BEYOND ONE-SIZE-FITS-ALL: SAMPLE-ADAPTIVE STRATEGY ROUTING FOR VISUAL TOKEN PRUNING IN MLLMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Multimodal large language models (MLLMs) process hundreds or thousands of visual tokens per image, incurring substantial inference costs. While existing visual token pruning methods mitigate this overhead, they implicitly assume that a single fixed pruning strategy can be applied uniformly across all inputs. Our analysis reveals that ranking pruning methods by average benchmark accuracy conceals substantial sample-wise complementarity: although the average-best strategy excels overall, alternative strategies prove superior on a considerable fraction of individual samples. To harness this diversity, we propose **VIP-Router**, a lightweight **VI**sion **P**runing **Router** that, conditioned on low-cost visual and textual features, selects for each input the pruning strategy predicted to be most suitable at a specified pruning level, while retaining full-token inference as a fallback when pruning is predicted to be unfavorable. On VTC-Bench Group A, a pruning-sensitive suite spanning eight vision-language benchmarks, VIP-Router consistently outperforms the best fixed pruning strategy at every reduction ratio, with relative gains of 26.9% in average accuracy and 22.0% in average utility, a cost-aware metric that penalizes each method for the visual tokens it actually retains. Under stricter image-disjoint splits, VIP-Router still improves average utility over the best fixed strategy by 4.7% in relative terms, with gains in all 15 runs (three splits × five seeds). Moreover, VIP-Router is plug-and-play: it modifies neither the pruning algorithms nor the model weights and adds only 1.38M trainable parameters (0.017% of the backbone). It also remains effective on four additional MLLMs and yields gains on four unseen benchmarks, highlighting the potential of sample-adaptive routing for visual token pruning.

1 INTRODUCTION

Multimodal large language models (MLLMs) have demonstrated impressive capabilities across a broad spectrum of vision-language tasks (Liu et al., 2024a; Li et al., 2024a; Wang et al., 2024; Bai et al., 2025; Zhu et al., 2025). However, the high computational cost of processing visual tokens remains a critical bottleneck for their practical deployment: an image can introduce hundreds or even thousands of tokens into the language model, dominating both memory and latency (Jin et al., 2025). To alleviate this overhead, visual token pruning has emerged as a mainstream and effective approach (Chen et al., 2025; Shang et al., 2025; Yang et al., 2025; Wen et al., 2025). These methods leverage diverse cues, including attention signals (Chen et al., 2025; Shang et al., 2025), token dominance (Yang et al., 2025), and token duplication (Wen et al., 2025), to discard redundant tokens.

However, VTC-Bench (Liao et al., 2026) shows that benchmarks commonly used to evaluate pruning methods were designed for general perception rather than stress-testing visual token pruning. It therefore constructs Group A, a pruning-sensitive subset of samples that are answered correctly with full tokens but incorrectly after equivalent-ratio image downsampling. On Group A, we observe that no single pruning strategy dominates across all benchmarks: as shown in Figure 1 (a), different strategies excel on different tasks, indicating a non-trivial degree of complementarity. More importantly, ranking pruning methods by average benchmark performance also obscures complementarity at the sample level. Among samples for which at least one pruning strategy succeeds, over one-third

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.

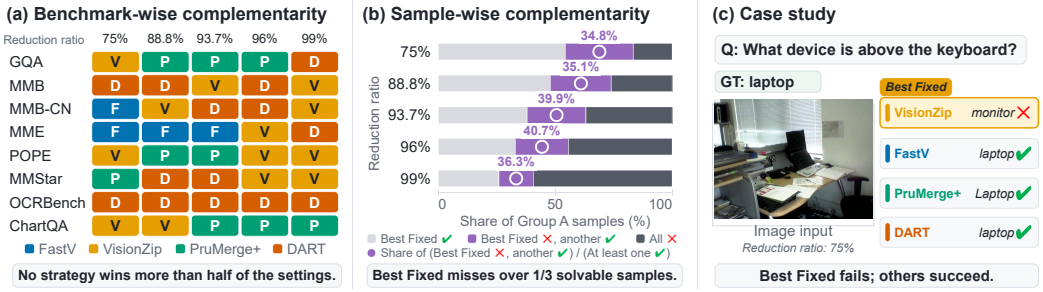


Figure 1: **Complementarity among pruning strategies.** Results are computed on all Group A samples of VTC-Bench with Qwen2-VL-7B, not only on the test split. (a) Accuracy-best strategy for each benchmark and reduction ratio (F: FastV, V: VisionZip, P: PruMerge+, D: DART). (b) Stacked bars split Group A samples by whether Best Fixed answers correctly, fails while another strategy succeeds, or all strategies fail; purple dots show the share of the second category among samples that at least one strategy answers correctly. (c) A GQA sample at 75% reduction on which Best Fixed (VisionZip) fails while the other three strategies succeed.

are cases where Best Fixed, the single strategy with the highest average performance, fails but an alternative strategy answers correctly (Figure 1 (b)). In other words, the strategy that performs best on average can still be suboptimal for a substantial fraction of inputs: **average-best does not imply sample-best**.

Therefore, rather than designing yet another heuristic to pursue higher average performance, we treat this complementarity as an untapped source of improvement and exploit it through sample-adaptive selection. The main challenge is that the sample-best strategy depends on whether each strategy yields a correct answer, which is unknown before inference. Determining it exactly would require running every candidate and verifying its output, negating the very computational savings that pruning is meant to provide. Yet, intuitively, the suitability of a pruning strategy should depend on the visual and linguistic characteristics of the input. Our key finding is that low-cost visual and textual features already provide an informative signal for predicting strategy suitability.

To this end, we propose **VI**sion **P**runing **Router** (**VIP-Router**), a lightweight, plug-and-play module compatible with existing pruning methods. VIP-Router builds a query-conditioned preview representation from visual and textual features via text-to-vision cross-attention and scores each candidate with a ratio-conditioned utility predictor. At inference time, only the highest-scoring candidate is executed; full-token inference is included as an explicit candidate and is selected when pruning is predicted to be unfavorable.

In our evaluations, VIP-Router consistently outperforms Best Fixed across all five reduction ratios on VTC-Bench Group A. Even after its realized token cost is charged, VIP-Router improves average utility from 30.88 to 37.68, corresponding to a 22.0% relative improvement over Best Fixed; meanwhile, average accuracy increases from 40.35% to 51.19%, a 26.9% relative gain. Under a stricter image-disjoint protocol, VIP-Router continues to outperform Best Fixed on three independently constructed splits, improving average utility by 4.7% in relative terms (3.7–5.6% per split), with gains in all 15 runs (three splits × five seeds). These gains are obtained without modifying either the candidate pruning methods or the MLLM, and VIP-Router adds only 1.38M trainable parameters (0.017% of the backbone).

Our main contributions are summarized as follows:

- We characterize the sample-level complementarity among existing visual token pruning strategies, revealing that the average-best strategy is suboptimal on a substantial fraction of samples, a fact masked by standard aggregate evaluation.
- We introduce VIP-Router, which uses a single ratio-conditioned predictor shared across pruning levels to score all candidates, including full-token inference, under a cost-aware utility objective.

- Extensive experiments show that VIP-Router consistently improves accuracy and utility over Best Fixed at every pruning level, and that these gains persist under image-disjoint splits, across four additional MLLM backbones, and in zero-shot transfer to four unseen benchmarks.

2 Related Work

2.1 Visual token pruning for MLLMs

Visual token pruning is a widely studied approach to improving MLLM efficiency by removing or consolidating redundant visual tokens before or within the language model. Existing methods primarily differ in the criterion used to identify which visual information should be retained. Attention-based methods such as FastV and PruMerge exploit attention signals from the language model or visual encoder to identify salient tokens (Chen et al., 2025; Shang et al., 2025), while FitPrune (Ye et al., 2025a) derives pruning configurations by matching attention statistics before and after pruning. Other methods explicitly target redundancy among visual representations: VisionZip (Yang et al., 2025) preserves dominant tokens while compressing contextual ones, DivPrune (Alvar et al., 2025) promotes diversity among retained tokens, and DART (Wen et al., 2025) removes tokens according to their duplication with representative pivots. Recent work has also examined failure modes of pruning; for example, RVIS (Kim et al., 2026) adapts pruning during decoding in response to changes in task-relevant visual information.

Despite these differences, each existing method applies a single, fixed criterion to all inputs. VIP-Router instead treats existing pruning strategies as complementary candidates and performs sample-level selection among them.

2.2 Adaptive Inference for Efficient MLLMs

Adaptive inference adjusts computation according to the input or runtime efficiency requirements rather than committing to a single fixed configuration (Zhou et al., 2026). Within MLLMs, existing methods mainly adapt the amount, location, or form of visual computation. Dynamic-LLaVA (Huang et al., 2025) sparsifies visual and linguistic contexts during prefill and decoding; ATP-LLaVA and SparseVLM (Ye et al., 2025b; Zhang et al., 2025) predict input- or layer-dependent visual-token retention ratios; AIM (Zhong et al., 2025) combines pre-LLM merging with progressive in-LLM pruning; and F^3A (Huang et al., 2026) uses question-conditioned signals to adapt visual-token allocation at a specified pruning level. OccamToken (Li et al., 2026a) performs training-free, sample-adaptive visual token pruning and calibrates its pruning threshold to match a target average retained-token budget. M^3 (Cai et al., 2024), MQT (Hu et al., 2024), and AdaLLaVA (Xu et al., 2025) further adapt representation granularity or the execution pathway. These approaches do not explicitly select among distinct pruning criteria for each sample. A more closely related line of work learns a selector over candidate inference options. RouteLLM (Ong et al., 2024) routes queries between LLMs to balance quality and cost, while MetaKV (Wang et al., 2026) selects among KV-cache compression configurations based on predicted efficiency and correctness. Within MLLM compression, QMoP (Li et al., 2026b) uses a query-guided router over learned compression branches. Unlike QMoP, VIP-Router leaves existing pruning operators unchanged and makes a discrete selection among off-the-shelf strategies.

Overall, prior adaptive methods mainly vary how much computation is used, where it is allocated, or which representation or execution pathway is adopted. VIP-Router operates on a different adaptation axis: which pruning criterion to apply to each sample at a specified pruning level.

3 Vision Pruning Router

VIP-Router selects a pruning strategy per sample using three components (Figure 2): frozen visual and textual preview encoders, a text-to-vision cross-attention module, and a ratio-conditioned MLP that predicts the utility of every candidate at the given pruning level.

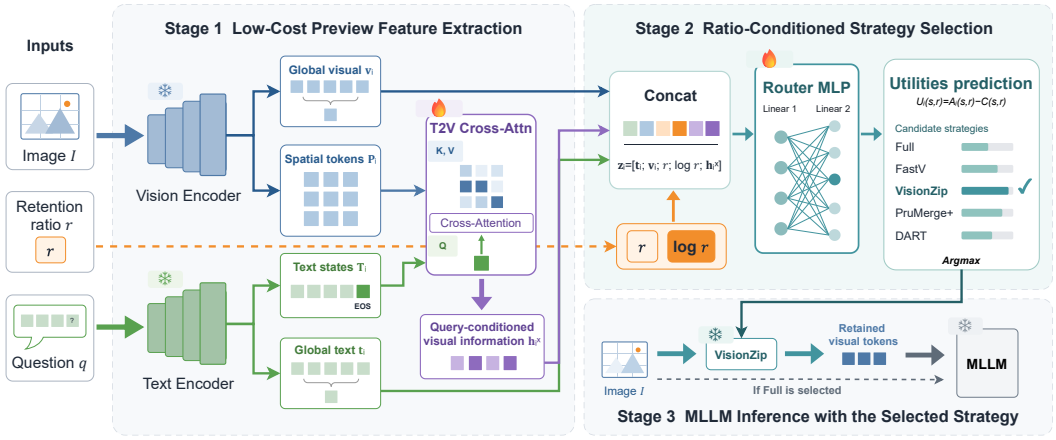


Figure 2: **Overview of VIP-Router.** Snowflakes mark frozen modules and flames trainable ones. Frozen visual and textual preview features, together with query-conditioned cross-modal features and the retained-token ratio, are used to predict the utility of each candidate pruning strategy and of Full; only the highest-utility candidate is then executed.

3.1 Problem Formulation

We adopt the Group A setting of VTC-Bench (Liao et al., 2026), which retains only samples for which full-token inference is correct but the equivalent-ratio image downsampling baseline fails. Given a sample $x_i = (I_i, q_i)$, we select a strategy from $\mathcal{S} = \{\text{Full}, \text{FastV}, \text{VisionZip}, \text{PruMerge+}, \text{DART}\}$. The candidate Full retains all visual tokens, while the remaining candidates apply existing pruning rules.

Let $r \in (0, 1]$ denote the retained-token ratio used by the pruning candidates; the corresponding reduction ratio is $1 - r$. Here, r specifies the target pruning operating point rather than a hard per-sample token constraint, since VIP-Router may select Full when pruning is predicted to be unfavorable. Under strategy s, let $a_i^s \in \{0, 1\}$ indicate whether the MLLM answers sample i correctly.

We define

$$U_i(s, r) = a_i^s - c(s, r), \qquad c(s, r) = \begin{cases} 1, & s = \text{Full}, \\ r, & s \neq \text{Full}. \end{cases} \tag{1}$$

Under Group A, this objective ranks a correctness-preserving pruning strategy above Full, and Full above an incorrect pruning strategy, so that Full serves as a cost-aware fallback.

