REROUTE, DON'T REMOVE: RECOVERABLE VISUAL TOKEN ROUTING FOR VISION-LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision-language models (VLMs) project images into hundreds to thousands of visual tokens, making decoder inference expensive in both attention computation and KV-cache memory. Many decoder-side visual-token pruning methods follow a *rank-and-remove* paradigm: they score visual tokens, keep a compact subset, and permanently discard the rest. We show that this irreversible action can be fragile because visual-token importance changes across decoder depth; tokens ranked low at one stage may become relevant in later layers, especially for grounding-sensitive queries. We propose **Reroute**, a training-free, scorer-preserving plug-in that replaces removal with recoverable routing. At each routing stage, selected visual tokens pass through decoder blocks, while deferred tokens bypass the stage and re-enter the candidate pool at the next routing decision. By preserving the underlying ranking signal and modifying only the post-selection action, Reroute enables a controlled study of irreversible visual-token pruning. Reroute inherits existing attention-score ranking rules within a stage-wise routing framework, largely preserving the TFLOPs and KV-cache efficiency of the pruning method it augments while retaining most of the corresponding speedup. Across LLaVA-1.5 and Qwen backbones, Reroute improves matched-budget grounding performance across decoder-side visual-token pruning settings while maintaining general VQA performance under aggressive token reduction. These results suggest that VLM token reduction should not be viewed only as irreversible pruning, but also as recoverable routing.

1 INTRODUCTION

Vision-language models (VLMs) (Liu et al., 2023a; Bai et al., 2023) project images into hundreds to thousands of tokens that are fed to a language-model decoder, making inference cost grow quadratically with attention and linearly with KV-cache. Reducing the number of visual tokens that participate in decoder computation has therefore become a key lever for deploying VLMs at scale (Chen et al., 2024b; Xing et al., 2024; Huang et al., 2026).

Existing methods follow a common *rank-and-remove* recipe. Vision-encoder-side approaches (Yang et al., 2025a; Shang et al., 2025) compress tokens before they enter the decoder. Decoder-side methods intervene inside the LLM: FastV (Chen et al., 2024b) makes a single-shot pruning decision at an early layer using text-to-vision attention; PDrop (Xing et al., 2024) generalizes this to a progressive multi-stage schedule; FEATHER (Endo et al., 2025) and Nüwa (Huang et al., 2026) refine the scoring rule to better preserve spatial evidence. Across these methods, the underlying *action* is identical. Once a visual token is deemed unimportant, it is removed from the sequence and cannot contribute to any later layer.

This paradigm assumes that pruning decisions made at one layer remain valid at greater decoder depth. Our analysis shows that this assumption does not always hold. Figure 2(a) tracks attention from query words to visual tokens across depth: shallow layers attend diffusely, while target-relevant regions emerge only in middle and deep layers. Figure 2(b) follows one ground-truth token for "man in green shirt": its attention percentile is **0.11 at layer 3** (where FastV decides) and **0.25 at layer 8** (where PDrop decides). Both methods irreversibly remove it, yet it climbs to **0.97 at layer 25**. As Figure 1 shows, this irrecoverability causes outright grounding collapse under 88.9% token reduction across three backbones, with IoU falling below 0.4.

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.

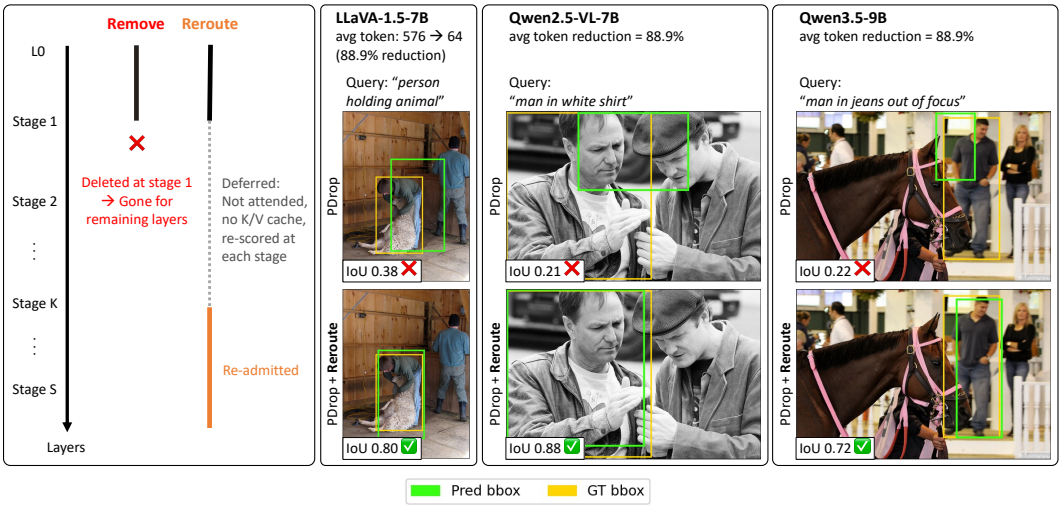


Figure 1: **Reroute, don't remove.** Conventional pruning irreversibly deletes dropped visual tokens, whereas Reroute defers them from computation and re-evaluates them at later routing stages for possible re-entry. Under aggressive 88.9% visual token reduction (576→64 tokens for LLaVA-1.5), irreversible pruning can discard evidence needed for spatial grounding. Using the same PDrop scorer and token budget, Reroute re-admits useful visual evidence and substantially improves grounding across three distinct MLLM designs: LLaVA-1.5-7B, Qwen2.5-VL-7B with native dynamic resolution, and Qwen3.5-9B with hybrid linear and gated attention. Reroute is training-free and retains comparable inference efficiency.

We argue the limitation lies not only in how visual tokens are scored, but also in the action taken after scoring. We propose **Reroute**, a training-free plug-in that replaces irreversible removal with recoverable routing (Figure 3). At each routing layer, Reroute reuses text-to-vision attention as the ranking signal and selects a top-$r\%$ fraction of tokens to pass through Attn+FFN. Deferred tokens are *not* deleted. They bypass the stage via a residual path and re-enter the candidate pool at the next routing layer, evaluated against the *full* visual-token set. Since only selected tokens enter Attn+FFN per stage, Reroute largely preserves the FLOPs and KV-cache efficiency of the pruning baseline it augments, while retaining most of the corresponding prefill speedup. From a conditional-computation view, Reroute is a training-free, attention-driven instantiation of Mixture-of-Depths (Raposo et al., 2024) for VLM visual tokens, in contrast to learned-router approaches like γ-MoD (Luo et al., 2024). We validate Reroute across decoder-side visual-token pruning settings on LLaVA-1.5-7B, Qwen2.5-VL-7B, and Qwen3.5-9B. Across these backbones, Reroute yields grounding improvements at matched token budgets, including under aggressive token reduction.

Our contributions are as follows:

- **Training-free plug-in with comparable efficiency.** Reroute augments existing decoder-side visual-token pruning with a scorer-preserving post-selection transformation, requiring no retraining while largely preserving the efficiency of the corresponding pruning baselines.
- **Controlled study of irreversibility.** By keeping the underlying ranking signal unchanged and replacing permanent removal with deferred-token bypass, Reroute enables a controlled study of the effect of irreversible visual-token pruning across decoder depth.
- **Substantial gains across backbones.** Across three evaluated backbones, Reroute improves visual grounding over matched-budget pruning baselines while preserving general multimodal performance, including under aggressive token reduction.

