is a stabilized approximation otherwise. We retain the symbol f_i for this numerically evaluated descriptor in the implementation.

The complementary value descriptor is

$$p_i = c_H \sum_{h\in\mathcal{H}} u_i^h. \tag{16}$$

Let $\mathcal{I}_v$ index all N_v positions in the registered visual span, including mandatory visual separators where applicable. For a component $b_i \in \mathbb{R}^{d_b}$, the scale in Eq. 2 is implemented as

$$\sigma_b = \max\left(\sqrt{\frac{1}{N_v}\sum_{i\in\mathcal{I}_v} \|b_i\|_2^2},\ \epsilon\sqrt{d_b}\right). \tag{17}$$

Setting $b = f$ or $b = p$ defines σ_f and σ_p. Away from the floor, each normalized component has unit average squared token norm. This is a scalar normalization shared across tokens, not unit normalization of each token. It preserves relative token magnitudes within the component. The common factor c_H cancels when the floor is inactive.

The final response has dimension $D_f = (|\mathcal{T}| + 1)r$. Mandatory visual separators participate in the builder's normalization but are excluded from candidate selection. The additional attention-probability component supported by the implementation is disabled in the reported configuration.

C.2 Functional Coverage and Reconstruction

C.2.1 Selection Metric and Classical Covering Guarantee

Let $J \in \mathbb{R}^{D_f\times s}$ be a fixed Rademacher matrix with entries $\pm 1/\sqrt{s}$, and let $\psi_i = J^\top\phi_i$. If selection projection is disabled, $\psi_i = \phi_i$. For nonzero ψ_i, define

$$d_f(i,j) = \left\| \frac{\psi_i}{\|\psi_i\|_2} - \frac{\psi_j}{\|\psi_j\|_2} \right\|_2. \tag{18}$$

The procedure initializes with a maximum-norm original response and repeatedly selects a farthest point in this metric. Nearest-representative assignments are updated during traversal. Each selected token is finally assigned to itself. The original, unprojected vectors ϕ_i are retained for computing the transfer weights.

Proposition 1 (covering radius). Assume nonzero normalized selection vectors and one arbitrary initial center followed by farthest-first traversal. Let $R_f(S) = \max_i \min_{j \in S} d_f(i,j)$. If S_K is the selected set and S^* minimizes Eq. 3, then

$$R_f(S_K) \leq 2R_f(S^*). \tag{19}$$

Proof. If the final radius is zero, the result is immediate. Otherwise, the K centers and a final farthest point form $K+1$ points whose pairwise distances are at least $R_f(S_K)$. At least two share an optimal center among K centers. Their distance is at most $2R_f(S^*)$ by the triangle inequality. This proves Eq. 19. This is the classical metric k-center guarantee (Gonzalez, 1985).

The bound holds in the projected selection geometry and does not guarantee downstream task accuracy. Optional relevance anchors are outside the stated initialization and are disabled in the reported configuration. Floating-point norm floors and tie-breaking can affect traversal. In particular, the implementation uses $2 - 2\cos$ for squared distances, whose metric interpretation requires nonzero normalized vectors.

C.2.2 Directional Coverage Controls Nonnegative Reconstruction

Let d_0 denote Eq. 18 using original functional vectors ϕ instead of ψ. For any nonzero pair, write c_{ij} for their cosine similarity and $c_{ij,+} = \max(c_{ij}, 0)$. Let α_{ij}^{∞} minimize $\|\phi_i - \alpha\phi_j\|_2$ over $\alpha \geq 0$ without an upper bound.