The oracle sample-level decision is

$$s_i^\star = \arg\max_{s \in \mathcal{S}} U_i(s, r). \tag{2}$$

At test time, these utilities are unknown. We therefore learn a router g_θ that jointly predicts the utility of all candidates:

$$\hat{\mathbf{u}}_i = g_\theta(I_i, q_i, r), \qquad \hat{s}_i = \arg\max_{s \in \mathcal{S}} \hat{u}_{i,s}. \tag{3}$$

Under Group A, $U_i(\text{Full}, r) \equiv 0$ while the Bayes-optimal prediction of a pruning candidate's utility is $\Pr(a_i^s = 1 \mid z_i) - r$, so utility regression amounts to estimating conditional pruning success and comparing it against the known cost: Full is preferred only when every pruning candidate is predicted to preserve correctness with probability below r (Appendix A.1). Two consequences follow. First, a predictor that observes only the router's input features should select Full far less often than the label-aware oracle in Equation 2, which selects Full whenever all pruning candidates fail. Second, the margin between Full and a failed pruning candidate is only r, so such decisions become increasingly sensitive to noise as pruning becomes more aggressive.

3.2 Cross-Modal Preview Representation

We instantiate the preview representation using frozen CLIP visual and text encoders. For image I_i, we extract penultimate-layer visual representations, yielding spatial patch features $\mathbf{P}_i$, whose mean gives the global visual feature $\mathbf{v}_i$. For question q_i, the text encoder produces token representations $\mathbf{T}_i$ and a global text representation $\mathbf{t}_i$.

While $\mathbf{v}_i$ and $\mathbf{t}_i$ provide low-cost global summaries, their direct concatenation captures only coarse cross-modal interactions. We therefore introduce a lightweight text-to-vision cross-attention module. After projecting both modalities to a common dimension, the text tokens query the spatial visual features:

$$\mathbf{H}_i = \mathrm{MHA}\left(Q = \mathbf{T}_i\mathbf{W}_T,\ K = \mathbf{P}_i\mathbf{W}_P,\ V = \mathbf{P}_i\mathbf{W}_P\right). \tag{4}$$

We take the cross-attended EOS representation $\mathbf{h}_i^{\times} = \mathbf{H}_{i,\mathrm{EOS}}$ as a compact query-conditioned visual summary.

3.3 Ratio-Conditioned Utility Routing

Training a separate router for each retained-token ratio would cause the number of routers to scale linearly with the number of supported ratios. We instead share a single router across all ratios and explicitly condition it on r. The final router input is

$$\mathbf{z}_i = \left[\mathbf{t}_i; \mathbf{v}_i; r; \log r; \mathbf{h}_i^{\times}\right]. \tag{5}$$

A two-layer MLP maps $\mathbf{z}_i$ to a utility estimate for each candidate strategy. Predicting utility rather than correctness directly aligns the routing decision with the cost-aware objective in Equation 1, placing Full and the pruning candidates on a common scale at every retained-token ratio.

Training uses offline per-strategy correctness labels. For each available sample–ratio pair, these labels are converted into the utility target

$$\mathbf{U}_i = \left[U_i(s_1, r), \ldots, U_i(s_K, r)\right]^{\top}.$$

For stable optimization, we standardize each candidate's utility targets using training-set statistics and regress onto the standardized targets $\widetilde{\mathbf{U}}_i$; predictions are mapped back to the original utility scale before strategy selection. The complete standardization procedure is provided in Appendix A.3.2. We optimize the router using an element-wise Huber loss:

$$\mathcal{L}(\theta) = \frac{1}{NK}\sum_{i=1}^{N}\sum_{k=1}^{K}\ell_{\mathrm{Huber}}\left(\hat{\tilde{u}}_{i,k}, \widetilde{U}_{i,k}\right). \tag{6}$$

Only the feature projections, cross-attention module, and utility predictor are updated; the preview encoders and MLLM remain frozen. At inference time, VIP-Router predicts candidate utilities at the specified retained-token ratio, selects the highest-utility candidate, and runs the unchanged MLLM once with the selected strategy.

4 Experiments

4.1 Experimental Setup

Models, strategies, and data. Following VTC-Bench (Liao et al., 2026), we use Qwen2-VL-7B (Wang et al., 2024) as the backbone and adopt the four pruning implementations released by VTC-Bench—FastV (Chen et al., 2025), VisionZip (Yang et al., 2025), PruMerge+ (Shang et al., 2025), and DART (Wen et al., 2025)—without modifying their code or hyperparameters. The VTC-Bench Group A set is drawn from eight benchmarks: GQA (Hudson & Manning, 2019), MMBench and MMBench-CN (Liu et al., 2025), MME (Fu et al., 2024), POPE (Li et al., 2023), MMStar (Chen et al., 2024), OCRBench (Liu et al., 2024b), and ChartQA (Masry et al., 2022), and is constructed independently at each of five pruning levels, with retained-token ratios $r \in \{0.25, 0.1111, 0.0625, 0.04, 0.01\}$, i.e., 75%–99% reduction. The per-strategy correctness labels released with VTC-Bench constitute our routing dataset.

Table 1: **Cost-aware comparison across reduction ratios.** On the VTC-Bench Group A test set under the primary record-disjoint protocol. VIP-Router results are averaged over five random seeds; Best Fixed (VisionZip) and the Per-Sample Oracle are deterministic. Best Fixed retains a fraction r of the visual tokens, whereas VIP-Router's realized Token Cost can exceed r because it may select Full. Accuracy and utility are multiplied by 100 for reporting; token cost remains on the $[0, 1]$ scale.

Reduction Ratio	Oracle			Best Fixed *VisionZip*			VIP-Router			OHR
	Acc.	Cost	Utility	Acc.	Cost	Utility	Acc.	Cost	Utility	
75.00%	100.00	0.351	64.91	57.54	0.250	32.54	66.91	0.287	**38.23**	17.58%
88.89%	100.00	0.353	64.70	47.94	0.111	36.83	57.65	0.144	**43.24**	23.00%
93.75%	100.00	0.390	61.02	37.42	0.063	31.17	49.62	0.095	**40.08**	29.85%
96.00%	100.00	0.460	53.96	33.05	0.040	29.05	44.29	0.073	**36.95**	31.71%
99.00%	100.00	0.586	41.40	25.82	0.010	24.82	37.48	0.076	**29.89**	30.58%

Evaluation protocols. Our primary protocol splits the data at the level of image–question records, keeping all ratio-specific examples of a record in one partition. The record is the natural unit here, since a new question about an already-seen image carries its own answer, its own task-relevant visual evidence, and hence potentially a different optimal pruning strategy. However, because several constituent benchmarks pair multiple questions with the same image, visual content can recur across splits: our 12,919 records correspond to 4,372 unique images, matched by decoded pixel content. Since VIP-Router conditions directly on visual preview features, we also evaluate a stricter image-disjoint protocol, which assigns all records sharing an image—across questions, ratios, and benchmarks—to the same partition; we construct three such splits independently and train the router with five random seeds on each, yielding the 15 runs reported in Section 4.2.

Baselines and metrics. Besides the four pruning strategies applied uniformly to all test samples, we compare against *Best Fixed*, our primary baseline, which selects the single strategy that maximizes average utility over all training samples and ratios and applies it to every test sample at every ratio. Best Fixed instantiates the one-size-fits-all setting studied here and mirrors common practice, in which pruning criteria are compared in aggregate across ratios and benchmarks, and a single criterion is fixed while the pruning level remains a deployment-time hyperparameter. We further report two comparisons in the appendix: *Best Fixed@Ratio*, which adapts to the pruning level but not to the sample and thus separates ratio-level from sample-level adaptation (Appendix A.5.3); and OccamToken (Li et al., 2026a), the most closely related training-free, sample-adaptive method, which adapts token allocation rather than the pruning criterion (Appendix A.5.2). The *Per-Sample Oracle* selects the highest-utility strategy using ground-truth test labels and serves as an unattainable upper bound. We report Accuracy on Group A, Utility (Equation 1) averaged over test samples, Token Cost (the realized fraction of retained visual tokens, which can exceed r when Full is selected), and Oracle Headroom Recovery, $\mathrm{OHR} = (U_{\text{VIP-Router}} - U_{\text{best-fixed}})/(U_{\text{oracle}} - U_{\text{best-fixed}}) \times 100\%$. Implementation details, including optimization, split statistics, and hardware, are given in Appendix A.3.

4.2 Main Results

Cost-aware performance. Under the primary record-disjoint protocol, VIP-Router consistently outperforms Best Fixed even after its realized visual-token cost is charged (Table 1). Averaged across the five reduction ratios, utility rises from 30.88 to 37.68, a 22.0% relative improvement, and the gain holds at every ratio, ranging from 5.07 to 8.91 utility points. It also holds against each fixed strategy individually: the strongest fixed criterion shifts with the pruning level—PruMerge+ at 93.75% reduction, DART at 96.00%, VisionZip elsewhere—yet VIP-Router attains higher utility than all four at every ratio (Appendix A.5.4), and letting the fixed criterion vary with the ratio closes almost none of this gap (Appendix A.5.3). Relative to the Per-Sample Oracle, VIP-Router recovers 17.6–31.7% of the available utility headroom, with a macro-average OHR of 26.5%. This gain is not driven by full-token inference. VIP-Router may select Full when pruning is predicted to be unfavorable, but the resulting token cost is explicitly charged in utility, and Full accounts for only 3.48–6.66% of routing decisions across the five ratios (Figure 5). A worst-case counterfactual further

Table 2: **Record-disjoint versus image-disjoint evaluation.** Macro-averaged results over the five reduction ratios, under the primary record-disjoint protocol and three independently constructed image-disjoint splits in which no image is shared across training, validation, and test sets. Best Fixed is re-selected on each training split and is VisionZip throughout, so its Token Cost is constant at $\bar{r} = 0.095$ and is omitted; Oracle accuracy is 100.00 by construction of Group A and is likewise omitted. VIP-Router entries are means over five training seeds, with macro-average Utility standard deviations of 0.64 (record-disjoint) and 0.51–0.84 (image-disjoint). Seeds ↑ counts the training runs whose macro-average Utility exceeds that of Best Fixed. OHR is macro-averaged across the five reduction ratios and therefore cannot be recomputed from the macro-averaged Utilities in the same row. Accuracy and Utility are multiplied by 100; Token Cost remains on the $[0, 1]$ scale. Per-seed and per-ratio results are provided in Appendix A.5.1.

Evaluation Split	Oracle	Best Fixed		VIP-Router			OHR	Seeds ↑
	Utility	Acc.	Utility	Acc.	Cost	Utility		
Record-disjoint (primary)	57.20	40.35	30.88	51.19	0.135	37.68	26.5%	5/5
Image-disjoint I	56.53	40.17	30.70	43.98	0.115	**32.43**	7.4%	5/5
Image-disjoint II	55.40	39.01	29.54	42.87	0.122	**30.62**	4.5%	5/5
Image-disjoint III	55.87	40.56	31.09	43.27	0.107	**32.60**	6.7%	5/5
Image-disjoint (mean)	55.93	39.91	30.44	43.37	0.115	**31.88**	6.2%	15/15

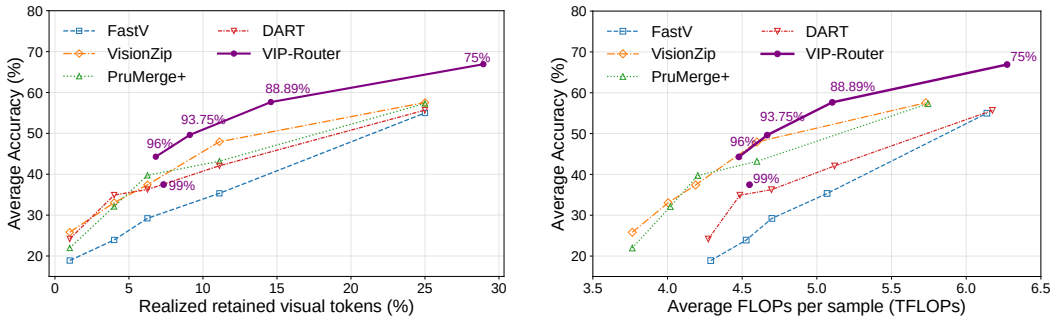
limits its contribution to at most 0.41 utility points on average, or 6.1% of the observed gain over Best Fixed (Remark 1). The gain therefore comes predominantly from sample-adaptive selection among pruning strategies.

Accuracy under the primary protocol. The same trend holds for plain accuracy (Table 1): VIP-Router exceeds Best Fixed by at least 9.37 points at every ratio and raises the five-ratio average from 40.35% to 51.19%, a 26.9% relative improvement. The relative margin widens under aggressive pruning, reaching 45.2% at 99% reduction. The gain is also spread across tasks rather than concentrated in a few: VIP-Router achieves the highest accuracy among pruning methods in 30 of the 40 benchmark–ratio combinations (75.0%), with the complete per-benchmark results in Appendix A.5.4.

Generalization to unseen images. Under the record-disjoint protocol, an image seen during training may reappear in the test set paired with a different question. We therefore repeat the comparison on three independently constructed image-disjoint splits, re-selecting Best Fixed on the training portion of each split. VIP-Router outperforms Best Fixed in both accuracy and utility on all three splits (Table 2), with relative utility gains of 3.7–5.6% (4.7% on average) and relative accuracy gains of 6.7–9.9% (8.7% on average); all 15 runs (three splits $\times$ five seeds) improve macro-average utility over Best Fixed. As expected, the margins are narrower than under the primary protocol, since the router can no longer draw on visual content seen during training. Note, however, that the two protocols use different test sets, so their magnitudes are not directly comparable. The persistence of this advantage across independently constructed splits and all training runs suggests that the signal driving sample-adaptive selection is not tied to specific images. Per-ratio results and split statistics are provided in Appendix A.5.1.