2 Related Work

Vision-encoder-side visual token reduction. ViT pruning supplies the primitives: score-and-drop (Rao et al., 2021; Liang et al., 2022), merging (Bolya et al., 2022), token halting (Yin et al., 2022), and slow-fast evolution that keeps uninformative tokens as cheaply updated placeholders (Xu et al., 2022). VLMs apply these on the encoder or projector (Shang et al., 2025; Yang et al., 2025a; Zhang et al., 2025c; Arif et al., 2025), with instruction-, query-, or alignment-guided variants (Huang et al.,

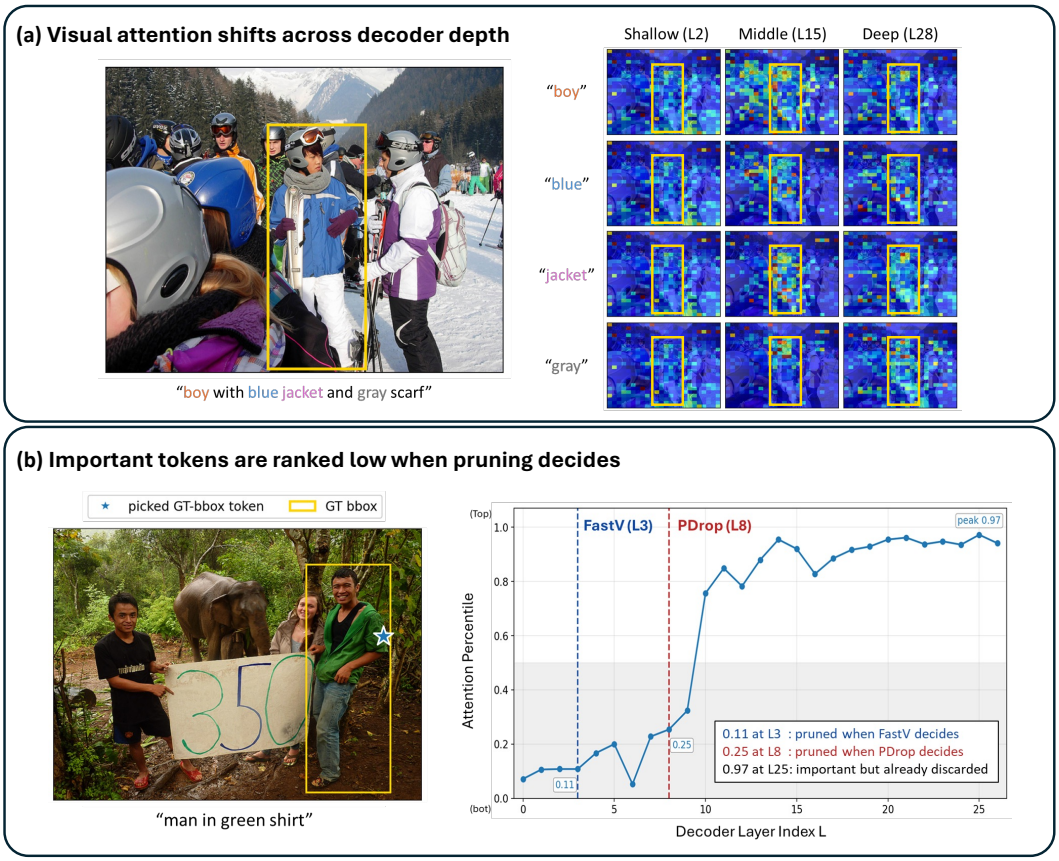


Figure 2: **Why reroute? Important visual tokens are ranked low when pruning decides.** (a) On a single RefCOCO image, attention from each query word ("boy", "blue", "jacket", "gray") to visual tokens drifts substantially across decoder depth. Shallow layers attend diffusely, while target-relevant regions only emerge at middle and deep layers. (b) Tracking one ground-truth token ($\star$) inside the GT bbox of "man in green shirt": its attention percentile is 0.11 at L3 (where FastV drops) and 0.25 at L8 (where PDrop drops), so both methods irreversibly remove it. Yet its attention percentile reaches 0.97 by L25; a token overlooked when pruning commits becomes highly ranked at depth. Reroute keeps such tokens recoverable.

2024; Gao et al., 2026; Cao et al., 2024); others redesign the projector itself (Li et al., 2025b; Cha et al., 2024; Hu et al., 2024b; Cai et al., 2024a; Ye et al., 2025b; Chen et al., 2024a; Wang et al., 2025), prune patches by saliency (Mahmud et al., 2024), or compress both encoder- and decoder-side representations (Hu et al., 2024a; Huang et al., 2026). Reroute instead changes whether tokens excluded from decoder computation remain eligible for later selection.

Decoder-side visual token pruning. FastV prunes once with a text-to-vision attention rule (Chen et al., 2024b); PyramidDrop makes the schedule progressive (Xing et al., 2024), FitPrune fits it from attention statistics (Ye et al., 2025a), and Dynamic-LLaVA learns predictors for prefill and decoding (Huang et al., 2025b). Later work refines the scorer with spatial, representation-aware, or budget-adaptive selection (Endo et al., 2025; Li et al., 2026a; Chen et al., 2026; Ji et al., 2026), or with conditional diversity, duplication, entropy, graph propagation, sink-aware saliency, and other training-free criteria (Zhang et al., 2026; Wen et al., 2025b; Wang et al., 2026; Jiang et al., 2025; Li et al., 2026b; 2025a; Zhang et al., 2025b; Luan et al., 2025), motivated by findings that attention-based ranking often fails to beat random or pooling baselines (Wen et al., 2025a), is distorted by visual attention sinks (Kang et al., 2025), and that pruning can collapse visual grounding (Chien et al., 2025). Others mitigate the loss after selection: recycling discarded tokens (Zhang et al., 2024; Yang et al., 2025b), transferring their information before removal (Sun et al., 2026), text-guided recovery (Chen et al., 2025), or re-requesting a higher-resolution image (Yang et al., 2026). Closest to us, concurrent work reactivates pruned tokens whose attention trend rises (Ma et al., 2026a). Reroute is

scorer-preserving and needs no trend model or recovery module: deferred tokens keep their states on the residual path and their eligibility, so the inherited rule can re-admit them at every stage.

Conditional computation and visual-token skipping. Conditional computation varies depth per input or token (Graves, 2016; Teerapittayanon et al., 2016; Huang et al., 2017; Xin et al., 2020; Liu et al., 2020; Schuster et al., 2022). Token-level variants skip layers via the residual path: Token Dropping re-attaches dropped tokens only at the last layer (Hou et al., 2022), Transkimmer forwards skimmed tokens straight to the output (Guan et al., 2022), Random-LTD drops a random subset per layer during training (Yao et al., 2022), and CoLT5 routes tokens to light or heavy branches (Ainslie et al., 2023). Mixture-of-Depths routes tokens per block with a learned router (Raposo et al., 2024), obtainable by router-only tuning (He et al., 2025) and extended to MLLMs (Luo et al., 2024; Zhang et al., 2025a). For visual tokens, bypass replaces removal: operation pruning (GSOP/DOP) searches an offline, input-agnostic policy that skips attention or FFN for token groups, deeper layers first (Liu et al., 2025); ViCA retrains so that all visual tokens bypass self-attention and FFN, interacting only via sparse cross-attention (Liu et al., 2026); Skip-Vision skips FFN (Zeng et al., 2025), ShortV freezes visual updates in low-contribution layers (Yuan et al., 2025), DyVTE exits all visual tokens at a predicted layer (Wu et al., 2024), HiDrop combines late injection, progressive pruning, and early exit (Wu et al., 2026), and concurrent operator-level methods bypass attention (Ma et al., 2026b), or attention, FFN, or both (Luo et al., 2026). Reroute needs neither a learned router nor offline search: it reuses the pruning scorer, decides per token, and makes the decision revocable by re-scoring the full candidate pool at every routing stage.

Efficient inference and KV-cache compression. Reroute is complementary to KV-cache eviction (Xiao et al., 2023; Zhang et al., 2023; Liu et al., 2023b; Ge et al., 2023; Li et al., 2024; Cai et al., 2024b), its multimodal extensions (Wan et al., 2024; Liu et al., 2024d; Tu et al., 2025; Huang et al., 2025a), KV quantization (Su et al., 2025), and system-level optimizations (Dao et al., 2022; Kwon et al., 2023): those decide which past states to keep; we decide which visual tokens deserve full computation, and the two can be combined.

3 PRELIMINARIES

Multimodal decoder notation. Let $\mathcal{T}, \mathcal{V}$ index text and vision token positions. At layer ℓ, hidden states $H^\ell = \{h_j^\ell\}_{j \in \mathcal{T} \cup \mathcal{V}}$ are updated by a standard decoder block:

$$H^{\ell+1} = \tilde{H}^{\ell+1} + \mathrm{FFN}_\ell(\mathrm{LN}_2(\tilde{H}^{\ell+1})), \quad \tilde{H}^{\ell+1} = H^\ell + \mathrm{Attn}_\ell(\mathrm{LN}_1(H^\ell)). \tag{1}$$

Following the visual-token pruning literature, reduction targets only vision tokens; text tokens stay active throughout decoding, isolating the post-ranking action on low-ranked vision tokens.