Proposition 2 (relative reconstruction). In exact arithmetic with nonzero vectors, the optimal relative residual is

$$\frac{\|\phi_i - \alpha_{ij}^{\infty}\phi_j\|_2^2}{\|\phi_i\|_2^2} = 1 - c_{ij,+}^2. \tag{20}$$

It is bounded by the squared directional distance, giving

$$\frac{\|\phi_i - \alpha_{ij}^{\infty}\phi_j\|_2}{\|\phi_i\|_2} \leq d_0(i,j). \tag{21}$$

Proof. With $v = \alpha\|\phi_j\|_2/\|\phi_i\|_2$, the relative squared residual is $1 - 2c_{ij}v + v^2$. Its constrained minimum occurs at $v = c_{ij,+}$, proving Eq. 20. For $c_{ij} \geq 0$, $1 - c_{ij}^2 \leq 2(1 - c_{ij})$. For $c_{ij} < 0$, the residual is one and $2(1 - c_{ij}) > 2$. Since $d_0(i,j)^2 = 2(1 - c_{ij})$, Eq. 21 follows.

Consequently, if selection uses d_0 and the coefficient upper bound is inactive, the maximum relative response residual is bounded by the selected covering radius. This connects coverage and functional fitting without claiming equality of merged hidden-state behavior.

C.2.3 Coefficient Clipping and Projection Distortion

The same calculation quantifies the effect of a finite upper bound. Let α_{ij}^{A} be the constrained solution for $0 \leq \alpha \leq A$, and let $b_{ij} = A\|\phi_j\|_2/\|\phi_i\|_2$. Then

$$\frac{\|\phi_i - \alpha_{ij}^{A}\phi_j\|_2^2}{\|\phi_i\|_2^2} = 1 - c_{ij,+}^2 + \left(c_{ij,+} - b_{ij}\right)_+^2. \tag{22}$$

Here $(z)_+ = \max(z, 0)$. For $c_{ij} \leq 0$, the optimum remains zero. For positive c_{ij}, minimizing the quadratic over $v \in [0, b_{ij}]$ gives $v = \min(c_{ij}, b_{ij})$, which proves Eq. 22.

To distinguish the selection geometry from the fitting geometry, let $\pi_f(i)$ be the assignment produced under d_f, and assume the relevant projected vectors are nonzero. Define the directional distance discrepancy

$$\eta = \max_{i,j \in \mathcal{V}} |d_0(i,j) - d_f(i,j)|. \tag{23}$$

For $j = \pi_f(i)$ and $A = \alpha_{\max}$, write $q_i = (c_{ij,+} - b_{ij})_+$. Applying Eqs. 21–23 gives

$$\max_i \frac{\|\phi_i - \alpha_i\phi_{\pi_f(i)}\|_2}{\|\phi_i\|_2} \leq \sqrt{\left(R_f(S) + \eta\right)^2 + \max_i q_i^2}. \tag{24}$$

This deterministic bound separates coverage, projection distortion, and coefficient clipping. It assumes the norm-denominator floor is inactive and uses the constrained coefficients in Eq. 4. The

quantity η is not presumed small or measured by the algorithm. Thus the result supplies an explicit conditional connection, not a dimension-independent guarantee for an arbitrary projection. Constant-dimensional random projection alone is insufficient to claim unchanged original-space coverage.

C.3 Functional Transport and Attention Calibration

C.3.1 Closed-Form Functional Fit

For a nonzero representative vector, solving Eq. 4 gives the nonnegative projection coefficient clipped to $\alpha_{\max}$. The implementation adds a norm floor:

$$\alpha_i = \mathrm{clip}_{[0,\alpha_{\max}]}\left(\frac{\phi_i^\top \phi_{\pi(i)}}{\max(\|\phi_{\pi(i)}\|_2^2, \epsilon)}\right). \tag{25}$$

The closed form follows by differentiating the convex quadratic and projecting its unconstrained minimizer onto the interval. It exactly solves the stated fitting problem when the denominator floor is inactive. Selected representatives have self-weight one. Hence each accumulated weight m_j is at least one. A zero coefficient transfers no feature from that token.

C.3.2 Weighted Feature Reconstruction

For fixed nonnegative transfer weights and an assignment group C_j, define

$$L_j(z) = \sum_{i \in C_j} \alpha_i \|x_i - z\|_2^2. \tag{26}$$

Proposition 3 (weighted representative). If $m_j > 0$, the unique minimizer of L_j is $\widetilde{x}_j$ in Eq. 5.