4.3 Efficiency Analysis

Accuracy–efficiency comparison. Figure 3 examines whether the accuracy gains persist once VIP-Router's realized visual-token cost and routing computation are counted. In Figure 3a, VIP-Router reaches higher accuracy than the fixed strategies at each prescribed reduction ratio despite its higher realized token cost, which exceeds the target ratio because Full is occasionally selected. Figure 3b further accounts for the FLOPs of routing and of the selected strategy; the advantage generally persists. The main exception is the 99% setting, where Full is selected most often (6.66%), so both the realized token cost and FLOPs deviate furthest from those of the fixed strategies; the efficiency margin narrows there, although accuracy still improves substantially on that subset.



(a) Accuracy versus realized token cost.

(b) Accuracy versus total inference FLOPs.

Figure 3: **Accuracy–efficiency comparison.** Point labels indicate prescribed reduction ratios. Because Group A is built separately at each ratio, points at different ratios cover different evaluation subsets and do not form a single Pareto frontier; comparisons are therefore made within each prescribed ratio. Total FLOPs include routing overhead and the selected pruning strategy.

Table 3: **Cross-backbone evaluation and benchmark transfer.** Macro-averaged over the five reduction ratios under the record-disjoint protocol; parenthesized percentages report OHR, also macro-averaged. In (a), each backbone's Best Fixed strategy follows its name in parentheses. In (b), the source-trained router and source-selected strategy are applied without target-specific training or reselection. Best Fixed Token Cost is constant at $\bar{r} = 0.095$. All entries are single runs.

Method	Acc.	Cost	Utility (OHR)
LLaVA-1.5-7B (PruMerge+)			
Oracle	100.00	0.458	54.23
Best Fixed	43.38	0.095	33.91
VIP-Router	51.82	0.128	**38.97** (26.15%)
LLaVA-OneVision-7B (DART)			
Oracle	100.00	0.431	56.95
Best Fixed	45.22	0.095	35.74
VIP-Router	48.24	0.117	**36.56** (2.93%)
InternVL3-8B (DART)			
Oracle	100.00	0.385	61.46
Best Fixed	50.93	0.095	41.46
VIP-Router	53.81	0.118	**42.01** (4.17%)
Qwen2.5-VL-7B (DART)			
Oracle	100.00	0.450	54.96
Best Fixed	40.01	0.095	30.54
VIP-Router	48.75	0.134	**35.40** (20.52%)

(a) Evaluation across MLLM backbones.

Method	Acc.	Cost	Utility (OHR)
ScienceQA-IMG			
Oracle	100.00	0.518	48.16
Best Fixed	29.59	0.095	20.11
VIP-Router	36.18	0.106	**25.54** (19.47%)
MMMU			
Oracle	100.00	0.533	46.71
Best Fixed	31.99	0.095	22.52
VIP-Router	35.22	0.106	**24.59** (6.53%)
RealWorldQA			
Oracle	100.00	0.373	62.67
Best Fixed	38.41	0.095	28.94
VIP-Router	43.64	0.096	**34.08** (13.62%)
SEED-Bench Image			
Oracle	100.00	0.445	55.48
Best Fixed	39.27	0.095	29.80
VIP-Router	40.21	0.103	**29.87** (0.47%)

(b) Transfer to unseen benchmarks.

Router parameter overhead. The default VIP-Router introduces only 1.38M trainable parameters, i.e., 0.017% of the 8.291B Qwen2-VL backbone (Wang et al., 2024). Its external frozen preview encoders are optional: by reusing native MLLM representations, VIP-Router can eliminate additional frozen parameters entirely while keeping its trainable parameter overhead at or below 0.05% of the backbone. The complete parameter breakdown and native-feature variants are provided in Appendix A.4.

4.4 Cross-Backbone Evaluation and Benchmark Transfer

Evaluation across MLLM backbones. We repeat the pipeline on four additional MLLMs: LLaVA-1.5-7B, LLaVA-OneVision-7B, InternVL3-8B, and Qwen2.5-VL-7B (Liu et al., 2024a; Li et al., 2024a; Zhu et al., 2025; Bai et al., 2025). For each, we rerun inference with full tokens, with equivalent-ratio image downsampling, and with each pruning strategy, rebuild Group A as in Appendix A.3.1, and then train a router and select Best Fixed on the resulting training split. VIP-Router improves utility on all four backbones by 0.55 to 5.06 points (Table 3a). Gains on all four

Table 4: **Staged ablation of the VIP-Router design.** Under the record-disjoint protocol, macro-averaged over the five reduction ratios. Checkmarks indicate included components. Pooling specifies how spatial visual tokens are aggregated into the router input, and is marked with (–) when they are collapsed into V_{global}; the Mean entry in Stage C denotes that same global average. All results are single runs and comparisons are made within each stage; the best utility within each stage is bolded. Stages A–B and Stage C were trained under different random seeds, so absolute values should not be compared across stages: the last row of Stage B and the first row of Stage C specify the same configuration and differ only in training trajectory (36.96 versus 36.41).

# Routers	r Input	T_{global}	V_{global}	Spatial Info	Cross Attn.	Pooling	Acc.	Cost	Utility
A. Global feature inputs									
5	–	✓	–	–	–	–	47.99	0.122	35.81
5	–	–	✓	–	–	–	51.26	0.132	38.02
5	–	✓	✓	–	–	–	52.01	0.139	**38.07**
B. Cross-ratio parameter sharing									
5	–	✓	✓	–	–	–	52.01	0.139	**38.07**
1	✓	✓	✓	–	–	–	49.59	0.126	36.96
C. Query-conditioned spatial interaction									
1	✓	✓	✓	–	–	Mean	48.29	0.119	36.41
1	✓	–	✓	✓	–	Learnable	50.82	0.136	37.23
1	✓	✓	✓	✓	–	Learnable	49.81	0.124	37.45
1	✓	✓	✓	✓	✓	Masked mean	50.39	0.125	37.94
1	✓	✓	✓	✓	✓	EOS	51.51	0.134	**38.09**

backbones indicate that the formulation applies broadly, whereas their varying magnitudes suggest that how well strategy preferences can be predicted from preview features depends on the backbone.

Zero-shot transfer to unseen benchmarks. We further evaluate on ScienceQA-IMG, MMMU, RealWorldQA, and SEED-Bench Image (Lu et al., 2022; Yue et al., 2024; xAI, 2024; Li et al., 2024b), rebuilding each target Group A by the same protocol and transferring the source-trained router and source-selected Best Fixed strategy without target-specific training or reselection. Utility gains are positive on all four (Table 3b), largest on ScienceQA-IMG and RealWorldQA and marginal on SEED-Bench Image; decoded-image matching confirms that no target image occurs in the VTC-Bench source pool. These results suggest that learned pruning-strategy preferences can transfer to unseen tasks and visual content, although the degree of transfer is sensitive to domain shift.

4.5 Ablation Studies

Table 4 reports a staged ablation. Global visual features are the stronger single source of routing signal, and adding global text features on top of them yields only a marginal gain ($38.02 \rightarrow 38.07$), suggesting that coarse text summaries add little when simply concatenated (Stage A). Replacing five ratio-specific routers with a single ratio-conditioned router costs 1.11 utility points but consolidates all pruning levels into one model (Stage B). Within Stage C, richer inputs to the shared router help consistently: retaining spatial visual tokens rather than collapsing them to a global average, and letting the question attend to them via text-to-vision cross-attention with the cross-attended EOS position as the pooled summary, together raise utility from 36.41 to 38.09. This 1.68-point within-stage gain more than offsets the 1.11-point cost of cross-ratio sharing in Stage B. Because the stages use different seeds, we compare only matched within-stage differences and interpret the ablation as indicating the relative ordering of design choices rather than resolving differences of a few tenths of a point.

5 Conclusion

We introduce VIP-Router, which selects a pruning strategy per sample rather than applying one criterion to all inputs. Its gains hold across evaluation protocols, backbones, and unseen benchmarks, though they are smaller on unseen images, indicating that pruning-strategy suitability is partially predictable at the sample level.

AI USE STATEMENT

In this work, we used generative AI tools to assist with software implementation and code review, data processing and table preparation, refinement of research ideas and experimental methodology, drafting of mathematical proofs, and editing the manuscript for clarity and readability. Generative AI was not used to generate datasets; other disclosure categories do not apply to this work.

All AI-assisted code was manually reviewed and tested by the authors. AI-assisted data processing, tables, methodological suggestions, and manuscript edits were likewise checked against the underlying implementations, experimental results, and source materials. The authors made all final research and writing decisions and take full responsibility for the content of this work, including all text, claims, code, results, and artifacts produced with the assistance of generative AI.

REPRODUCIBILITY STATEMENT

We provide the information required to reproduce VIP-Router and all reported experiments in the main paper, appendix, and supplementary materials. The routing objective and model architecture are specified in Section 3, with pseudocode provided in Appendix A.2. Dataset construction, the primary record-disjoint protocol, the additional image-disjoint evaluation protocols, baseline selection, and evaluation metrics are described in Section 4.1 and Appendix A.3. Appendix A.3 further specifies the definition of image identity, split construction, training configuration, training configuration, random seeds, software environment, hardware, and implementation details. Detailed image-disjoint split statistics and multi-split results are provided in Appendix A.5.1. Additional encoder configurations and detailed experimental results are provided in Appendices A.4 and A.5.

LIMITATIONS

Our work has three main limitations. (i) The routing-performance evaluation is primarily conducted on the pruning-sensitive VTC-Bench Group A setting. Although our analysis on the full VTC-Bench sample set shows that complementarity among pruning strategies persists without Group A filtering, the effectiveness of VIP-Router on unfiltered benchmarks containing both pruning-sensitive and pruning-insensitive samples remains to be comprehensively evaluated.

(ii) The routing advantage is substantially smaller under image-disjoint evaluation than under the primary record-disjoint protocol. Although VIP-Router retains positive gains across three independently constructed image-disjoint splits and all corresponding training runs, the reduced effect size indicates that generalization to entirely unseen visual content remains more challenging and leaves substantial room for improvement. Because the two protocols induce different training and test populations, their performance difference should not be interpreted as a direct estimate of the effect of repeated images.

(iii) Routing preferences are also partly backbone-specific. In preliminary direct-transfer experiments, a router trained on a source MLLM and deployed on unseen target architectures without target-specific adaptation generally fails to preserve the gains obtained by backbone-specific training and can fall behind fixed-strategy baselines. Thus, while the routing formulation is broadly applicable across MLLM families, robust zero-shot transfer of a learned routing policy across backbone architectures remains an open challenge.

REFERENCES

Saeed Ranjbar Alvar, Gursimran Singh, Mohammad Akbari, and Yong Zhang. DivPrune: Diversity-based Visual Token Pruning for Large Multimodal Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 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, February 2025.

Mu Cai, Jianwei Yang, Jianfeng Gao, and Yong Jae Lee. Matryoshka Multimodal Models. In *The Thirteenth International Conference on Learning Representations*, October 2024.

Bingyi Cao, Koert Chen, Kevis-Kokitsi Maninis, Kaifeng Chen, Arjun Karpur, Ye Xia, Sahil Dua, Tanmaya Dabral, Guangxing Han, Bohyung Han, Joshua Ainslie, Alex Bewley, Mithun Jacob, René Wagner, Washington Ramos, Krzysztof Choromanski, Mojtaba Seyedhosseini, Howard Zhou, and Andre Araujo. TIPSv2: Advancing Vision-Language Pretraining with Enhanced Patch-Text Alignment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 29325–29335, 2026.

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 Aleš Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol (eds.), *Computer Vision – ECCV 2024*, pp. 19–35, Cham, 2025. Springer Nature Switzerland. ISBN 978-3-031-73004-7. doi: 10.1007/978-3-031-73004-7_2.

Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Jiaqi Wang, Yu Qiao, Dahua Lin, and Feng Zhao. Are We on the Right Way for Evaluating Large Vision-Language Models? In *Advances in Neural Information Processing Systems*, volume 37, pp. 27056–27087. Curran Associates, Inc., 2024. doi: 10.52202/079017-0850.

Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, Yunsheng Wu, and Rongrong Ji. MME: A Comprehensive Evaluation Benchmark for Multimodal Large Language Models, March 2024. Comment: Project page: https://github.com/BradyFU/Awesome-Multimodal-Large-Language-Models.

Wenbo Hu, Zi-Yi Dou, Liunian Harold Li, Amita Kamath, Nanyun Peng, and Kai-Wei Chang. Matryoshka Query Transformer for Large Vision-Language Models. In *Advances in Neural Information Processing Systems*, volume 37, pp. 50168–50188. Curran Associates, Inc., 2024. doi: 10.52202/079017-1588.

Wenxuan Huang, Zijie Zhai, Yunhang Shen, Shaosheng Cao, Fei Zhao, Xiangfeng Xu, Zheyu Ye, and Shaohui Lin. Dynamic-LLaVA: Efficient Multimodal Large Language Models via Dynamic Vision-language Context Sparsification. *International Conference on Learning Representations*, 2025:69927–69955, May 2025.

YiJie Huang, Yiqun Zhang, Zhuoyue Jia, Xiaocui Yang, Junzhao Huang, Zihan Wang, Shi Feng, Daling Wang, Yifei Zhang, and Yongkang Liu. How Many Visual Tokens Do Multimodal Language Models Need? Scaling Visual Token Pruning with Fˆ3A, May 2026.