Stage-wise visual-token selection. We cast decoder-side visual-token reduction as stage-wise selection. The decoder is partitioned into S stages, with stage i starting at routing layer ℓ_i with keep ratio $r_i\%$. Let $\mathcal{C}_i \subseteq \mathcal{V}$ denote the candidate vision-token set at the start of stage i, with $\mathcal{C}_1 = \mathcal{V}$. At layer ℓ_i, a scorer (typically text-to-vision attention) ranks $\mathcal{C}_i$ and picks $\mathcal{V}_i^{\mathrm{sel}} = \text{Top-K}_i(\mathcal{C}_i)$ with $K_i = \lfloor |V| \cdot r_i\% \rfloor$; the deferred set is $\mathcal{V}_i^{\mathrm{def}} = \mathcal{C}_i \setminus \mathcal{V}_i^{\mathrm{sel}}$. Since text tokens remain active in every stage, the active set is $\mathcal{A}_i = \mathcal{T} \cup \mathcal{V}_i^{\mathrm{sel}}$. The reduction method is defined by the action on $\mathcal{V}_i^{\mathrm{def}}$: pruning removes them, whereas Reroute keeps them recoverable.

Rank-and-remove as irreversible pruning. Conventional pruning removes $\mathcal{V}_i^{\mathrm{def}}$, contracting the candidate pool to $\mathcal{C}_{i+1}^{\mathrm{prune}} = \mathcal{V}_i^{\mathrm{sel}}$. Once deferred, a vision token cannot be re-scored or re-selected in later stages. This irreversibility defines rank-and-remove pruning.

4 METHOD

4.1 REROUTE: RECOVERABLE STAGE-WISE ROUTING

Reroute keeps the stage-wise ranking interface of pruning but replaces the post-selection action. At routing layer ℓ_i, the inherited scorer and the prescribed keep ratio yield $\mathcal{V}_i^{\mathrm{sel}}$ and $\mathcal{V}_i^{\mathrm{def}}$: selected tokens traverse the full decoder path, while deferred tokens bypass the current stage yet remain eligible for later selection. As summarized in Figure 3, each stage ranks vision tokens via text-to-vision attention, routes $\mathcal{V}_i^{\mathrm{sel}}$ through Attn+FFN, and defers the rest until the next routing decision.

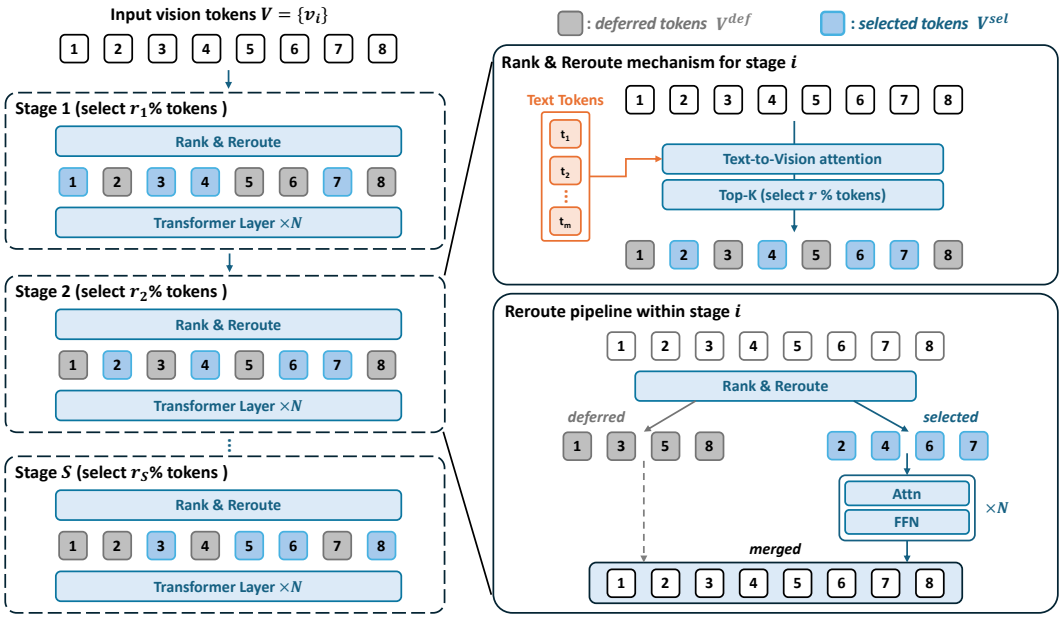


Figure 3: **Overview of Reroute.** Reroute organizes decoder-side visual token reduction into S routing stages, each consisting of a `Rank & Reroute` operator followed by N transformer layers. *(Left)*: Input visual tokens $V = \{v_i\}$ pass through stages $1...S$; at each stage i, the operator selects a top-r_i% subset to route through the transformer block, while the remaining *deferred* tokens bypass it via a residual path. Crucially, deferred tokens are not removed from V. They re-enter the next stage's ranking and may be reselected. *(Right)*: Within one stage, selected and deferred tokens take separate computational paths and merge back at the stage output, restoring the original sequence ordering for the next stage. The `Rank & Reroute` operator itself reuses the existing text-to-vision attention from the current decoder layer as the ranking signal. No additional router is trained. The top-K visual tokens ($K = \lfloor |V| \cdot r_i\% \rfloor$) are marked *selected*; the rest are marked *deferred*. Reroute largely preserves the FLOPs and KV-cache efficiency of conventional pruning (only K visual tokens enter Attn + FFN), while breaking the irreversibility constraint that pruning imposes on V.

Let Block_ℓ denote the full decoder block at layer ℓ (self-attention, FFN, residuals, and normalization). Reroute applies it only to the active subsequence $H^\ell_{\mathcal{A}_i}$, scatters outputs to their original positions, and leaves deferred tokens untouched:

$$h_j^{\ell+1} = \begin{cases} \mathrm{Block}_\ell(H^\ell_{\mathcal{A}_i})_j, & j \in \mathcal{A}_i, \\ h_j^\ell, & j \in \mathcal{V}_i^{\mathrm{def}}. \end{cases}$$

At the next routing layer, the full candidate set is restored by $\mathcal{C}_{i+1}^{\mathrm{reroute}} = \mathcal{V}$, so a token deferred in one stage may be re-selected later if its score rises. Reroute thus converts irreversible deletion into recoverable deferral.

4.2 Pruning as a Degenerate Routing Policy

Under a fixed scorer, routing layout, and per-stage token budget, the two policies differ in the candidate-set transition:

$$\mathcal{C}_{i+1}^{\mathrm{prune}} = \mathcal{V}_i^{\mathrm{sel}} \quad \text{and} \quad \mathcal{C}_{i+1}^{\mathrm{reroute}} = \mathcal{V}. \tag{2}$$

Sharing the same scoring rule, routing layers, and stage budgets, pruning is the special case that permanently excludes deferred tokens from subsequent candidate sets, whereas Reroute keeps them valid for future stages. Figure 4 illustrates this: prior to an additional reroute decision the two policies can produce identical selections, but later stages may re-admit tokens that pruning has already discarded.

4.3 Matched-Budget Efficiency

Reroute is designed for budget-matched comparison with the pruning baseline it augments. Under a shared schedule $\{(\ell_i, r_i)\}_{i=1}^S$, both methods activate the same number of vision tokens per stage.

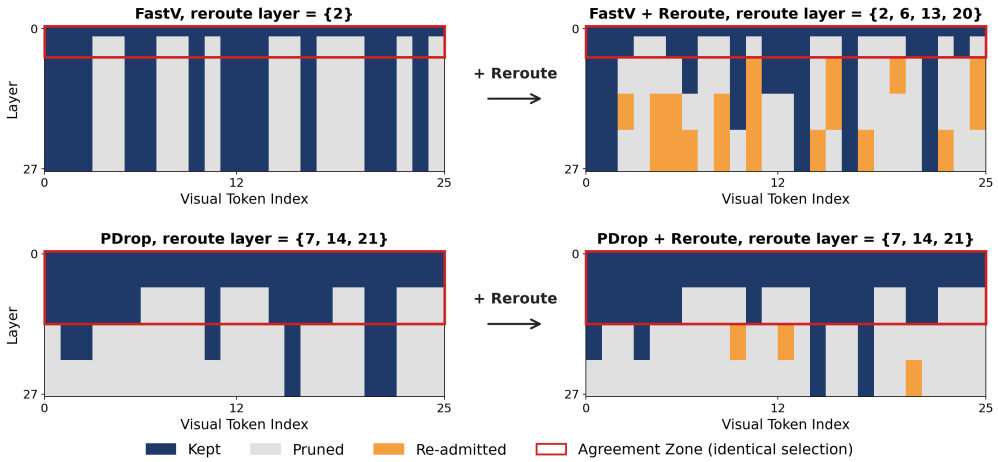


Figure 4: **Pruning as a degenerate routing policy.** We compare FastV and PDrop with their Reroute variants on Qwen2.5-VL-7B. Blue, gray, and orange denote selected, pruned/deferred, and re-admitted tokens. The red box marks the shared selected-token prefix before pruning and reroute diverge: pruning keeps only the previous candidate set, whereas reroute restores the full visual-token candidate set. For compact visualization, we show only the leading portion of the visual-token sequence.