Proof. The gradient is $2m_j z - 2\sum_i \alpha_i x_i$. Its zero is the weighted average, and the Hessian is $2m_j I$, which is positive definite. This establishes optimality in the fixed weighted feature objective. It does not imply optimality for the downstream task loss or for nonlinear attention outputs. Functional fitting chooses the weights, while feature pooling solves this separate aggregation objective.

C.3.3 Accumulated Weight and Proportional Attention

Let b_{tj}^h be the usual attention logit including its accessibility mask. Assign each retained visual representative its accumulated weight, with unit weight for unaffected and newly generated positions. The corrected attention is

$$\widetilde{a}_{tj}^h = \mathrm{softmax}_j\left(b_{tj}^h + \log m_j\right). \tag{27}$$

This is proportional attention (Bolya et al., 2023). The implementation calls m_j an effective mass. Mathematically, it is the total nonnegative transfer weight, not an information-theoretic measure of visual content.

Exact weighted-group interpretation. Fix a query and head. Suppose every member of C_j is accessible and shares the same logit b_j, with a representative having that same key. Give member i weight α_i. The group's contribution to the weighted attention denominator is $m_j e^{b_j}$. Its numerator contribution is $e^{b_j}\sum_i \alpha_i v_i$. A representative with logit $b_j + \log m_j$ and value $m_j^{-1}\sum_i \alpha_i v_i$ has exactly these two contributions. Replacing such groups therefore preserves the weighted attention output.

This is equality with the weighted measure. It coincides with the original unweighted measure when all member weights equal one. In general, a pooled hidden state passes through nonlinear normalization and yields a different key and value. Group members can also differ in key, position, and accessibility. The log-weight correction therefore provides an approximate calibration in the deployed model rather than exact preservation of its original attention.

C.4 Efficiency and Integration

C.4.1 Response Computation

Only selected text positions require query projection. Their queries are projected across all heads for the norm-based head ranking. Key projection over the full sequence is restricted to the $U \leq H_s$ distinct KV groups used by the selected heads. Cached operators in Eq. 9 produce compact values for every accessible text and visual position.

Let d be hidden width, r the compact output dimension, and n sequence length. Warm-cache response evaluation has cost

$$O\big(Pd^2 + ndUd_h + nH_s dr + H_s Pn(d_h + r)\big)\,. \tag{28}$$

The terms account for query projection, selected-group keys, compact values, and attention/output evaluation. Composing uncached operators costs $O(H_s d d_h r)$. The implementation caches operators by module, head, KV group, output dimension, and device, assuming fixed weights and projection. Storage includes $O(H_s Pn)$ probabilities, $O(H_s nr)$ compact values, and an explicitly materialized $O(H_s PN_v r)$ value-difference tensor. Cached operators require $O(H_{\text{cached}} dr)$ elements per evaluated block.

C.4.2 Selection and Aggregation

Dense projection from dimension D_f to s costs $O(ND_f s)$. Streaming farthest-first traversal costs $O(NKs)$ time and $O(Ns + N)$ storage. A precomputed cosine matrix costs $O(N^2 s + NK)$ time and $O(N^2 + Ns)$ storage. Precomputation improves the organization of GPU operations by replacing repeated matrix-vector products with a matrix multiplication, rather than reducing asymptotic arithmetic.

With the script defaults, the automatic backend uses precomputation for $N \leq 1{,}024$, budgets of at least 16, and an fp32 similarity matrix fitting a 64 MiB allowance. Both backends implement the same traversal in exact arithmetic. Optional CountSketch reduces projection work to $O(ND_f)$ but changes the selection geometry and its distortion properties.

Functional coefficient evaluation costs $O(ND_f)$, and indexed hidden-state aggregation costs $O(Nd)$. Already populated KV caches are aggregated separately with the same weights. The cache aggregation work also depends on the number of populated decoder blocks at the insertion boundary. Earlier insertion reduces the number of existing blocks requiring this operation.