Drew A. Hudson and Christopher D. Manning. GQA: A New Dataset for Real-World Visual Reasoning and Compositional Question Answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6700–6709, 2019.

Yizhang Jin, Jian Li, Tianjun Gu, Yexin Liu, Bo Zhao, Jinxiang Lai, Zhenye Gan, Yabiao Wang, Chengjie Wang, Xin Tan, and Lizhuang Ma. Efficient multimodal large language models: A survey. *Visual Intelligence*, 3(1):27, December 2025. ISSN 2731-9008. doi: 10.1007/s44267-025-00099-6.

Jiwan Kim, Kibum Kim, Wonjoong Kim, Byung-Kwan Lee, and Chanyoung Park. Why and When Visual Token Pruning Fails? A Study on Relevant Visual Information Shift in MLLMs Decoding, April 2026. Comment: Preprint, Project : https://ptkjw1997.github.io/DSTP-page/.

Moshe Leshno, Vladimir Ya. Lin, Allan Pinkus, and Shimon Schocken. Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural Networks*, 6(6):861–867, January 1993. ISSN 0893-6080. doi: 10.1016/S0893-6080(05)80131-5.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. LLaVA-OneVision: Easy Visual Task Transfer. *Transactions on Machine Learning Research*, October 2024a. ISSN 2835-8856.

Bohao Li, Yuying Ge, Yixiao Ge, Guangzhi Wang, Rui Wang, Ruimao Zhang, and Ying Shan. SEED-Bench: Benchmarking Multimodal Large Language Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13299–13308, 2024b.

Geng Li, Guohao Chen, Ting Chen, Shilin Shan, Kuangji Zuo, Bofan Lyu, Tuo An, Gen Li, and Jianfei Yang. OccamToken: Efficient VLM Inference with Training-Free and Budget-Adaptive Token Pruning, May 2026a. Comment: 26 pages,8 figures.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Xin Zhao, and Ji-Rong Wen. Evaluating Object Hallucination in Large Vision-Language Models. In *The 2023 Conference on Empirical Methods in Natural Language Processing*, December 2023.

Zhongyang Li, Yaqian Li, Faming Fang, Rinyoichi Takezoe, Zi-Hao Bo, Cheng Qian, Mo Guang, Guixu Zhang, and Kaiwen Long. QMoP: Query Guided Mixture-of-Projector for Efficient Visual Token Compression, March 2026b.

Chenfei Liao, Wensong Wang, Zichen Wen, Xu Zheng, Yiyu Wang, Haocong He, Yuanhuiyi Lyu, Lutao Jiang, Xin Zou, Yuqian Fu, Bin Ren, Linfeng Zhang, and Xuming Hu. Are We Using the Right Benchmark: An Evaluation Framework for Visual Token Compression Methods. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 4236–4253, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.195.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved Baselines with Visual Instruction Tuning. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 26286–26296, June 2024a. doi: 10.1109/CVPR52733.2024.02484.

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. MMBench: Is Your Multi-modal Model an All-Around Player? In Aleš Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol (eds.), *Computer Vision – ECCV 2024*, pp. 216–233, Cham, 2025. Springer Nature Switzerland. ISBN 978-3-031-72658-3. doi: 10.1007/978-3-031-72658-3_13.

Yuliang Liu, Zhang Li, Mingxin Huang, Biao Yang, Wenwen Yu, Chunyuan Li, Xu-Cheng Yin, Cheng-Lin Liu, Lianwen Jin, and Xiang Bai. OCRBench: On the hidden mystery of OCR in large multimodal models. *Science China Information Sciences*, 67(12):220102, December 2024b. ISSN 1869-1919. doi: 10.1007/s11432-024-4235-6.

Pan Lu, Swaroop Mishra, Tony 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. In *Advances in Neural Information Processing Systems*, October 2022.

Ahmed Masry, Do Xuan Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. ChartQA: A Benchmark for Question Answering about Charts with Visual and Logical Reasoning. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Findings of the Association for Computational Linguistics: ACL 2022*, pp. 2263–2279, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-acl.177.

Isaac Ong, Amjad Almahairi, Vincent Wu, Wei-Lin Chiang, Tianhao Wu, Joseph E. Gonzalez, M. Waleed Kadous, and Ion Stoica. RouteLLM: Learning to Route LLMs from Preference Data. In *The Thirteenth International Conference on Learning Representations*, October 2024.

Alec Radford, Jong Wook Kim, Chris Hallacy, A. Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and I. Sutskever. Learning Transferable Visual Models From Natural Language Supervision. In *International Conference on Machine Learning*, February 2021. [TLDR] It is demonstrated that the simple pre-training task of predicting which caption goes with which image is an efficient and scalable way to learn SOTA image representations from scratch on a dataset of 400 million (image, text) pairs collected from the internet.

Yuzhang Shang, Mu Cai, Bingxin Xu, Yong Jae Lee, and Yan Yan. LLaVA-PruMerge: Adaptive Token Reduction for Efficient Large Multimodal Models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 22857–22867, 2025.

Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, Olivier Hénaff, Jeremiah Harmsen, Andreas Steiner, and Xiaohua Zhai. SigLIP 2: Multilingual Vision-Language Encoders with Improved Semantic Understanding, Localization, and Dense Features, February 2025. Comment: Model checkpoints are available at https://github.com/google-research/big_vision/tree/main/big_vision/configs/proj/image_text/README_siglip2.md.

Michael Wang, Keith Li, and Roozbeh Bostandoost. MetaKV: Adaptive KV Cache Compression for Constrained LLM Inference. https://arxiv.org/abs/2609.07966v1, September 2026.

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. Qwen2-VL: Enhancing Vision-Language Model's Perception of the World at Any Resolution, October 2024. Comment: Code is available at https://github.com/QwenLM/Qwen2-VL. arXiv admin note: text overlap with arXiv:2408.15262 by other authors.

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. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 9961–9980, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.505.

xAI. Grok-1.5 vision preview. https://x.ai/blog/grok-1.5v, 2024.

Zhuoyan Xu, Khoi Duc Nguyen, Preeti Mukherjee, Saurabh Bagchi, Somali Chaterji, Yingyu Liang, and Yin Li. Learning to Inference Adaptively for Multimodal Large Language Models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3552–3563, 2025.

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. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19792–19802, 2025.

Weihao Ye, Qiong Wu, Wenhao Lin, and Yiyi Zhou. Fit and Prune: Fast and Training-free Visual Token Pruning for Multi-modal Large Language Models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(21):22128–22136, April 2025a. ISSN 2374-3468. doi: 10.1609/aaai.v39i21.34366.

Xubing Ye, Yukang Gan, Yixiao Ge, Xiao-Ping Zhang, and Yansong Tang. ATP-LLaVA: Adaptive Token Pruning for Large Vision Language Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24972–24982, 2025b.

Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, Cong Wei, Botao Yu, Ruibin Yuan, Renliang Sun, Ming Yin, Boyuan Zheng, Zhenzhu Yang, Yibo Liu, Wenhao Huang, Huan Sun, Yu Su, and Wenhu Chen. MMMU: A Massive Multi-discipline Multimodal Understanding and Reasoning Benchmark for Expert AGI. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9556–9567, 2024.

Yuan Zhang, Chun-Kai Fan, Junpeng Ma, Wenzhao Zheng, Tao Huang, Kuan Cheng, Denis A. Gudovskiy, Tomoyuki Okuno, Yohei Nakata, Kurt Keutzer, and Shanghang Zhang. SparseVLM: Visual Token Sparsification for Efficient Vision-Language Model Inference. In *Proceedings of the 42nd International Conference on Machine Learning*, pp. 74840–74857. PMLR, October 2025.

Yiwu Zhong, Zhuoming Liu, Yin Li, and Liwei Wang. AIM: Adaptive Inference of Multi-Modal LLMs via Token Merging and Pruning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 20180–20192, 2025.

Pengfei Zhou, Zhiwei Tang, Yixing Ma, Jiasheng Tang, Yizeng Han, Zhenglin Wan, Fanqing Meng, Wei Wang, Bohan Zhuang, Wangbo Zhao, et al. Agent-as-a-router: Agentic model routing for coding tasks. *arXiv preprint arXiv:2606.22902*, 2026.

Chenchen Zhu, Saksham Suri, Cijo Jose, Maxime Oquab, Marc Szafraniec, Wei Wen, Yunyang Xiong, Patrick Labatut, Piotr Bojanowski, Raghuraman Krishnamoorthi, and Vikas Chandra. Efficient Universal Perception Encoder, March 2026. Comment: Code: https://github.com/facebookresearch/EUPE; Model: https://huggingface.co/collections/facebook/eupe.

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, Hao Li, Jiahao Wang, Nianchen Deng, Songze Li, Yinan He, Tan Jiang, Jiapeng Luo, Yi Wang, Conghui He, Botian Shi, Xingcheng Zhang, Wenqi Shao, Junjun He, Yingtong Xiong, Wenwen Qu, Peng Sun, Penglong Jiao, Han Lv, Lijun Wu, Kaipeng Zhang, Huipeng Deng, Jiaye Ge, Kai Chen, Limin Wang, Min Dou, Lewei Lu, Xizhou Zhu, Tong Lu, Dahua Lin, Yu Qiao, Jifeng Dai, and Wenhai Wang. InternVL3: Exploring Advanced Training and Test-Time Recipes for Open-Source Multimodal Models, April 2025. Comment: Technical Report.

A Appendix

A.1 Theoretical Motivation: Utility Prediction from a Compact Representation

Setup. Treat a benchmark sample $x = (I, q)$ together with a retained-token ratio r as a random draw from the data distribution over (x, r); we drop the sample index i used in Section 3. The retained-token ratios lie on the finite grid

$$\mathcal{R} = \{0.25,\ 0.1111,\ 0.0625,\ 0.04,\ 0.01\}.$$

We write the candidate set as $\mathcal{S} = \{\text{Full}\} \cup \mathcal{S}_{\text{p}}$, where $\mathcal{S}_{\text{p}}$ contains the pruning strategies.

Recall the token cost $C(s, r)$ and per-sample utility

$$U(s, r) = A(s, r) - C(s, r), \qquad A(s, r) \in \{0, 1\}.$$

More generally, this objective belongs to the family $A(s, r) - \lambda C(s, r)$. We use $\lambda = 1$ throughout the paper because, under Group A, it induces the desired ordering between successful pruning, full-token inference, and failed pruning, as detailed below.

Structure of the oracle. Under Group A, full-token inference is correct by construction, so

$$U(\text{Full}, r) = 0.$$

All pruning strategies have the same token cost r: a correct pruning strategy has utility $1 - r$, whereas an incorrect one has utility $-r$. The oracle therefore has a simple structure: (i) if at least one pruning strategy is correct, every correct pruning strategy is optimal with utility $1 - r$; (ii) if all pruning strategies fail, Full is uniquely optimal with utility 0. Thus, the per-sample optimum may contain multiple equivalent pruning strategies, and the router only needs to select one zero-regret member of this set.

What the router must learn. Let $\mathbf{z}(x, r)$ denote the preview representation in Equation 5. For a pruning strategy $s \in \mathcal{S}_{\text{p}}$, the Bayes-optimal squared-loss utility predictor is

$$\bar{u}_s(\mathbf{z}) = \mathbb{E}[U(s, r) \mid \mathbf{z}] = \bar{p}_s(\mathbf{z}) - r, \qquad \bar{p}_s(\mathbf{z}) = \Pr(A(s, r) = 1 \mid \mathbf{z}). \tag{7}$$

For Full, Group A gives $\bar{u}_{\text{Full}} = 0$ identically. Utility prediction therefore reduces, in principle, to estimating the conditional success probabilities of the pruning strategies and comparing them against a known cost threshold.

At the Bayes optimum,

$$s^\star_{\mathbf{z}} = \text{Full} \quad \Longleftrightarrow \quad \max_{s \in \mathcal{S}_{\text{p}}} \bar{p}_s(\mathbf{z}) < r. \tag{8}$$

Thus, full-token inference is preferred only when every pruning strategy is predicted to preserve correctness with probability below the retained-token ratio. Our implementation predicts all candidate utilities using a common output head for simplicity, but the Full target remains constant under Group A.

Error decomposition. Let $\mathbf{U} = [U(s_1, r), \ldots, U(s_K, r)]$ denote the random utility vector and $\bar{\mathbf{u}}(\mathbf{z}) = \mathbb{E}[\mathbf{U} \mid \mathbf{z}]$. For any square-integrable predictor g, the orthogonality of the L^2 projection onto $\sigma(\mathbf{z})$ yields

$$\mathbb{E}\big\|g(\mathbf{z}) - \mathbf{U}\big\|_2^2 = \underbrace{\mathbb{E}\big\|\bar{\mathbf{u}}(\mathbf{z}) - \mathbf{U}\big\|_2^2}_{\delta^2_{\text{rep}}\ \text{(representation gap)}} + \underbrace{\mathbb{E}\big\|g(\mathbf{z}) - \bar{\mathbf{u}}(\mathbf{z})\big\|_2^2}_{\delta^2_{\text{est}}\ \text{(estimation error)}}. \tag{9}$$

The representation gap is the utility uncertainty that survives compressing (x, r) into $\mathbf{z}$; it cannot be reduced by enlarging the predictor, only by enriching $\mathbf{z}$. Our three-branch design (global text, global vision, and query-conditioned spatial evidence) is designed to reduce this representation gap, and we validate it empirically by ablation. [1]

[1] We train with Huber loss (δ=1, Equation 6) for robustness. Each target $U(s, r)$ takes values in an interval of length one ($[-r,\ 1 - r]$ for $s \in \mathcal{S}_{\text{p}}$; $[-1, 0]$ for Full). For the unstandardized utility targets, predictions inside the interval incur residuals in the quadratic regime, while predictions outside it are strictly dominated; the population minimizer therefore coincides with the conditional mean. We use this observation only to motivate utility regression, since our actual training objective applies Huber loss to standardized utility targets, for which exact equivalence to the conditional mean is not assumed.