Token identities may diverge once re-entry is allowed. The active set is $\mathcal{A}_i = \mathcal{T} \cup \mathcal{V}_i^{\text{sel}}$. With an optimized implementation, only $\mathcal{A}_i$ executes the decoder block at stage i, so Reroute targets the same theoretical TFLOPs and KV-cache class as the corresponding pruning schedule. The sole distinction is the post-selection action: pruning discards $\mathcal{V}_i^{\text{def}}$, while Reroute retains these tokens in the sequence state for future routing. Practical latency still depends on gather/scatter kernels and KV-cache management.

4.4 Relation to Conditional Computation

Reroute can be viewed as a training-free application of Mixture-of-Depths (MoD)-style conditional computation to VLM visual-token pruning. Whereas standard MoD methods learn a router that decides which tokens execute a block and which bypass it, Reroute reuses the same conditional-computation pattern but inherits its routing signal, without training, from the text-to-vision attention scores already used by pruning.

This framing clarifies the design motivation. Visual-token pruning already poses a token-wise compute-allocation problem: selected tokens receive decoder computation while low-ranked ones are excluded. Rank-and-remove resolves the allocation by deletion. Reroute instead treats the excluded tokens as a bypass branch. They skip the stage but remain eligible for later selection. Reroute is therefore not a new general-purpose MoD architecture, but a training-free instantiation of the MoD principle in the MLLM visual-token pruning setting, with irreversible removal replaced by recoverable deferral.

5 Experiments

We evaluate Reroute under matched visual-token budgets on grounding, general VQA, cross-backbone transfer, and efficiency.

5.1 Experimental Setup

Benchmarks. We evaluate grounding on RefCOCO, RefCOCO+, and RefCOCOg (Kazemzadeh et al., 2014; Yu et al., 2016) using bounding-box accuracy at IoU 0.5 (Acc@0.5). We also evaluate general multimodal performance on standard benchmarks, including GQA (Hudson & Manning, 2019), TextVQA (Singh et al., 2019), and MME (Fu et al., 2023), with benchmark coverage determined by model support. **Models.** We evaluate LLaVA-1.5-7B (Liu et al., 2024a), Qwen2.5-VL-7B (Bai et al., 2025b), and Qwen3.5-9B-Hybrid (Qwen Team, 2026). LLaVA-1.5-7B enables direct comparison with

Table 1: **Visual grounding performance on RefCOCO-series benchmarks using LLaVA-1.5-7B.** We compare visual-token reduction methods and their Reroute variants under matched average-token budgets. Panel (a) reports the original LLaVA-format results, with * marking Nüwa results reproduced in the official Nüwa codebase; panel (b) reports HuggingFace-format reproduced results under the same LLaVA-1.5-7B backbone. When shown, A/B denote testA/testB splits. Avg. Ratio is normalized by the corresponding full-token baseline within each panel. Best and second-best results within each average-token block are shown in **bold** and underlined, respectively, and unavailable results are denoted by "–".

(a) Original LLaVA format results.

Method	Src.	RefCOCO		RefCOCO+			RefCOCOg		Avg.
		val	test	val	A	B	val	test	Ratio
Average Token 576									
Vanilla*	CVPR'24	**56.2**	**58.0**	**50.0**	**59.3**	**38.8**	**48.9**	**48.4**	**100.0%**
Average Token 192 ↓ 66.7%									
FEATHER	ICCV'25	–	27.7	–	24.7	–	–	27.2	48.5%
Nüwa*	ICLR'26	**59.1**	**60.1**	**50.5**	60.9	**40.2**	**51.8**	**50.1**	**103.7%**
Nüwa+Reroute*	Ours	58.6	59.6	50.4	**61.3**	39.8	50.9	49.9	103.0%
Average Token 128 ↓ 77.8%									
FastV	ECCV'24	10.1	10.3	8.2	8.5	9.8	9.1	8.9	18.4%
SparseVLM	ICML'25	6.2	6.3	9.9	5.8	4.2	6.5	6.4	12.6%
VisionZip	CVPR'25	4.1	4.5	3.9	4.1	4.9	3.5	3.5	8.1%
Nüwa*	ICLR'26	**52.3**	**53.5**	43.7	53.2	**35.6**	**45.0**	**43.5**	**90.9%**
Nüwa+Reroute*	Ours	51.8	53.0	**44.3**	**55.5**	35.0	44.3	43.1	90.8%
Average Token 64 ↓ 88.9%									
FastV	ECCV'24	2.0	2.7	2.4	1.2	1.0	2.0	2.2	3.8%
SparseVLM	ICML'25	1.0	1.0	1.0	1.0	1.3	0.7	0.6	1.9%
VisionZip	CVPR'25	3.8	4.0	3.5	3.7	3.9	3.2	3.4	7.2%
Nüwa*	ICLR'26	**32.8**	33.2	26.5	34.3	20.5	25.6	24.5	54.6%
Nüwa+Reroute*	Ours	**32.8**	**33.7**	**27.2**	**35.3**	**21.1**	**26.3**	**25.2**	**55.8%**

(b) HuggingFace-format reproduced results.

Method	Src.	RefCOCO			RefCOCO+			RefCOCOg		Avg.
		val	A	B	val	A	B	val	test	Ratio
Average Token 576										
Vanilla (HF)	CVPR'24	**54.0**	**59.4**	**48.0**	**46.5**	**54.1**	**39.1**	**46.9**	**47.6**	**100.0%**
Average Token 192 ↓ 66.7%										
FastV	ECCV'24	25.3	27.8	22.3	21.0	23.6	18.0	21.5	20.5	45.5%
FastV+Reroute	Ours	32.7	37.7	26.8	26.7	32.3	21.3	28.7	29.1	59.2%
PDrop	CVPR'25	39.8	45.1	33.4	33.5	39.5	25.9	33.2	32.8	71.3%
PDrop+Reroute	Ours	**45.3**	**49.5**	**39.0**	**37.9**	**44.3**	**31.1**	**39.7**	**39.2**	**82.3%**
Average Token 128 ↓ 77.8%										
FastV	ECCV'24	11.0	12.7	9.1	8.9	10.3	8.0	9.3	9.0	19.8%
FastV+Reroute	Ours	17.4	20.7	13.9	14.5	17.5	11.6	15.8	15.1	31.8%
PDrop	CVPR'25	30.7	36.5	23.7	25.2	30.5	18.3	24.4	23.9	53.4%
PDrop+Reroute	Ours	**34.1**	**39.6**	**27.2**	**27.6**	**33.2**	**22.0**	**29.8**	**29.7**	**61.2%**
Average Token 64 ↓ 88.9%										
FastV	ECCV'24	0.3	0.2	0.6	0.3	0.1	0.5	0.1	0.1	0.6%
FastV+Reroute	Ours	0.4	0.2	0.4	0.3	0.1	0.7	0.1	0.1	0.6%
PDrop	CVPR'25	12.8	14.8	10.2	10.1	12.1	8.0	10.3	10.1	22.2%
PDrop+Reroute	Ours	**18.9**	**23.3**	**14.5**	**15.4**	**19.9**	**11.5**	**16.5**	**15.6**	**34.0%**

Vanilla baseline Other methods Base methods Reroute variants.

existing visual-token pruning methods, while the Qwen-family models evaluate cross-backbone transfer. **Baselines.** We compare against the full-token baseline, FastV (Chen et al., 2024b), PDrop (Xing et al., 2024), and the corresponding budget-matched Reroute variants. We further evaluate Reroute on Nüwa (Huang et al., 2026) by augmenting the FastV-style decoder pruning in the official implementation. For each backbone and pruning method, we use the same routing schedule across benchmarks; detailed routing configurations are provided in Appendix Table 6. **Efficiency.** TFLOPs and KV-cache memory are obtained from runtime profiling, while wall-clock prefill latency is measured on a single NVIDIA RTX 4090 GPU with batch size 1.

5.2 Main results

Visual grounding on LLaVA-1.5. Table 1 reports RefCOCO-series results on LLaVA-1.5 under two checkpoint formats. Panel (a) follows the original LLaVA-1.5 format, combining paper-reported baselines with rows reproduced in the official Nüwa codebase (*). Panel (b) uses the HuggingFace-format checkpoint and reports fully reproduced results from our unified pipeline. Results are grouped by average visual-token budget, and average ratios are normalized by the corresponding full-token baseline.