The budget, insertion depth, response dimensions, number of text queries, and selected-head count provide explicit performance–efficiency controls. They should be assessed together with response and cache overhead when reporting end-to-end latency. The sequence reduction alone does not establish a numerical wall-clock speedup.

C.5 Budgets and Model Integration

C.5.1 Budget Accounting

The default `compressible` scope counts visual content candidates. With N_{struct} mandatory visual separators, a content budget K retains $K + N_{\text{struct}}$ total visual positions. Structural positions are not selected or merged. The optional `all_visual` scope includes these positions in the requested total and rejects budgets leaving no content candidate. Text positions are retained. These details do not alter the functional selection rule, but content-token and total-sequence retention ratios must be distinguished.

A proportional budget uses $\max(1, \text{round}(\rho N))$, capped by availability. This makes token count depend on input sequence length at a prescribed ratio. It does not imply that the ratio itself adapts to semantic complexity. The optional adaptive stopping rule uses $1 - R_f(S)/2 \geq \tau$ after a minimum count, bounded by a maximum count. The parameter `functional_vepr_target` supplies τ in this minimax path. It is a geometric threshold rather than task accuracy or a measured fraction of preserved information.

C.5.2 INSERTION AND PERSISTENT COMPRESSION

The general method applies at a supported decoder boundary with an available following attention block. In the early configuration, `after_observe` with `observe_layers=1` compresses after the first block and evaluates responses with the second block's operators. The separate `pre_decoder` configuration uses the first block's operators before any block has run. The final decoder output boundary has no following block and is outside the described response construction.

Representatives retain their original order and position identifiers. Existing visual keys and values in populated blocks are averaged using the same transfer weights. Rotated keys are aggregated as stored. Later blocks build their caches from the compressed sequence. The compressed visual positions are maintained during subsequent prefill and decoding, and newly generated positions receive unit attention weight. This compresses the prompt cache without replaying earlier blocks.

The method uses decoder attention operators and hidden states. The supplied adapters support LLaVA-1.5, LLaVA-NeXT, and Qwen2.5-VL, including grouped-query attention and multimodal RoPE. The current runtime assumes batch size one and a registered visual span. Full-budget no-op requests bypass response construction and selection. The vision encoder itself is unchanged.

C.5.3 MAIN-EXPERIMENT SCRIPT CONFIGURATION

The reported TFPrune main experiments use `after_observe` with `OBSERVE_LAYERS=1`, compressing after the first decoder block and evaluating the second block's attention operators. The shared launchers default to four observed blocks, so first-block runs must pass `OBSERVE_LAYERS=1` explicitly. The remaining settings use a maximum of $P = 16$ text probes, compact output dimension to $r = 64$, and dense selection-projection dimension to $s = 128$. The head budget is zero, which the implementation interprets as evaluating all query heads rather than selecting no heads. Both the removal response and auxiliary value component are enabled, giving $D_f \leq (16 + 1)64 = 1{,}088$. The additional attention-probability component and relevance anchors are disabled. The scripts also set $\alpha_{\max} = 8$, $\gamma_{\max} = 20$, $\epsilon = 10^{-8}$ in the builder. The code calls the removal-response builder `build_counterfactual`.

C.6 ABLATION CONFIGURATION

The ablation study in Table 4 uses LLaVA-1.5 at $K = 64$ with the first-block insertion point. The *Full* variant enables the removal response, auxiliary value descriptor, minimax selection, functional transport, effective mass, and attention bias. *w/o value component* sets `FUNCTIONAL_VALUE_POTENTIAL=false`. *Average coverage* sets `FUNCTIONAL_SELECTION_MODE=coverage`, replacing minimax traversal with a greedy total-coverage objective. *w/o attention bias* sets `MASS_ATTENTION_BIAS=false` while retaining transport and effective mass. *w/o functional transport* sets `FUNCTIONAL_TRANSPORT=false` but retains effective mass and attention bias. *Hard pruning* disables transport, effective mass, and attention bias. Both variants retain the same functional selection criterion, but only the former keeps the effective-mass attention correction. All other switches follow the `run_ablation_llava15.sh` configuration.