Approximation by a lightweight MLP. Since $\mathcal{R}$ is finite and the frozen encoders produce bounded features, $\mathbf{z}$ ranges over a compact set $\mathcal{Z}$. Assuming each $\bar{u}_s$ is continuous on $\mathcal{Z}$, universal approximation for non-polynomial activations (Leshno et al., 1993) implies that for any $\delta_{\text{mlp}} > 0$ there exists a finite-width two-layer GELU network g_θ such that

$$\sup_{\mathbf{z}\in\mathcal{Z}} \left\| g_\theta(\mathbf{z}) - \bar{\mathbf{u}}(\mathbf{z}) \right\|_\infty \le \delta_{\text{mlp}}. \tag{10}$$

This is an existence statement; optimization and generalization residuals of the trained router are absorbed into δ_{est}.

From utility prediction to decision quality. Let $\hat{\mathbf{u}} = g_\theta(\mathbf{z})$ and $\hat{s} \in \arg\max_{s\in\mathcal{S}} \hat{u}_s$.

Lemma 1 (Pointwise regret)**.** *If $\|\hat{\mathbf{u}} - \mathbf{U}\|_\infty \le \varepsilon$ on a sample (x, r), then $U(s^\star) - U(\hat{s}) \le 2\varepsilon$.*

Proof. $U(s^\star) \le \hat{u}_{s^\star} + \varepsilon \le \hat{u}_{\hat{s}} + \varepsilon \le U(\hat{s}) + 2\varepsilon$. □

Corollary 1 (Decision margin at aggressive pruning levels)**.** *Let $\mathcal{S}^\star(x, r)$ denote the set of optimal strategies and define the nonzero decision margin as the gap between the optimal utility and the best strictly suboptimal utility. Under Group A,*

$$\Delta(x, r) \in \{r,\ 1 - r\}.$$

If at least one pruning strategy is correct, the gap between a correct pruning strategy and Full *is $1 - r$. If all pruning strategies fail, the gap between* Full *and a failed pruning strategy is r.*

Consequently, a uniform utility-prediction error $\varepsilon < \Delta(x, r)/2$ is sufficient to preserve an optimal decision. For $r \in \{0.04, 0.0625, 0.1111, 0.25\}$, $\varepsilon < 0.02$ is sufficient for all samples, whereas at $r = 0.01$ the corresponding guarantee requires $\varepsilon < 0.005$. The shrinking margin at the most aggressive pruning level occurs only at the boundary between Full *and failed pruning strategies; crossing this boundary incurs utility regret r.*

Proposition 1 (Expected regret)**.**

$$\mathbb{E}\big[\, U(s^\star) - U(\hat{s}) \,\big] \;\le\; 2\,\mathbb{E}\big\|\hat{\mathbf{u}} - \mathbf{U}\big\|_\infty \;\le\; 2\sqrt{\delta_{\text{rep}}^2 + \delta_{\text{est}}^2}, \tag{11}$$

where the first inequality applies the proof of Lemma 1 pointwise, and the second uses $\|\cdot\|_\infty \le \|\cdot\|_2$ and Jensen's inequality.

Scope of the analysis. The analysis formalizes a limited claim: sample-adaptive pruning-strategy selection can be reduced to estimating conditional pruning success accurately enough to preserve its ordering relative to known token costs. It does not establish that the preview representation is sufficient a priori. This is an empirical question, which we evaluate through the feature ablations and the observed utility headroom recovered relative to the Per-Sample Oracle.

Remark 1 (Upper bound on the contribution of full-token fallback)**.** *Let f_r denote the fraction of samples for which VIP-Router selects* Full *at retained-token ratio r. Consider a counterfactual policy that disables* Full *and instead routes these samples to a pruning strategy. In the worst case, all such replacements are incorrect. Accuracy can therefore decrease by at most f_r, while Token Cost decreases by exactly $f_r(1 - r)$. Consequently,*

$$U_{\text{forced-pruning}} \ge U_{\text{VIP-Router}} - f_r r. \tag{12}$$

Using the five-seed mean routing frequencies, the corresponding bounds are 1.23, 0.41, 0.22, 0.14, *and* 0.07 *utility points at the five reduction ratios, respectively. The average bound is only* 0.41 *utility points, corresponding to at most* 6.1% *of VIP-Router's observed utility gain over Best Fixed. Thus, even under this worst-case counterfactual, access to full-token inference can explain only a small fraction of the overall improvement.*

A.2 ALGORITHM

Algorithm 1 Training procedure of VIP-Router.

Require: Training set $\mathcal{D}_{\text{train}} = \{(I_i, q_i, r_i, \mathbf{a}_i)\}_{i=1}^{N}$, candidate set $\mathcal{S} = \{s_1, \ldots, s_K\}$, frozen preview encoders, and router g_θ

Ensure: Trained router parameters $\theta^\star$ and training-set normalization statistics $\{(\mu_k, \sigma_k)\}_{k=1}^{K}$

Construct utility targets for all training samples:

$$U_i(s_k, r_i) \leftarrow a_i^{s_k} - c(s_k, r_i), \qquad i = 1, \ldots, N, \;\; k = 1, \ldots, K$$

Compute per-strategy normalization statistics on the training set:

$$\mu_k \leftarrow \frac{1}{N}\sum_{i=1}^{N} U_i(s_k, r_i), \qquad \sigma_k \leftarrow \sqrt{\frac{1}{N}\sum_{i=1}^{N}\big(U_i(s_k, r_i) - \mu_k\big)^2}, \quad k = 1, \ldots, K$$

for each training epoch **do**
  for each minibatch $\mathcal{B} \subset \mathcal{D}_{\text{train}}$ **do**
    for each $(I_i, q_i, r_i, \mathbf{a}_i) \in \mathcal{B}$ **do**
      Standardize the utility target for each strategy:

$$\widetilde{U}_{i,k} \leftarrow \frac{U_i(s_k, r_i) - \mu_k}{\sigma_k}, \qquad k = 1, \ldots, K$$

      Form the standardized target utility vector:

$$\widetilde{\mathbf{U}}_i \leftarrow [\widetilde{U}_{i,1}, \ldots, \widetilde{U}_{i,K}]^\top$$

      Extract frozen visual and textual preview features from (I_i, q_i)
      Compute query-conditioned cross-modal features
      Predict standardized strategy utilities:

$$\hat{\widetilde{\mathbf{u}}}_i \leftarrow g_\theta(I_i, q_i, r_i)$$

    end for
    Compute the element-wise Huber loss:

$$\mathcal{L} \leftarrow \frac{1}{|\mathcal{B}|K}\sum_{i\in\mathcal{B}}\sum_{k=1}^{K} \ell_{\text{Huber}}\big(\hat{\widetilde{u}}_{i,k}, \widetilde{U}_{i,k}\big)$$

    Update θ by minimizing $\mathcal{L}$
  end for
  Evaluate utility on the validation set after recovering utilities to the original scale
end for
return checkpoint $\theta^\star$ with the highest validation utility and $\{(\mu_k, \sigma_k)\}_{k=1}^{K}$

Algorithm 2 Inference procedure of VIP-Router.

Require: Image I, question q, retained-token ratio r, candidate set $\mathcal{S} = \{s_1, \ldots, s_K\}$, trained router $g_{\theta^\star}$, training-set normalization statistics $\{(\mu_k, \sigma_k)\}_{k=1}^{K}$, and frozen MLLM f

1: Extract frozen visual and textual preview features from (I, q)

2: Compute query-conditioned cross-modal features

3: Predict standardized strategy utilities:

$$\hat{\tilde{\mathbf{u}}} \leftarrow g_{\theta^\star}(I, q, r)$$

4: Recover predicted utilities on the original scale:

$$\hat{u}_k \leftarrow \sigma_k \hat{\tilde{u}}_k + \mu_k, \qquad k = 1, \ldots, K$$

5: Select the highest-utility candidate:

$$\hat{s} \leftarrow s_{\arg\max_{k \in \{1,\ldots,K\}} \hat{u}_k}$$

6: Apply $\hat{s}$ and run the frozen MLLM once:

$$\hat{y} \leftarrow f_{\hat{s}}(I, q; r)$$

7: **return** $\hat{y}$

A.3 Implementation and Reproducibility Details

A.3.1 Dataset Construction and Split Details

We construct the routing dataset from the per-sample inference results officially released by VTC-Bench (Liao et al., 2026) for Qwen2-VL-7B-Instruct. The released results include full-token inference, equivalent-ratio image downsampling, and the four candidate pruning methods evaluated at multiple reduction ratios. For each benchmark, binary correctness is derived from its task-specific evaluation fields. For MMBench and MMBench-CN, we compare the extracted multiple-choice prediction with the ground-truth answer. For MMBench-CN, we retain the original Chinese question and answer options without translation or language-specific preprocessing. The resulting Chinese–English mixed prompt is used directly for routing feature extraction and is processed by the same frozen CLIP text encoder as all other benchmarks.

Following VTC-Bench, we reconstruct Group A independently at each reduction ratio. An image–question record is included at a given ratio when full-token inference is correct while equivalent-ratio image downsampling is incorrect. We then associate the correctness labels of FastV, VisionZip, PruMerge$^+$, and DART with each eligible record–ratio pair. This procedure produces 33,091 routing examples from 12,919 unique image–question records. Because Group A membership depends on the reduction ratio, the routing data form a sparse record–ratio grid: the same record may appear at only a subset of the evaluated ratios.

We evaluate five retained-token ratios,

$$r \in \{0.25,\ 0.1111,\ 0.0625,\ 0.04,\ 0.01\},$$

corresponding to visual-token reduction ratios of 75.00%, 88.89%, 93.75%, 96.00%, and 99.00%, respectively. The ratio inputs to VIP-Router are r and $\ln(r + 10^{-8})$.

Primary record-disjoint protocol. The primary evaluation partitions data at the image–question-record level. Records are identified by

$$\texttt{sample_key} = \texttt{benchmark} : \texttt{doc_id}$$

and assigned deterministically to training, validation, and test sets using a fixed hash-based $70/15/15$ split. All ratio-specific examples derived from the same record inherit the same partition. This produces 8,988 training, 1,956 validation, and 1,975 test records, with no image–question record shared across splits.

This protocol is record-disjoint but not necessarily image-disjoint: different questions associated with the same underlying image can have different document identifiers and therefore appear in

Table 5: Number of routing examples at each visual-token reduction ratio under the primary record-disjoint protocol. Group A membership is ratio-dependent, so the number of eligible examples varies across ratios.

Reduction	Retained r	Train	Val.	Test	Total
75.00%	0.2500	1,998	411	431	2,840
88.89%	0.1111	3,720	810	801	5,331
93.75%	0.0625	4,772	1,044	1,037	6,853
96.00%	0.0400	5,391	1,136	1,183	7,710
99.00%	0.0100	7,218	1,551	1,588	10,357
Total	–	23,099	4,952	5,040	33,091

different partitions. Across the 12,919 image–question records, we identify 4,372 unique decoded images.

Image-disjoint protocols. To evaluate generalization to completely unseen images, we additionally construct three image-disjoint splits. Image identity is defined from the decoded RGB image rather than from file paths or document identifiers. Specifically, we compute a SHA-256 hash from the decoded RGB image dimensions and pixel values and use this hash as the global image identifier. All records sharing the same decoded image—including different questions, ratio-specific examples, and exact duplicates appearing across benchmarks— are grouped together and assigned to a single partition. Consequently, the training, validation, and test image sets are pairwise disjoint.

The three reported image-disjoint splits use grouping seeds 826, 360, and 266. These grouping seeds are selected solely according to how closely the resulting benchmark–ratio composition matches the target $70/15/15$ partition proportions. Candidate correctness labels, utility values, router predictions, validation performance, and test performance are not used for grouping-seed selection. Group A membership, candidate pruning outcomes, retained-token ratios, utility definition, and router architecture remain identical to the primary protocol; only the train/validation/test assignment changes. Detailed partition statistics, overlap audits, and per-split results are reported in Appendix A.5.1.

A.3.2 Router Architecture and Training Configuration

Preview encoder. The default VIP-Router uses the frozen `openai/clip-vit-base-patch32` checkpoint for both visual and textual preview encoding. Images follow the default CLIP preprocessing and are resized and center-cropped to 224×224. We extract the penultimate visual hidden representation, discard the CLS token, and retain the resulting 49 patch tokens $\mathbf{P} \in \mathbb{R}^{49\times768}$. Their mean forms the 768-dimensional global visual representation.

Questions are tokenized to a maximum length of 77 tokens. We retain the 512-dimensional text-token representations and use the CLIP-projected EOS representation as the 512-dimensional global text feature. All CLIP parameters remain frozen during router training.

Cross-modal representation and utility prediction. Text and visual token representations are projected independently to 256 dimensions. A single four-head text-to-vision cross-attention layer uses the projected text tokens as queries and the projected visual tokens as keys and values. The attention output is followed by LayerNorm, and the EOS-position output is used as the 256-dimensional query-conditioned visual representation $\mathbf{h}^{\times}$. The attention layer uses dropout 0.1; no additional residual or feed-forward block is introduced.