In the HuggingFace-format setting, Reroute provides clear improvements over both matched pruning baselines. At 192 average tokens, it raises the average ratio from 45.5% to 59.2% for FastV and from 71.3% to 82.3% for PDrop. At 128 tokens, the gains remain 12.0 and 7.8 points, respectively. At the extreme 64-token budget, one-shot FastV leaves little room for recovery, whereas PDrop+Reroute still improves from 22.2% to 34.0%, indicating that repeated re-entry opportunities are important for effective rerouting.

In the original-format setting, Nüwa provides a stronger comparison through a two-stage reduction pipeline that combines spatially preserving pre-decoder aggregation with text-guided pruning inside the decoder. We apply Reroute to the second-stage decoder pruning while leaving the spatially preserving front end unchanged. Nüwa+Reroute remains comparable to Nüwa at the 192- and 128-token budgets and further improves the average ratio from 54.6% to 55.8% at 64 tokens. This result shows that Reroute can provide additional benefit even when the underlying pipeline already incorporates explicit spatial preservation.

Table 2: **RefCOCO-series visual grounding with Qwen-family VLMs.** FastV, PDrop, and their Reroute variants are compared under matched average-token budgets. Panel (a) uses Qwen2.5-VL-7B with decoder-side retained visual tokens; panel (b) uses Qwen3.5-9B-Hybrid with gated-attention routed visual tokens. A/B denote testA/testB. Avg. Ratio is normalized to the corresponding vanilla baseline. Best and second-best results within each budget block are bolded and underlined.

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

Method	Src.	RefCOCO			RefCOCO+			RefCOCOg		Avg.
		val	A	B	val	A	B	val	test	Ratio
Average Token 100%										
Vanilla	Qwen'25	**87.7**	**89.8**	**82.3**	**80.5**	**85.9**	**72.8**	**82.7**	**83.3**	**100.0%**
Average Token Reduction ↓ 66.7%										
FastV	ECCV'24	46.4	48.1	44.4	40.5	42.6	39.3	48.4	47.2	53.7%
FastV+Reroute	Ours	53.4	57.8	49.4	45.0	50.9	40.6	52.8	52.1	60.3%
PDrop	CVPR'25	55.9	59.2	52.0	48.4	53.0	43.6	53.6	52.9	62.9%
PDrop+Reroute	Ours	**57.9**	**63.0**	**53.6**	**49.9**	**55.7**	**44.8**	**54.7**	**54.0**	**65.1%**
Average Token Reduction ↓ 77.8%										
FastV	ECCV'24	19.2	21.5	17.8	16.7	18.4	14.6	19.1	18.6	21.9%
FastV+Reroute	Ours	26.8	31.1	22.9	21.7	26.0	18.0	25.3	24.1	29.3%
PDrop	CVPR'25	25.9	30.9	24.2	21.2	26.7	19.2	24.1	23.7	29.3%
PDrop+Reroute	Ours	**33.2**	**37.6**	**31.5**	**27.7**	**33.5**	**25.1**	**31.0**	**29.7**	**37.4%**
Average Token Reduction ↓ 88.9%										
FastV	ECCV'24	1.6	1.6	2.2	1.7	1.7	2.6	2.0	2.3	2.4%
FastV+Reroute	Ours	6.2	5.5	5.9	4.8	4.2	4.8	6.7	6.5	6.7%
PDrop	CVPR'25	10.2	10.9	9.0	7.9	9.8	6.7	9.9	9.5	11.1%
PDrop+Reroute	Ours	**16.8**	**19.2**	**15.2**	**13.9**	**16.9**	**12.2**	**15.0**	**14.9**	**18.6%**

(b) Qwen3.5-9B-Hybrid.

Method	Src.	RefCOCO			RefCOCO+			RefCOCOg		Avg.
		val	A	B	val	A	B	val	test	Ratio
Average Gated Attention Token 100%										
Vanilla	Qwen'26	**90.4**	**90.7**	**85.5**	**80.0**	**84.3**	**75.7**	**87.3**	**86.8**	**100.0%**
Average Gated Attention Token Reduction ↓ 66.7%										
FastV	ECCV'24	28.7	32.5	25.6	25.2	28.3	20.3	26.3	25.6	31.1%
FastV+Reroute	Ours	**70.8**	75.1	**64.8**	60.7	67.0	53.9	68.5	68.0	77.5%
PDrop	CVPR'25	36.4	40.4	32.4	32.7	36.1	27.4	33.7	33.4	40.0%
PDrop+Reroute	Ours	70.0	**76.1**	62.4	**61.1**	**68.6**	**54.2**	**69.2**	**68.3**	**77.7%**
Average Gated Attention Token Reduction ↓ 77.8%										
FastV	ECCV'24	18.0	22.2	15.9	14.5	18.3	12.3	16.6	15.9	19.5%
FastV+Reroute	Ours	51.7	58.4	44.6	42.3	49.7	35.7	50.2	49.6	55.9%
PDrop	CVPR'25	23.9	28.6	20.4	20.4	25.0	16.7	20.8	20.4	25.8%
PDrop+Reroute	Ours	**54.9**	**62.7**	**47.0**	**46.4**	**53.5**	**38.0**	**52.9**	**52.2**	**59.7%**
Average Gated Attention Token Reduction ↓ 88.9%										
FastV	ECCV'24	12.4	14.7	12.3	6.8	6.8	5.8	8.3	8.1	10.9%
FastV+Reroute	Ours	14.2	15.0	13.2	8.8	9.4	7.6	11.9	11.6	13.4%
PDrop	CVPR'25	19.5	23.8	15.7	14.2	17.8	10.0	15.8	15.5	19.3%
PDrop+Reroute	Ours	**35.0**	**41.1**	**27.8**	**27.8**	**34.8**	**22.2**	**33.7**	**33.8**	**37.4%**

Vanilla baseline Base methods Reroute variants.

Table 3: **VQA performance on LLaVA-1.5-7B under the HuggingFace model format.** Bold and underline denote best and second-best results within each average-token block.

Method	GQA	MMB	MMMU	MME	TextVQA	POPE	SQA	SEED	Avg. Ratio
Average Token 576									
Vanilla	**60.5**	**62.4**	**34.4**	**1782**	**57.5**	**85.9**	**66.1**	**65.5**	**100.0%**
Average Token 192 ↓ 66.7%									
FastV	55.0	58.6	34.2	1637	53.7	79.6	66.4	58.2	93.9%
FastV+Reroute	57.4	60.7	34.2	1672	56.7	81.0	**67.1**	60.4	96.5%
PDrop	**59.2**	**61.4**	**34.7**	**1776**	**57.5**	**84.4**	65.9	**63.4**	**98.9%**
PDrop+Reroute	58.9	61.3	34.4	1775	57.4	84.2	66.1	62.6	98.6%
Average Token 128 ↓ 77.8%									
FastV	50.5	53.7	34.0	1451	50.4	73.0	65.5	54.3	88.0%
FastV+Reroute	54.8	57.0	34.3	1557	53.6	75.1	**68.4**	55.4	92.2%
PDrop	57.0	59.5	34.2	1728	56.2	**82.1**	66.8	59.2	96.3%
PDrop+Reroute	**57.3**	**59.8**	**34.6**	**1733**	**56.5**	80.9	66.6	**60.8**	**96.8%**
Average Token 64 ↓ 88.9%									
FastV	40.0	22.4	32.1	950	41.8	51.1	65.1	36.0	66.8%
FastV+Reroute	44.3	23.3	32.2	954	43.5	51.4	65.3	39.8	69.1%
PDrop	52.8	55.1	33.1	1486	51.6	**74.9**	68.2	53.9	89.7%
PDrop+Reroute	**54.4**	**57.4**	**33.8**	**1542**	**52.5**	74.8	**69.2**	**54.8**	**91.7%**

Vanilla baseline Base methods Reroute variants

Table 4: **Efficiency comparison on LLaVA-1.5-7B.** TFLOPs, KV-cache memory, and measured prefill latency are reported relative to the vanilla full-token baseline.

Method	Avg.	TFLOPs		KV Cache		Prefill	
	Tok.	Val.	/Van.	MB	/Van.	ms	/Van.
Vanilla	576	8.67	–	297.5	–	69.73	–
FastV	192	3.32	↓ 62%	105.4	↓ 65%	37.21	↓ 47%
FastV+Reroute	192	3.45	↓ 60%	105.4	↓ 65%	40.38	↓ 42%
PDrop	192	3.35	↓ 61%	105.2	↓ 65%	37.57	↓ 46%
PDrop+Reroute	192	3.42	↓ 61%	105.2	↓ 65%	40.36	↓ 42%
FastV	128	2.45	↓ 72%	73.2	↓ 75%	31.58	↓ 55%
FastV+Reroute	128	2.58	↓ 70%	73.2	↓ 75%	34.90	↓ 50%
PDrop	128	2.47	↓ 72%	73.3	↓ 75%	34.42	↓ 51%
PDrop+Reroute	128	2.56	↓ 70%	73.2	↓ 75%	35.95	↓ 48%
FastV	64	1.60	↓ 82%	41.5	↓ 86%	29.82	↓ 57%
FastV+Reroute	64	1.73	↓ 80%	41.5	↓ 86%	33.49	↓ 52%
PDrop	64	1.83	↓ 79%	49.7	↓ 83%	31.90	↓ 54%
PDrop+Reroute	64	1.98	↓ 77%	51.6	↓ 83%	32.93	↓ 53%

Vanilla baseline Base methods Reroute variants

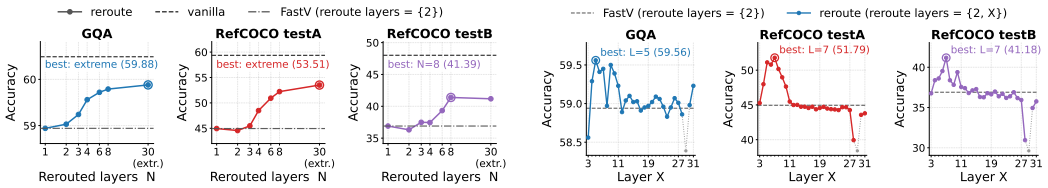
These results show that recoverable deferral substantially improves matched-budget grounding for decoder-side pruning and provides additional gains under aggressive reduction when integrated into Nüwa's stronger two-stage pipeline.

General VQA results on LLaVA-1.5. Table 3 reports general VQA results on the HuggingFace-format LLaVA-1.5 checkpoint. These benchmarks serve as an auxiliary evaluation: our main analysis focuses on recoverable visual evidence for grounding under aggressive token reduction, while VQA evaluates whether the same routing mechanism preserves general multimodal capability.

Across the 192-, 128-, and 64-token budgets, Reroute remains comparable to or better than its matched FastV and PDrop counterparts. These results show that the grounding improvements reported above are achieved without sacrificing general VQA performance, indicating that Reroute preserves general multimodal capability under matched token budgets.

Qwen-series visual grounding results. We next evaluate Reroute beyond LLaVA-1.5 on Qwen2.5-VL-7B and Qwen3.5-9B-Hybrid. Table 2 reports RefCOCO-series results under matched visual-token reduction budgets.

On Qwen2.5-VL, Reroute improves both FastV and PDrop across all evaluated budgets. At 66.7% reduction, FastV+Reroute increases the average ratio from 53.7% to 60.3%, while PDrop+Reroute improves from 62.9% to 65.1%. The advantage is retained under more aggressive reduction, with PDrop+Reroute improving from 29.3% to 37.4% at 77.8% reduction and from 11.1% to 18.6% at 88.9%.



(a) Frequency sweep. (b) Placement sweep.

Figure 5: **Ablation on reroute scheduling.** With the retention ratio fixed at $r_i\% = 0.5$, we vary (a) the number of routing stages N and (b) the placement X of the second routing stage while keeping the first fixed. More frequent reassessment generally improves performance before saturation, while an early-to-middle placement performs best and later re-entry reduces the benefit of rerouting, approaching the single-stage FastV baseline.

Qwen3.5-9B-Hybrid interleaves Gated DeltaNet linear-attention blocks with softmax-attention blocks, with Reroute applied only to the latter. Reroute also provides substantial gains in this hybrid setting: at 66.7% reduction, the average ratio improves from 31.1% to 77.5% for FastV and from 40.0% to 77.7% for PDrop. For PDrop, the corresponding improvements are 25.8% to 59.7% at 77.8% reduction and 19.3% to 37.4% at 88.9%.

These results extend the grounding improvements of Reroute across both standard Transformer and hybrid decoder architectures.

Ablation on reroute scheduling. Figure 5 examines the sensitivity of Reroute to routing frequency and placement at fixed $r_i\% = 0.5$. The single-stage setting recovers FastV, while additional routing stages provide repeated opportunities to reassess deferred tokens.

The frequency sweep shows that additional routing stages generally improve over the single-stage baseline, with diminishing returns as routing becomes denser. Across all three datasets, performance improves with more frequent reassessment of deferred tokens before gradually saturating, suggesting that a moderate number of routing stages is sufficient to capture most of the benefit.

The placement sweep further shows that the timing of reassessment matters. With the first routing stage fixed, placing the second stage in the early-to-middle decoder performs best, whereas later placement reduces the benefit of rerouting and approaches the single-stage FastV baseline.

Efficiency metrics for Reroute. Table 4 summarizes the computational and runtime efficiency of Reroute on LLaVA-1.5 across different visual-token budgets. All percentage reductions in TFLOPs, KV-cache memory, and prefill latency are measured relative to the full-token vanilla baseline.

Under the most aggressive evaluated budget of 64 average visual tokens, FastV+Reroute reduces TFLOPs, KV-cache memory, and prefill latency by 80%, 86%, and 52%, respectively, compared with 82%, 86%, and 57% for FastV. Similarly, PDrop+Reroute achieves reductions of 77%, 83%, and 53%, compared with 79%, 83%, and 54% for PDrop. Comparable trends across the remaining budgets show that Reroute preserves most of the computational and practical efficiency of the corresponding pruning baselines, with only a modest cost for recoverable routing.

6 Conclusion

We revisited visual-token reduction from the perspective of the post-scoring action. Reroute is a training-free, scorer-preserving plug-in that replaces permanent removal with recoverable routing. Selected tokens receive full decoder computation, while deferred tokens bypass the current stage and remain eligible for later reassessment and re-entry. By retaining the underlying ranking rule, Reroute enables a controlled study of irreversible token exclusion under matched token budgets.

Experiments across LLaVA-1.5 and Qwen backbones demonstrate substantial improvements in visual grounding over the corresponding pruning baselines, including under aggressive token reduction, while preserving general VQA performance. Computational profiling and wall-clock measurements further show that Reroute retains most of the FLOPs, KV-cache memory, and prefill-latency savings of pruning. Reroute thus extends existing pruning methods with recoverable routing without retraining, improving grounding while largely preserving inference efficiency.

AI USE STATEMENT

In this work, we used generative AI tools to assist with the implementation and debugging of research code. We have not used generative AI tools to propose or refine the research ideas or hypotheses, design the research methodology or experiments, or interpret the experimental results; the remaining tasks requiring disclosure are not applicable to this work.

Additionally, we used generative AI tools to assist with literature search and summarization, as well as drafting and editing parts of the manuscript for clarity and readability. We reviewed all AI-assisted work. AI-assisted code was reviewed and tested by the authors, literature-related outputs were verified against the original sources, and AI-assisted text was reviewed and revised by the authors. We take responsibility for the final content of this work, including text, claims, code, or other artifacts produced with the aid of generative AI.

REFERENCES

Joshua Ainslie, Tao Lei, Michiel de Jong, Santiago Ontañón, Siddhartha Brahma, Yury Zemlyanskiy, David C Uthus, Mandy Guo, James Lee-Thorp, Yi Tay, et al. Colt5: Faster long-range transformers with conditional computation. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pp. 5085–5100, 2023.

Kazi Hasan Ibn Arif, JinYi Yoon, Dimitrios S Nikolopoulos, Hans Vandierendonck, Deepu John, and Bo Ji. Hired: Attention-guided token dropping for efficient inference of high-resolution vision-language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 1773–1781, 2025.

Natan Bagrov, Eugene Khvedchenia, Borys Tymchenko, Shay Aharon, Lior Kadoch, Tomer Keren, Ofri Masad, Yonatan Geifman, Ran Zilberstein, Tuomas Rintamaki, Matthieu Le, and Andrew Tao. Efficient video sampling: Pruning temporally redundant tokens for faster vlm inference, 2025. URL https://arxiv.org/abs/2510.14624.

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond, 2023. URL https://arxiv.org/abs/2308.12966.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025a.

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, 2025b. URL https://arxiv.org/abs/2502.13923.

Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your vit but faster. *arXiv preprint arXiv:2210.09461*, 2022.

Mu Cai, Jianwei Yang, Jianfeng Gao, and Yong Jae Lee. Matryoshka multimodal models. *arXiv preprint arXiv:2405.17430*, 2024a.