The router input is

$$\mathbf{z} = [\mathbf{t}_{\text{global}}; \mathbf{v}_{\text{global}}; r; \ln(r + 10^{-8}); \mathbf{h}^{\times}] \in \mathbb{R}^{1538}.$$

A two-layer MLP,

$$1538 \to 512 \to 5,$$

with GELU activation and dropout 0.1, predicts the utilities of

$$\{\text{FastV}, \text{VisionZip}, \text{PruMerge}^{+}, \text{DART}, \text{Full}\}.$$

Full is treated as an ordinary candidate whose utility is predicted by the router rather than analytically fixed. The complete router contains 1,382,405 trainable parameters.

Standardization. All standardization statistics are estimated exclusively from the corresponding training split. The frozen global preview features and the two ratio features are standardized feature-wise using their training-set means and standard deviations.

Utility targets are standardized separately for each candidate. For candidate k, let μ_k and σ_k denote the mean and standard deviation of its training-set utility. The regression target is

$$\widetilde{U}_{i,k} = \frac{U_{i,k} - \mu_k}{\sigma_k}. \tag{13}$$

For constant dimensions, including the Full utility under Group A, the standardization scale is set to 1. At inference time, predicted standardized utilities are transformed back to the original utility scale,

$$\hat{U}_{i,k} = \hat{\widetilde{U}}_{i,k}\sigma_k + \mu_k, \tag{14}$$

and the candidate with the largest predicted utility is selected. Validation and test samples do not contribute to any standardization statistics.

Optimization. We train the projection layers, cross-attention module, LayerNorm, and utility predictor using AdamW with learning rate 10^{-3}, weight decay 10^{-4}, and batch size 256. Training runs for at most 50 epochs with a linear warm-up over the first 5% of optimization steps followed by cosine learning-rate decay. Gradients are clipped to a global norm of 1.0.

The objective is an element-wise Huber loss with $\delta = 1$, averaged over samples and the five candidate utilities in the standardized target space. Training is performed in FP32. We use early stopping with patience five. For checkpoint selection, validation utility is first averaged within each retained-token ratio and then macro-averaged across the five ratios. The checkpoint with the highest validation macro utility is used for final test evaluation.

Frozen CLIP representations are precomputed once for each unique image–question record and reused across its available retained-token ratios; all trainable VIP-Router components are evaluated online.

Unless otherwise stated, VIP-Router results within each evaluation split are averaged over five independent training runs with random seeds $0, 1, 2, 3, 4$. For a given split, all five runs use the same data partition, cached preview features, architecture, target standardization procedure, and optimization configuration; only the random training trajectory differs. For the image-disjoint evaluation, this five-seed training procedure is repeated independently on each of the three image-group partitions described above.

A.3.3 Hardware and Software Environment

All experiments are conducted on a single NVIDIA H200 GPU. We use Python 3.10.12, PyTorch 2.3.0 with CUDA 12.4, and Transformers 4.49.0. The CLIP preview encoder is implemented with `transformers.CLIPModel` and `CLIPProcessor`. Router training is performed in FP32.

FLOPs accounting. The FLOPs reported in Figure 3 are profiler-based operator counts obtained with `torch.profiler`, rather than analytical estimates. We count one multiply–accumulate operation as two FLOPs. The reported computation corresponds to a single multimodal prefill forward pass; autoregressive decoding and KV-cache reuse are excluded, and therefore no output-length assumption is required.

For VIP-Router, the FLOPs include the frozen CLIP preview encoding, the trainable routing modules, the selected pruning operation, and the subsequent MLLM forward pass. For fixed pruning baselines, the corresponding pruning operation and MLLM forward pass are counted without the additional routing computation. Non-tensor preprocessing such as image decoding/resizing and text tokenization is not included in the FLOPs count. Because the counts are obtained from profiler-supported tensor operators, operations not exposed to the profiler as explicit FLOPs may be omitted.

Table 6: Parameter overhead and performance of the default VIP-Router and its native-feature variants. Extra frozen parameters include modules loaded exclusively for routing beyond Qwen2-VL; reused native MLLM components are excluded. Percentages are relative to Qwen2-VL parameters. All variants in this table are single-run results; consequently, the default single-run result differs from the five-seed mean reported in Table 7.

Vision Features	Text Features	Cross Attn.	# Extra Frozen Params	# Trainable Params	Acc.	Utility
CLIP ViT-B/32	CLIP Text	Yes	151.28 M (1.82%)	1.38 M (0.017%)	51.51	**38.09**
Native ViT	CLIP Text	Yes	63.43 M (0.77%)	1.78 M (0.021%)	51.57	37.98
Native ViT	Native LM	Yes	0.00 M (0.00%)	4.13 M (0.050%)	51.86	37.37
Native ViT	CLIP Text	No	63.43 M (0.77%)	**0.92 M** (0.011%)	50.47	37.19
Native ViT	Native LM	No	0.00 M (0.00%)	2.49 M (0.030%)	52.21	37.62

Table 7: Performance of VIP-Router with alternative frozen preview encoders. The default CLIP-B/32 result is averaged over five random seeds, whereas the alternative encoder variants are single-run results and are therefore intended as robustness evidence rather than a strict encoder ranking.

Variant	**Vision Encoder**	**Text Encoder**	**Acc.**	**Cost**	**Utility**
Default	CLIP ViT-B/32	CLIP Text	51.19	0.135	37.68
CLIP-B/16	CLIP ViT-B/16	CLIP Text	50.31	0.130	37.28
SigLIP2-B/32	SigLIP2-B/32	SigLIP2 Text	47.38	0.110	36.36
TIPSv2-B/14	TIPSv2-B/14	TIPSv2 Text	49.87	0.125	37.34
EUPE + CLIP Text	EUPE-ViT-B	CLIP Text	51.12	0.132	37.96
EUPE + SigLIP2 Text	EUPE-ViT-B	SigLIP2 Text	50.47	0.131	37.34

Accordingly, these values are used as a consistent computational-cost estimate across methods rather than as measurements of end-to-end wall-clock latency.

A.4 Encoder Analysis

Parameter overhead and native-feature reuse. The default VIP-Router uses frozen CLIP visual and textual encoders for preview feature extraction and introduces only 1.38M trainable parameters, corresponding to 0.017% of the Qwen2-VL backbone. Table 6 examines how this footprint changes when the external preview encoders are partially or fully replaced with native MLLM representations. Reusing native features can eliminate the additional frozen parameter footprint entirely while keeping the trainable router below 0.05% of the backbone. For example, the fully native variant without cross-attention requires no external frozen parameters and 2.49M trainable parameters, while remaining within 0.5 utility points of the default single-run configuration. Thus, the external CLIP encoders are a design choice rather than a requirement of the routing formulation.

Encoder alternatives. The default VIP-Router uses CLIP ViT-B/32 and its paired text encoder to construct the preview representation. To examine whether the routing framework remains effective with alternative preview encoders, we replace the default frozen encoders while keeping the routing objective, candidate strategy set, dataset split, and architecture unchanged. All variants retain the same text-to-vision cross-attention module, EOS-style pooling, ratio conditioning, and two-layer utility predictor.

As shown in Table 7, VIP-Router remains effective across substantially different preview encoder families. CLIP-B/16 (Radford et al., 2021), TIPSv2-B/14 (Cao et al., 2026), and both EUPE (Zhu et al., 2026) variants achieve utility close to the default CLIP-B/32 configuration, while SigLIP2-B/32 (Tschannen et al., 2025) exhibits a somewhat larger decrease. Notably, the EUPE–CLIP variant achieves slightly higher utility than the default configuration, with comparable accuracy. These results indicate that the routing formulation is not restricted to a single preview feature family and can operate with multiple frozen vision–language representations.

We do not interpret the small differences among encoder variants as a strict ranking. The alternative configurations were evaluated as single runs, and their training RNG trajectories are not perfectly

Table 8: Comparison with the sample-adaptive OccamToken baseline on Qwen2-VL-7B and VTC-Bench Group A under the record-disjoint evaluation protocol. OccamToken is calibrated separately at each pruning level using 100 source-validation samples and retained-token counts only. Accuracy and Utility are multiplied by 100 for reporting, whereas Cost is the realized mean retained-token ratio. VIP-Router results are averaged over five random seeds; Best Fixed and OccamToken are deterministic under the reported configurations. The Average row is the macro-average over the five reduction ratios.

Reduction Ratio	Best Fixed			OccamToken			**VIP-Router**		
	Acc.	**Cost**	**Utility**	**Acc.**	**Cost**	**Utility**	**Acc.**	**Cost**	**Utility**
75.00%	57.54	0.250	32.54	25.29	0.328	-7.52	66.91	0.287	**38.23**
88.89%	47.94	0.111	36.83	16.60	0.153	1.28	57.65	0.144	**43.24**
93.75%	37.42	0.063	31.17	20.93	0.063	14.60	49.62	0.095	**40.08**
96.00%	33.05	0.040	29.05	23.67	0.053	18.37	44.29	0.073	**36.95**
99.00%	25.82	0.010	24.82	30.35	0.042	26.12	37.48	0.076	**29.89**
Average	40.35	0.095	30.88	23.37	0.128	10.57	51.19	0.135	**37.68**

matched across variants. Accordingly, this experiment is intended to assess the robustness of VIP-Router to encoder replacement rather than to identify an optimal preview encoder.

A.5 Additional Results

A.5.1 Image-Disjoint Evaluation

Image identity and split construction. We construct three image-disjoint splits from the same routing dataset, containing 33,091 record–ratio examples from 12,919 image–question records and 4,372 unique decoded images. Image identity is determined by a SHA-256 hash of the decoded RGB image dimensions and pixel values rather than file paths or document identifiers. All records sharing the same decoded image are grouped together, including different questions, ratio-specific examples, and duplicates across benchmarks. Each image group is assigned entirely to training, validation, or test, ensuring that the three partitions share no identical decoded images.

Splits I, II, and III correspond to grouping seeds 826, 360, and 266, respectively. Grouping seeds are selected using only the balance of benchmark–ratio composition relative to the target $70/15/15$ partition proportions; candidate correctness, utility values, router predictions, and evaluation performance are not used for this selection. Table 9 reports the resulting partition sizes. The image-disjoint training sets contain 3,053–3,072 unique images, compared with 3,566 in the primary record-disjoint training set.

Group A membership, candidate pruning outcomes, retained-token ratios, the utility objective, and the router architecture remain unchanged. For each split, we reselect the fixed-strategy baselines, refit feature and utility standardization statistics using only the training partition, and train VIP-Router with training seeds $0, 1, 2, 3, 4$. Checkpoint selection uses the corresponding validation partition. Means and sample standard deviations are computed across the five training runs separately for each image-disjoint split.

Per-ratio performance and training-seed consistency. Table 10 supplements the macro-average comparison in Table 2 with results at each reduction ratio. Best Fixed selects VisionZip in all three splits. We additionally report Best Fixed@Ratio as a ratio-conditioned reference: it selects one pruning strategy separately for each ratio using only the corresponding training data. This reference selects PruMerge$^+$ at 93.75% reduction in all three splits, DART at 75.00% in Split II, and VisionZip otherwise. These are training-selected strategies rather than retrospective test-set winners.

VIP-Router achieves higher five-seed mean accuracy and utility than Best Fixed at every reduction ratio in all three splits. The positive mean utility gains also hold against Best Fixed@Ratio, showing that the improvement extends beyond selecting a different fixed strategy for each pruning level. The margins are smallest at 88.89% reduction, where individual training runs can fall below the fixed-

Table 9: Partition statistics for the three image-disjoint splits. Records denote unique image–question records; rows denote record–ratio examples. Image overlap is the number of decoded images shared by any pair of training, validation, and test partitions. Grouping seeds determine the data partitions and are distinct from the five router-training seeds.

Split	Grouping seed	Unique images			Records			Rows			Image overlap
		Train	Val.	Test	Train	Val.	Test	Train	Val.	Test	
I	826	3,053	643	676	9,063	1,923	1,933	23,124	4,990	4,977	0
II	360	3,063	667	642	8,976	2,006	1,937	23,104	5,013	4,974	0
III	266	3,072	623	677	9,005	1,933	1,981	23,056	4,909	5,126	0

Table 10: Per-ratio performance under the three image-disjoint splits. Best Fixed is the primary baseline; Best Fixed@Ratio is an additional training-selected, ratio-conditioned reference. VIP-Router accuracy and cost are five-seed means, while its utility is reported as mean $\pm$ sample standard deviation. Accuracy and utility are multiplied by 100; token cost remains on the $[0, 1]$ scale. Both fixed-strategy baselines have token cost r. Oracle accuracy is 100% by Group A construction; Oracle utility is shown in gray and excluded from highlighting. OHR is computed relative to Best Fixed. Macro rows average metrics, including per-ratio OHR, equally across the five ratios; their test-row counts are totals. The best non-oracle mean utility in each row is bolded.

Reduction ratio	Test rows	Oracle	Best Fixed	Best Fixed @Ratio	VIP-Router			OHR
		Utility	Utility	Utility	Acc.	Cost	Utility	
Image-disjoint I								
75.00%	409	62.71	27.57	27.57	57.02	0.271	$\mathbf{29.89} \pm 3.24$	6.61%
88.89%	782	65.59	36.20	36.20	48.93	0.126	$\mathbf{36.29} \pm 2.18$	0.30%
93.75%	1,063	59.44	32.88	34.95	43.39	0.077	$\mathbf{35.64} \pm 1.46$	10.37%
96.00%	1,181	53.49	29.95	29.95	37.32	0.057	$\mathbf{31.58} \pm 1.40$	6.93%
99.00%	1,542	41.41	26.89	26.89	33.23	0.045	$\mathbf{28.76} \pm 0.96$	12.92%
Macro	4,977	56.53	30.70	31.11	43.98	0.115	$\mathbf{32.43} \pm 0.84$	7.42%
Image-disjoint II								
75.00%	438	62.50	30.94	29.11	58.90	0.273	$\mathbf{31.61} \pm 1.44$	2.13%
88.89%	785	64.20	34.24	34.24	48.10	0.134	$\mathbf{34.70} \pm 1.33$	1.55%
93.75%	1,057	58.27	31.21	32.07	41.67	0.087	$\mathbf{33.00} \pm 0.80$	6.61%
96.00%	1,135	50.92	26.40	26.40	34.70	0.069	$\mathbf{27.79} \pm 0.77$	5.67%
99.00%	1,559	41.09	24.91	24.91	30.97	0.050	$\mathbf{25.99} \pm 0.94$	6.67%
Macro	4,974	55.40	29.54	29.35	42.87	0.122	$\mathbf{30.62} \pm 0.68$	4.53%
Image-disjoint III								
75.00%	428	62.03	29.91	29.91	57.38	0.257	$\mathbf{31.72} \pm 1.08$	5.64%
88.89%	827	65.35	37.98	37.98	50.18	0.121	$\mathbf{38.13} \pm 0.73$	0.52%
93.75%	1,097	59.91	32.40	34.77	43.03	0.073	$\mathbf{35.72} \pm 1.00$	12.05%
96.00%	1,203	52.67	29.08	29.08	34.91	0.052	$\mathbf{29.70} \pm 0.69$	2.61%
99.00%	1,571	39.39	26.05	26.05	30.86	0.031	$\mathbf{27.75} \pm 0.37$	12.76%
Macro	5,126	55.87	31.09	31.56	43.27	0.107	$\mathbf{32.60} \pm 0.51$	6.72%

strategy baseline. Nevertheless, all 15 training runs improve macro-average utility over Best Fixed, as shown in Table 11; all also exceed the ratio-conditioned reference. Thus, the aggregate routing advantage is reproduced across both image-group partitions and training initializations, without implying that every individual run wins at every ratio.

Seen- versus unseen-image test records. We additionally examine the original record-disjoint test set while keeping the training data, five trained routers, and training-selected baselines fixed. Test records are divided according to whether their decoded image appears in the original training

Table 11: Macro-average utility of individual VIP-Router training runs under each image-disjoint split. Utility is multiplied by 100 and averaged equally across the five reduction ratios. Columns 0–4 identify training seeds, not image-grouping seeds. Best Fixed and the Best Fixed@Ratio reference are selected using the corresponding training partition and remain unchanged across training seeds.

Split	Best Fixed	Best Fixed @Ratio	VIP-Router training seed				
			0	1	2	3	4
I	30.70	31.11	31.69	31.75	33.16	32.07	33.49
II	29.54	29.35	29.67	31.21	30.61	31.31	30.30
III	31.09	31.56	32.62	33.41	32.11	32.67	32.20

Table 12: Seen- and unseen-image analysis within the primary record-disjoint protocol. Test rows count record–ratio decisions rather than unique images. Utility is macro-averaged across the five reduction ratios and multiplied by 100; VIP-Router values are means ± sample standard deviations across five training seeds. Best Fixed@Ratio is included only as a ratio-conditioned reference. ΔU and OHR are measured relative to Best Fixed on the same subset; OHR is averaged over the per-ratio values. Seeds ↑ counts runs with positive macro-average utility gains over Best Fixed.

Test subset	Test rows	Best Fixed	Best Fixed @Ratio	VIP-Router	ΔU	OHR	Seeds ↑
		Utility	Utility	Utility			
Seen images	3,565	35.53	35.36	$\mathbf{42.36} \pm 0.76$	+6.83	26.66%	5/5
Unseen images	1,475	19.72	20.50	$\mathbf{26.54} \pm 0.52$	+6.82	27.08%	5/5

partition. The seen-image subset contains 1,495 image–question records from 882 unique images, contributing 3,565 record–ratio decisions. The unseen-image subset contains 480 records from 461 unique images, contributing 1,475 decisions. Here, image novelty is defined relative to the router's training partition; this subset analysis does not change the original validation partition. Neither the router nor either fixed-strategy baseline is retrained or reselected for these subsets.

Table 12 shows that the routing gains under the primary protocol are not confined to test images that also occur in training. On the unseen-image subset, VIP-Router improves utility from 19.72 to 26.54 over Best Fixed, a 34.6% relative improvement, with positive macro-average gains in all five training runs. Its gain over the Best Fixed@Ratio reference is also positive in every run, averaging 6.03 utility points. Thus, substantial routing gains are also observed on images absent from the router's training set without changing the original trained models.

The two subsets have markedly different benchmark compositions. ChartQA accounts for 49.4% of unseen-image decision rows but only 10.7% of seen-image rows; OCRBench and MMStar are also more prevalent in the unseen-image subset, whereas GQA, POPE, and MMBench dominate the seen-image subset. Consequently, the difference between their gains is descriptive rather than a causal estimate of image memorization. Likewise, constructing image-disjoint partitions changes both training support and test composition, so the difference between the two protocols cannot isolate the effect of repeated images alone. Together, the grouped-split results and the within-protocol subset analysis show that sample-adaptive routing benefits extend beyond exact training-image recurrence.

Vision-only routing under image-disjoint evaluation. We further examine whether the strong global visual signal observed in the feature ablation persists when test images are completely unseen. For this diagnostic, we retrain the V_{global}-only ratio-specific router independently on each image-disjoint split. This variant uses only the mean-pooled global CLIP visual representation and trains a separate two-layer MLP router for each retained-token ratio, without text features, spatial visual tokens, cross-attention, or explicit ratio conditioning. It therefore corresponds to the V_{global}-only control in Stage A of Table 4.

This vision-only variant is not architecturally matched to the default VIP-Router. The default model instead uses a single shared ratio-conditioned router with global text and visual features together with query-conditioned cross-modal interaction. Accordingly, the difference between the two con-

Table 13: Vision-only routing under image-disjoint evaluation. The vision-only variant uses the mean-pooled global CLIP visual representation and five separate ratio-specific routers. Best Fixed@Ratio is included as a ratio-conditioned reference. Values for the vision-only router and the default VIP-Router are means over five training seeds; accuracy and utility are multiplied by 100. ΔU denotes the vision-only router's utility gain over Best Fixed@Ratio. Because the vision-only variant and the default VIP-Router differ in router topology and cross-ratio parameter sharing, their performance difference is not a single-factor estimate of the contribution of cross-modal interaction.

Split	Best Fixed@Ratio Utility	Vision-only ratio-specific router			Seeds ↑	Default VIP-Router Utility
		Acc.	Utility	ΔU		
Image-disjoint I	31.11	44.60	32.21 ± 0.23	+1.10	5/5	32.43 ± 0.84
Image-disjoint II	29.35	42.76	30.17 ± 0.38	+0.83	5/5	30.62 ± 0.68
Image-disjoint III	31.56	43.96	31.87 ± 0.69	+0.31	4/5	32.60 ± 0.51

figurations should not be interpreted as a single-factor estimate of the contribution of cross-modal interaction.

The vision-only ratio-specific router retains a positive mean utility gain over the ratio-conditioned fixed reference on all three image-disjoint splits. This shows that global visual features contain routing-relevant information that transfers to previously unseen images rather than being useful only when the same image recurs across training and test. However, the vision-only router and the default VIP-Router differ in both router topology and cross-ratio parameter sharing, so their absolute performance difference cannot be used to isolate the contribution of cross-modal interaction.

A.5.2 Comparison with Sample-Adaptive Visual Token Pruning

We further compare VIP-Router with OccamToken (Li et al., 2026a), a training-free visual token pruning method that adaptively retains different numbers of visual tokens for different samples. OccamToken is particularly suitable as an adaptive baseline in our setting because its pruning threshold can be calibrated so that the average retained token budget on a calibration set matches a prescribed operating point.

We reproduce the two-stage Qwen configuration of OccamToken on Qwen2-VL-7B. Stage I uses the mutual-scoring variant for models without an explicit CLS token, and Stage II performs text-conditioned pruning at decoder layer 10. For each retained-token ratio, we calibrate the budget-control parameter λ_2 using 100 fixed samples from the source validation split. Calibration uses only realized retained-token counts and never accesses correctness labels, utility values, or test-set statistics. The calibrated parameter is then fixed for evaluation on the same record-disjoint Group A test split used by Best Fixed and VIP-Router. Because OccamToken allocates tokens adaptively across samples, all reported costs are computed from the actual number of retained visual tokens rather than from the nominal target ratio.

As shown in Table 8, VIP-Router achieves higher Accuracy and Utility than OccamToken at all five evaluated pruning levels. The difference is particularly clear in the macro-average comparison: OccamToken and VIP-Router incur comparable realized token costs (0.128 versus 0.135), while VIP-Router achieves substantially higher Accuracy (51.19% versus 23.37%) and Utility (37.68 versus 10.57). OccamToken nevertheless shows the benefit of sample-adaptive token allocation at the most aggressive operating point: at 99% reduction, it improves over Best Fixed in both Accuracy (30.35% versus 25.82%) and Utility (26.12 versus 24.82%). VIP-Router remains higher at the same operating point, reaching 37.48% Accuracy and 29.89 Utility.

The reported OccamToken results should be interpreted as a paper-based reproduction rather than an official implementation. At the time of our experiments, no official OccamToken code was available, and several Qwen2-VL-specific implementation details are not fully specified in the paper. We therefore follow the reported Qwen design as closely as possible and use the same implementation consistently across all pruning levels. In addition, at the 99% reduction setting, OccamToken cannot attain the nominal 1% retained-token target because its register token imposes a minimum effective

Table 14: Comparison with a ratio-conditioned fixed-strategy control. Best Fixed uses a single pruning strategy selected jointly over all source-training samples and reduction ratios, yielding VisionZip for all operating points. Best Fixed@Ratio independently selects one pruning strategy on the source training split for each reduction ratio and applies that strategy uniformly to all test samples at the corresponding ratio. The selected strategy therefore does not necessarily coincide with the strategy that performs best retrospectively on the test split. VIP-Router results are reported as mean ± standard deviation over five random seeds. Utility values are multiplied by 100 for reporting. The Average row is the macro-average over the five reduction ratios.

Reduction Ratio	Best Fixed	Best Fixed@Ratio		VIP-Router	Δ vs. Best Fixed@Ratio
		Strategy	Utility		
75.00%	32.54	DART	30.68	$\mathbf{38.23 \pm 1.39}$	$\mathbf{+7.54 \pm 1.39}$
88.89%	36.83	VisionZip	36.83	$\mathbf{43.24 \pm 1.16}$	$\mathbf{+6.41 \pm 1.16}$
93.75%	31.17	PruMerge$^+$	33.48	$\mathbf{40.08 \pm 0.92}$	$\mathbf{+6.60 \pm 0.92}$
96.00%	29.05	VisionZip	29.05	$\mathbf{36.95 \pm 0.65}$	$\mathbf{+7.90 \pm 0.65}$
99.00%	24.82	VisionZip	24.82	$\mathbf{29.89 \pm 0.45}$	$\mathbf{+5.07 \pm 0.45}$
Average	30.88	—	30.97	$\mathbf{37.68 \pm 0.64}$	$\mathbf{+6.70 \pm 0.64}$

budget: the closest validation-set calibration retains 4.57% of visual tokens on average. We leave this behavior unchanged and compute test Utility using the actual realized token cost.

A.5.3 Ratio-Conditioned Fixed-Strategy Control

Throughout the main paper, we use *Best Fixed* as the primary fixed-strategy baseline. Best Fixed selects a single pruning strategy on the source training split by maximizing average utility jointly over all training samples and reduction ratios, and then applies that strategy to every test sample at every ratio. It corresponds directly to the fixed-criterion setting studied in this work, in which the pruning level controls how aggressively visual tokens are reduced while the criterion itself stays fixed, and it also matches the granularity at which such methods are compared in practice: superiority is established in aggregate across reduction ratios and benchmarks rather than at individual operating points.

VIP-Router, however, is additionally conditioned on the target reduction ratio r, so part of its advantage could in principle come from adapting the criterion to the pruning level rather than to the sample. To separate these two axes, we evaluate a ratio-conditioned control, denoted *Best Fixed@Ratio*. For each reduction ratio r, it independently selects

$$k_r^* = \arg\max_{k \in \mathcal{S}_{\text{prune}}} \frac{1}{|\mathcal{D}^r_{\text{train}}|} \sum_{i \in \mathcal{D}^r_{\text{train}}} U_{i,k},$$

using only the source-training samples at that ratio, and applies k_r^* uniformly to all test samples at the same ratio. Best Fixed@Ratio therefore adapts across reduction ratios, remains non-adaptive across samples within each ratio, and never consults test labels for strategy selection. Because its selection is made on the training split, it is not necessarily stronger than Best Fixed at a given ratio; its role is attributive rather than competitive.