Zefan Cai, Yichi Zhang, Bofei Gao, Yuliang Liu, Yucheng Li, Tianyu Liu, Keming Lu, Wayne Xiong, Yue Dong, Junjie Hu, et al. Pyramidkv: Dynamic kv cache compression based on pyramidal information funneling. *arXiv preprint arXiv:2406.02069*, 2024b.

Jianjian Cao, Peng Ye, Shengze Li, Chong Yu, Yansong Tang, Jiwen Lu, and Tao Chen. Madtp: Multimodal alignment-guided dynamic token pruning for accelerating vision-language transformer. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15710–15719. IEEE, 2024.

Junbum Cha, Wooyoung Kang, Jonghwan Mun, and Byungseok Roh. Honeybee: Locality-enhanced projector for multimodal llm. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13817–13827, 2024.

Jieneng Chen, Luoxin Ye, Ju He, Zhao-Yang Wang, Daniel Khashabi, and Alan Yuille. Efficient large multi-modal models via visual context compression. *Advances in neural information processing systems*, 37:73986–74007, 2024a.

Junjie Chen, Xuyang Liu, Zichen Wen, Yiyu Wang, Siteng Huang, and Honggang Chen. Variation-aware vision token dropping for faster large vision-language models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 3489–3499, 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 *European Conference on Computer Vision*, pp. 19–35. Springer, 2024b.

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

Yi Chen, Jian Xu, Xu-Yao Zhang, Wen-Zhuo Liu, Yang-Yang Liu, and Cheng-Lin Liu. Recoverable compression: A multimodal vision token recovery mechanism guided by text information. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 2293–2301, 2025.

Tzu-Chun Chien, Chieh-Kai Lin, Shiang-Feng Tsai, Ruei-Chi Lai, Hung-Jen Chen, and Min Sun. Grounding-aware token pruning: Recovering from drastic performance drops in visual grounding caused by pruning, 2025. URL https://arxiv.org/abs/2506.21873.

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

Mark Endo, Xiaohan Wang, and Serena Yeung-Levy. Feather the throttle: Revisiting visual token pruning for vision-language model acceleration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 22826–22835, 2025.

Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, et al. Mme: A comprehensive evaluation benchmark for multimodal large language models. *arXiv preprint arXiv:2306.13394*, 2023.

Tianxiao Gao, Shanwei Zhao, Shuo Fang, Shiai Zhu, and Chenguang Ma. Quietprune: Query-guided early token pruning for vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3553–3562, 2026.

Suyu Ge, Yunan Zhang, Liyuan Liu, Minjia Zhang, Jiawei Han, and Jianfeng Gao. Model tells you what to discard: Adaptive kv cache compression for llms. *arXiv preprint arXiv:2310.01801*, 2023.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6904–6913, 2017.

Alex Graves. Adaptive computation time for recurrent neural networks. *arXiv preprint arXiv:1603.08983*, 2016.

Yue Guan, Zhengyi Li, Jingwen Leng, Zhouhan Lin, and Minyi Guo. Transkimmer: Transformer learns to layer-wise skim. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7275–7286, 2022.

Shwai He, Tao Ge, Guoheng Sun, Bowei Tian, Xiaoyang Wang, and Dong Yu. Router-tuning: A simple and effective approach for dynamic depth. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 1925–1938, 2025.

Le Hou, Richard Yuanzhe Pang, Tianyi Zhou, Yuexin Wu, Xinying Song, Xiaodan Song, and Denny Zhou. Token dropping for efficient bert pretraining. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3774–3784, 2022.

Lianyu Hu, Liqing Gao, Fanhua Shang, Liang Wan, and Wei Feng. illava: An image is worth fewer than 1/3 input tokens in large multimodal models. *arXiv preprint arXiv:2412.06263*, 2024a.

Wenbo Hu, Zi-Yi Dou, Liunian H Li, Amita Kamath, Nanyun Peng, and Kai-Wei Chang. Matryoshka query transformer for large vision-language models. *Advances in Neural Information Processing Systems*, 37:50168–50188, 2024b.

Gao Huang, Danlu Chen, Tianhong Li, Felix Wu, Laurens Van Der Maaten, and Kilian Q Weinberger. Multi-scale dense networks for resource efficient image classification. *arXiv preprint arXiv:1703.09844*, 2017.

Kai Huang, Hao Zou, Ye Xi, BoChen Wang, Zhen Xie, and Liang Yu. Ivtp: Instruction-guided visual token pruning for large vision-language models. In *European conference on computer vision*, pp. 214–230. Springer, 2024.

Kai Huang, Hao Zou, Bochen Wang, Ye Xi, Zhen Xie, and Hao Wang. Aircache: Activating inter-modal relevancy kv cache compression for efficient large vision-language model inference. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 23958–23967. IEEE, 2025a.

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. In *International Conference on Learning Representations*, volume 2025, pp. 69927–69955, 2025b.

Yihong Huang, Fei Ma, Yihua Shao, Jingcai Guo, Zitong Yu, Laizhong Cui, and Qi Tian. Nüwa: Mending the spatial integrity torn by vlm token pruning. *arXiv preprint arXiv:2602.02951*, 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.

Huawei Ji, Yuanhao Sun, Yuan Jin, Cheng Deng, Jiaxin Ding, Luoyi Fu, and Xinbing Wang. Vispco: Visual token pruning configuration optimization via budget-aware pareto-frontier learning for vision-language models. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9281–9301, 2026.

Yutao Jiang, Qiong Wu, Wenhao Lin, Wei Yu, and Yiyi Zhou. What kind of visual tokens do we need? training-free visual token pruning for multi-modal large language models from the perspective of graph. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 4075–4083, 2025.

Seil Kang, Jinyeong Kim, Junhyeok Kim, and Seong Jae Hwang. See what you are told: Visual attention sink in large multimodal models. In *International Conference on Learning Representations*, volume 2025, pp. 87676–87703, 2025.

Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg. Referitgame: Referring to objects in photographs of natural scenes. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 787–798, 2014.

Aniruddha Kembhavi, Mike Salvato, Eric Kolve, Minjoon Seo, Hannaneh Hajishirzi, and Ali Farhadi. A diagram is worth a dozen images. In *European conference on computer vision*, pp. 235–251. Springer, 2016.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th symposium on operating systems principles*, pp. 611–626, 2023.

Ao Li, Yuxiang Duan, Jinghui Zhang, Congbo Ma, Yutong Xie, Gustavo Carneiro, Mohammad Yaqub, and Hu Wang. Transprune: Token transition pruning for efficient large vision-language model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 39529–39538, 2026a.

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

Kaiyuan Li, Xiaoyue Chen, Chen Gao, Yong Li, and Xinlei Chen. Balanced token pruning: Accelerating vision language models beyond local optimization. *arXiv preprint arXiv:2505.22038*, 2025a.

Shiyu Li, Zi-Yuan Hu, Shijia Huang, Yanyang Li, Yiwu Zhong, and Liwei Wang. Sinkpruner: Sink-free visual token pruning for multimodal large language models. *arXiv preprint arXiv:2609.01004*, 2026b.

Wentong Li, Yuqian Yuan, Jian Liu, Dongqi Tang, Song Wang, Jie Qin, Jianke Zhu, and Lei Zhang. Tokenpacker: Efficient visual projector for multimodal llm. *International Journal of Computer Vision*, 133(10):6794–6812, 2025b.

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

Yuhong Li, Yingbing Huang, Bowen Yang, Bharat Venkitesh, Acyr Locatelli, Hanchen Ye, Tianle Cai, Patrick Lewis, and Deming Chen. Snapkv: Llm knows what you are looking for before generation. *Advances in Neural Information Processing Systems*, 37:22947–22970, 2024.

Youwei Liang, Chongjian Ge, Zhan Tong, Yibing Song, Jue Wang, and Pengtao Xie. Not all patches are what you need: Expediting vision transformers via token reorganizations. *arXiv preprint arXiv:2202.07800*, 2022.

Aoming Liu, Reuben Tan, Boqing Gong, and Bryan A Plummer. Fine-grained token allocation via operation pruning for efficient mllms. *arXiv preprint arXiv:2507.02909*, 2025.

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

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

Weijie Liu, Peng Zhou, Zhiruo Wang, Zhe Zhao, Haotang Deng, and Qi Ju. Fastbert: a self-distilling bert with adaptive inference time. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pp. 6035–6044, 2020.

Wenjie Liu, Hao Wu, Xin Qiu, Xudong Wang, Yingqi Fan, Yihan Zhang, Anhao Zhao, Yunpu Ma, and Xiaoyu Shen. Vica: Efficient multimodal llms with vision-only cross-attention. *arXiv preprint arXiv:2602.07574*, 2026.

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

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, 2024c.