As shown in Table 14, allowing the fixed strategy to vary with the reduction ratio changes macro-average utility only marginally, from 30.88 for Best Fixed to 30.97 for Best Fixed@Ratio. VIP-Router reaches 37.68, exceeding Best Fixed@Ratio by 6.7 on average and by 5.07–7.9 at every evaluated reduction ratio. Essentially none of VIP-Router's gain is therefore attributable to ratio-level adaptation: the available headroom lies within each operating point and is recovered through sample-level strategy selection.

A.5.4 Detailed Per-Benchmark Results

Table 15 reports the complete per-benchmark accuracy results underlying the aggregate comparison in Section 4.2. VIP-Router achieves the best pruning result in 30 of the 40 benchmark–ratio combinations (75.0%), showing that its improvement is broadly distributed across benchmarks rather than

Table 15: Per-benchmark accuracy (%) of fixed pruning strategies and VIP-Router on VTC-Bench Group A test set across visual-token reduction ratios. Full is shown in gray as a reference and has 100% accuracy by the construction of Group A; it is excluded from highlighting. Fixed-strategy results are deterministic, whereas VIP-Router results are averaged over five random seeds. Within each reduction-ratio block, the best pruning result in each column is bolded, and the second-best result in the Overall column is underlined.

Method	GQA	MMB	MMB^{CN}	MME	POPE	MMStar	OCR	ChartQA	Overall
Full	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
Reduction ratio = 75.00%									
FastV	61.59	54.05	43.33	87.50	69.44	55.88	25.81	39.71	54.99
VisionZip	60.26	54.05	33.33	62.50	80.56	**58.82**	19.35	**55.88**	<u>57.54</u>
PruMerge+	**64.24**	62.16	40.00	62.50	77.78	52.94	22.58	42.65	57.31
DART	56.95	51.35	46.67	87.50	73.61	50.00	51.61	41.18	55.68
VIP-Router	64.11	**63.24**	**66.67**	**90.00**	**90.83**	50.59	**56.77**	47.65	**66.91**
Reduction ratio = 88.89%									
FastV	42.45	36.21	28.81	41.18	43.84	25.00	10.00	32.26	35.33
VisionZip	46.70	27.59	33.90	41.18	74.66	28.12	20.00	**51.61**	<u>47.94</u>
PruMerge+	47.64	25.86	22.03	47.06	71.23	31.25	10.00	41.01	43.20
DART	49.06	32.76	37.29	**58.82**	60.96	**46.88**	33.33	26.73	42.07
VIP-Router	**53.58**	**46.90**	**49.15**	57.65	**83.70**	34.38	**47.00**	51.52	**57.65**
Reduction ratio = 93.75%									
FastV	38.08	36.49	25.00	**52.94**	38.71	31.82	8.11	15.22	29.22
VisionZip	38.43	28.38	27.94	38.24	70.97	31.82	8.11	27.17	37.42
PruMerge+	46.26	27.03	22.06	38.24	73.12	31.82	9.46	27.90	<u>39.73</u>
DART	44.48	25.68	39.71	44.12	59.14	**34.09**	31.08	15.22	36.26
VIP-Router	**49.82**	**43.24**	**46.47**	48.24	**85.27**	31.36	**43.51**	**29.35**	**49.62**
Reduction ratio = 96.00%									
FastV	28.49	28.42	27.62	27.59	37.32	31.48	7.50	8.03	23.92
VisionZip	37.09	33.68	28.57	37.93	66.99	33.33	6.25	10.95	33.05
PruMerge+	39.17	25.26	22.86	34.48	62.20	24.07	5.00	15.69	32.12
DART	41.84	24.21	39.05	**44.83**	56.46	35.19	31.25	12.04	<u>34.91</u>
VIP-Router	**43.50**	**40.63**	**46.48**	40.69	**78.47**	**35.93**	**41.50**	**20.07**	**44.29**
Reduction ratio = 99.00%									
FastV	18.57	14.71	20.73	19.35	45.49	15.87	2.73	3.28	18.89
VisionZip	24.90	26.47	34.15	33.87	51.76	28.57	0.00	5.84	<u>25.82</u>
PruMerge+	22.04	15.29	16.46	24.19	53.33	23.81	1.82	7.30	21.98
DART	24.69	19.41	29.27	**37.10**	38.43	26.98	30.91	3.65	24.18
VIP-Router	**27.14**	**37.76**	**43.66**	36.45	**61.57**	**32.06**	**41.27**	**25.47**	**37.48**

driven by a small subset of tasks. Its gains also persist under the most aggressive reduction settings, where the accuracy of all fixed pruning strategies degrades substantially. Full-token inference is included only as a reference and has 100% accuracy by the construction of VTC-Bench Group A. Among the individual fixed pruning strategies, the strongest test-set result varies with the pruning level: VisionZip achieves the highest overall accuracy at 75.00%, 88.89%, and 99.00% reduction, whereas PruMerge+ and DART are strongest at 93.75% and 96.00%, respectively. VIP-Router nevertheless achieves higher overall accuracy than every fixed pruning strategy at every evaluated reduction ratio.

Because all four fixed strategies retain exactly r of the visual tokens, their utility is accuracy minus $100r$, so the accuracy ordering above is also their utility ordering. The best fixed utilities are therefore 32.54, 36.83, 33.48, 30.91, and 24.82 at the five ratios, all below the 38.23, 43.24, 40.08,

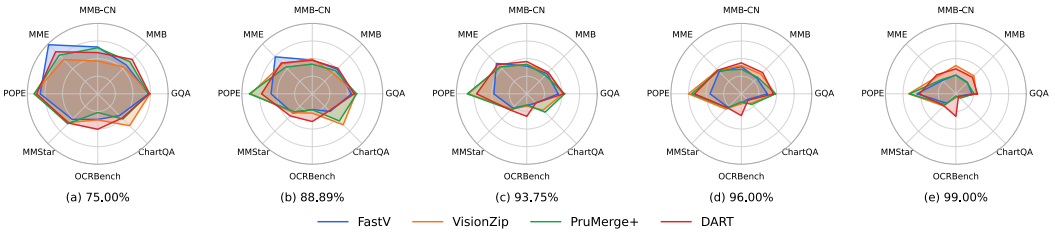


Figure 4: Per-benchmark accuracy of the four fixed pruning strategies on VTC-Bench Group A.

Table 16: Recoverable VisionZip failures on the full VTC-Bench evaluation set without Group A filtering. A recoverable failure is a sample for which VisionZip is incorrect while at least one alternative pruning strategy (FastV, PruMerge+, or DART) is correct. The final column reports the fraction of VisionZip failures recoverable by an alternative pruning strategy.

Reduction Ratio	All Samples	VisionZip Wrong	Recoverable Failures	% of All Samples	% of VisionZip Failures
75.00%	37,610	10,762	3,819	10.15%	35.49%
88.89%	37,610	13,055	5,222	13.88%	40.00%
93.75%	37,610	15,227	6,056	16.10%	39.77%
96.00%	37,610	16,345	5,935	15.78%	36.31%
99.00%	37,610	19,046	4,219	11.22%	22.15%

36.95, and 29.89 of VIP-Router in Table 1, which supports the corresponding claim in Section 4.2 despite the higher realized token cost of routing.

These per-ratio fixed-strategy rankings are descriptive test-set comparisons and should not be confused with Best Fixed@Ratio in Appendix A.5.3, whose strategy at each ratio is selected exclusively from the source training split.

A.5.5 Complementarity among Pruning Strategies

Figure 4 first shows complementarity at the benchmark level: the relative performance of fixed pruning strategies varies across benchmarks, and no single pruning criterion is uniformly strongest. This variation persists at the sample level. On Group A, a substantial fraction of samples for which the source-selected Best Fixed strategy (VisionZip) fails can still be correctly handled by at least one alternative pruning strategy. Thus, benchmark-average rankings obscure recoverable failures at the individual-sample level, complementing the Best Fixed–Oracle headroom reported in the main text. Because Group A explicitly selects pruning-sensitive samples, we further test whether this complementarity persists without Group A filtering. Table 16 reports the same recovery analysis over the full VTC-Bench evaluation set. Across reduction ratios, 10.15%–16.10% of all samples are cases where VisionZip fails but at least one of FastV, PruMerge+, or DART succeeds. Among VisionZip failures themselves, 22.15%–40.00% are recoverable by an alternative pruning criterion. Pooling all five reduction ratios gives 25,251 recoverable failures among 188,050 sample–ratio decisions, corresponding to 13.43% of all decisions and 33.92% of VisionZip failures. These results show that the sample-level complementarity observed on Group A is not solely a consequence of pruning-sensitive sample selection, but persists over the full evaluation set.

A.5.6 VIP-Router Selection Behavior

Figure 5 shows the strategy-selection distribution of VIP-Router across reduction ratios. The routing decisions remain distributed across all four pruning methods at every operating point, indicating that VIP-Router does not collapse to another globally fixed pruning policy. The relative preference among pruning strategies also changes with the pruning level; for example, DART is selected increasingly often under aggressive reduction, whereas VisionZip remains a substantial component throughout the evaluated range.

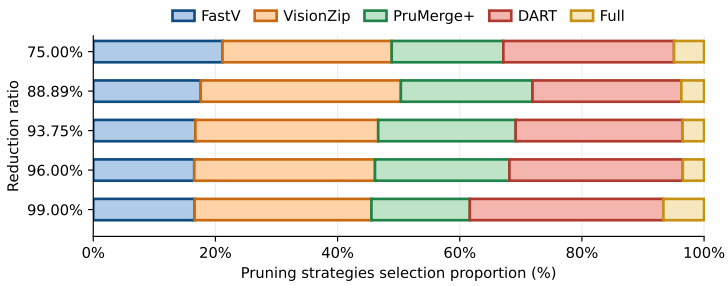


Figure 5: VIP-Router strategy-selection distribution across visual-token reduction ratios. Each bar reports the mean fraction of test samples routed to FastV, VisionZip, PruMerge+, DART, or Full over five random seeds.

Table 17: Routing-error diagnosis of VIP-Router on VTC-Bench Group A. Values are means across five random seeds. Mismatch percentages are fractions of all oracle-argmax mismatches, and regret percentages are fractions of total regret; both pool all test record–ratio pairs within each run rather than macro-averaging across ratios. Mean regret is computed within each decision group on the raw utility scale. Neighborhood statistics use $k = 10$.

Decision group	**Mismatch (%)**	**Regret (%)**	**Mean regret**	**Neighbor agreement**	**Neighbor entropy (nats)**
Oracle-argmax match	–	0	0	0.436	0.885
All oracle-argmax mismatches	100	100	0.234	0.444	0.905
Near-tie	59.1	36.5	0.145	0.485	0.865
High-margin	40.9	63.5	0.363	0.385	0.963

Full-token inference remains a minority choice. VIP-Router selects Full for 4.92%, 3.72%, 3.51%, 3.48%, and 6.66% of samples at reduction ratios of 75.00%, 88.89%, 93.75%, 96.00%, and 99.00%, respectively. Thus, the accuracy improvement is not obtained by routinely reverting to full-token inference. Together with the counterfactual bound in Remark 1, these results indicate that the primary gain comes from sample-adaptive selection among heterogeneous pruning strategies.

A.5.7 Routing Error Diagnosis

We analyze the saved test predictions of the final VIP-Router across five random seeds, with 5,040 image–question-record–ratio decisions per seed. Let $s_i^\star$ denote the oracle argmax returned by the evaluation procedure and $\hat{s}_i$ the router's selected strategy. We distinguish an oracle-argmax mismatch, $\hat{s}_i \neq s_i^\star$, from the incurred utility regret:

$$\Delta_i = U_i(s_i^\star, r_i) - U_i(\hat{s}_i, r_i).$$

An argmax mismatch is not a VQA answer error and may incur zero regret when multiple strategies attain the same optimal utility. For mismatched decisions, we define the oracle margin as

$$m_i = U_i^{(1)} - U_i^{(2)},$$

where $U_i^{(1)}$ and $U_i^{(2)}$ are the two highest candidate utilities, including ties. We classify mismatches as near-tie when $m_i \leq 0.03$ and high-margin otherwise.

Strategy disagreement versus utility loss. Table 17 shows that high-margin mismatches account for 40.9% of strategy disagreements but contribute 63.5% of total regret. Their contribution to the remaining utility loss is therefore substantially larger than their frequency alone suggests. Conversely, approximately 28.9% of all argmax mismatches incur zero regret, reflecting utility-equivalent choices rather than suboptimal routing. Near-tie mismatches nevertheless contribute 36.5% of total regret: a small gap between the two best candidates does not bound the loss from selecting a lower-utility candidate. Thus, exact agreement with a single oracle argmax is insufficient to assess routing quality; the utility consequences of a decision must also be considered.

Local structure of routing errors. To examine the feature-space neighborhoods of these decisions, we retrieve the ten nearest training examples at the same retained-token ratio using cosine similarity. The representation consists of the frozen global preview features and ratio inputs,

$$[\mathbf{t}_{\text{global}}; \mathbf{v}_{\text{global}}; r; \ln(r + 10^{-8})],$$

with training-set feature standardization followed by L2 normalization. The learned cross-attention representation is not included. Neighbor agreement measures the fraction of retrieved examples sharing the test example's oracle-argmax label, while neighbor entropy measures the entropy of the retrieved oracle-label distribution.

High-margin mismatches have lower neighbor agreement and higher neighbor entropy than oracle-matched decisions. Importantly, the lower-agreement pattern does not hold for argmax mismatches as a whole, indicating that the association is specific to the high-margin subset rather than strategy disagreement in general. These results associate high-margin routing errors with locally mixed oracle labels in the frozen global preview space. They characterize the structure of the remaining errors without establishing a representational limitation of the complete learned cross-modal router.