Zichang Liu, Aditya Desai, Fangshuo Liao, Weitao Wang, Victor Xie, Zhaozhuo Xu, Anastasios Kyrillidis, and Anshumali Shrivastava. Scissorhands: Exploiting the persistence of importance hypothesis for llm kv cache compression at test time. *Advances in Neural Information Processing Systems*, 36:52342–52364, 2023b.

Zuyan Liu, Benlin Liu, Jiahui Wang, Yuhao Dong, Guangyi Chen, Yongming Rao, Ranjay Krishna, and Jiwen Lu. Efficient inference of vision instruction-following models with elastic cache. In *European Conference on Computer Vision*, pp. 54–69. Springer, 2024d.

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

Bozhi Luan, Wengang Zhou, Hao Feng, Zhe Wang, Xiaosong Li, and Houqiang Li. Multi-cue adaptive visual token pruning for large vision-language models. *arXiv preprint arXiv:2503.08019*, 2025.

Yaxin Luo, Gen Luo, Jiayi Ji, Yiyi Zhou, Xiaoshuai Sun, Zhiqiang Shen, and Rongrong Ji. $\gamma-$mod: Exploring mixture-of-depth adaptation for multimodal large language models. *arXiv preprint arXiv:2410.13859*, 2024.

Zhaoyang Luo, Runmin Dong, Miao Yang, Fan Wei, Yushan Lai, Bin Luo, and Haohuan Fu. Attend, transform, or silence: Operator-level visual skipping for efficient multimodal llm inference. *arXiv preprint arXiv:2606.31903*, 2026.

Jie Ma, Zhike Qiu, Jie Gao, Jiayi Ji, Qian Chen, Xiaoshuai Sun, and Rongrong Ji. Capturing token tendencies for training-free token pruning in multimodal large language models. *arXiv preprint arXiv:2607.28341*, 2026a.

Jie Ma, Zhike Qiu, Jiayi Ji, Xiaoshuai Sun, and Rongrong Ji. Look less, reason more: Block-wise attention skipping for efficient multimodal llms. *arXiv preprint arXiv:2606.08511*, 2026b.

Tanvir Mahmud, Burhaneddin Yaman, Chun-Hao Liu, and Diana Marculescu. Papr: Training-free one-step patch pruning with lightweight convnets for faster inference. In *European Conference on Computer Vision*, pp. 110–128. Springer, 2024.

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

Minesh Mathew, Dimosthenis Karatzas, and CV Jawahar. Docvqa: A dataset for vqa on document images. In *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 2199–2208. IEEE, 2021.

Minesh Mathew, Viraj Bagal, Rubèn Tito, Dimosthenis Karatzas, Ernest Valveny, and CV Jawahar. Infographicvqa. In *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 2582–2591. IEEE, 2022.

Lingchen Meng, Jianwei Yang, Rui Tian, Xiyang Dai, Zuxuan Wu, Jianfeng Gao, and Yu-Gang Jiang. Deepstack: Deeply stacking visual tokens is surprisingly simple and effective for lmms. *Advances in Neural Information Processing Systems*, 37:23464–23487, 2024.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026. URL `https://qwen.ai/blog?id=qwen3.5`.

Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. Dynamicvit: Efficient vision transformers with dynamic token sparsification. *Advances in neural information processing systems*, 34:13937–13949, 2021.

David Raposo, Sam Ritter, Blake Richards, Timothy Lillicrap, Peter Conway Humphreys, and Adam Santoro. Mixture-of-depths: Dynamically allocating compute in transformer-based language models. *arXiv preprint arXiv:2404.02258*, 2024.

Tal Schuster, Adam Fisch, Jai Gupta, Mostafa Dehghani, Dara Bahri, Vinh Tran, Yi Tay, and Donald Metzler. Confident adaptive language modeling. *Advances in Neural Information Processing Systems*, 35:17456–17472, 2022.

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.

Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards vqa models that can read. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8317–8326, 2019.

Zunhai Su, Wang Shen, Linge Li, Zhe Chen, Hanyu Wei, Huangqi Yu, and Kehong Yuan. Akvq-vl: Attention-aware kv cache adaptive 2-bit quantization for vision-language models. In *2025 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1–6. IEEE, 2025.

Zixun Sun, Yubo Dong, Hehe Fan, and Yi Yang. Hi-lo prune: Look at what you'll lose before pruning with hierarchical token selection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 31941–31951, 2026.

Surat Teerapittayanon, Bradley McDanel, and Hsiang-Tsung Kung. Branchynet: Fast inference via early exiting from deep neural networks. In *2016 23rd international conference on pattern recognition (ICPR)*, pp. 2464–2469. IEEE, 2016.

Dezhan Tu, Danylo Vashchilenko, Yuzhe Lu, and Panpan Xu. Vl-cache: Sparsity and modality-aware kv cache compression for vision-language model inference acceleration. In *International Conference on Learning Representations*, volume 2025, pp. 219–239, 2025.

Zhongwei Wan, Ziang Wu, Che Liu, Jinfa Huang, Zhihong Zhu, Peng Jin, Longyue Wang, and Li Yuan. Look-m: Look-once optimization in kv cache for efficient multimodal long-context inference. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 4065–4078, 2024.

Haicheng Wang, Zhemeng Yu, Gabriele Spadaro, Chen Ju, Victor Quétu, Shuai Xiao, and Enzo Tartaglione. Folder: Accelerating multi-modal large language models with enhanced performance. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 23614–23625, 2025.

Yahong Wang, Juncheng Wu, Zhangkai Ni, Chengmei Yang, Yihang Liu, Longzhen Yang, Yuyin Zhou, Ying Wen, and Lianghua He. Entropyprune: Matrix entropy guided visual token pruning for multimodal large language models. *arXiv preprint arXiv:2602.17196*, 2026.

Zichen Wen, Yifeng Gao, Weijia Li, Conghui He, and Linfeng Zhang. Token pruning in multimodal large language models: Are we solving the right problem? In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 15537–15549, 2025a.

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 *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 9972–9991, 2025b.

Hao Wu, Yingqi Fan, Dai Jinyang, Junlong Tong, Yunpu Ma, and Xiaoyu Shen. Hidrop: Hierarchical vision token reduction in mllms via late injection, concave pyramid pruning, and early exit. In *International Conference on Learning Representations*, volume 2026, pp. 97296–97316, 2026.

Qiong Wu, Wenhao Lin, Yiyi Zhou, Weihao Ye, Zhanpeng Zen, Xiaoshuai Sun, and Rongrong Ji. Accelerating multimodal large language models via dynamic visual-token exit and the empirical findings. *arXiv preprint arXiv:2411.19628*, 2024.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. *arXiv preprint arXiv:2309.17453*, 2023.

Ji Xin, Raphael Tang, Jaejun Lee, Yaoliang Yu, and Jimmy Lin. Deebert: Dynamic early exiting for accelerating bert inference. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pp. 2246–2251, 2020.

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

Yifan Xu, Zhijie Zhang, Mengdan Zhang, Kekai Sheng, Ke Li, Weiming Dong, Liqing Zhang, Changsheng Xu, and Xing Sun. Evo-vit: Slow-fast token evolution for dynamic vision transformer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pp. 2964–2972, 2022.

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, 2025a.

Senqiao Yang, Junyi Li, Xin Lai, Jinming Wu, Wei Li, Zejun MA, Bei Yu, Hengshuang Zhao, and Jiaya Jia. Visionthink: Smart and efficient vision language model via reinforcement learning. *Advances in Neural Information Processing Systems*, 38:95187–95227, 2026.

Sihan Yang, Runsen Xu, Chenhang Cui, Tai Wang, Dahua Lin, and Jiangmiao Pang. Vflowopt: A token pruning framework for lmms with visual information flow-guided optimization. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 23924–23934. IEEE, 2025b.

Zhewei Yao, Xiaoxia Wu, Conglong Li, Connor Holmes, Minjia Zhang, Cheng Li, and Yuxiong He. Random-ltd: Random and layerwise token dropping brings efficient training for large-scale transformers. *arXiv preprint arXiv:2211.11586*, 2022.

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. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 22128–22136, 2025a.

Xubing Ye, Yukang Gan, Xiaoke Huang, Yixiao Ge, and Yansong Tang. Voco-llama: Towards vision compression with large language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 29836–29846, 2025b.

Hongxu Yin, Arash Vahdat, Jose M Alvarez, Arun Mallya, Jan Kautz, and Pavlo Molchanov. A-vit: Adaptive tokens for efficient vision transformer. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10799–10808. IEEE, 2022.

Licheng Yu, Patrick Poirson, Shan Yang, Alexander C Berg, and Tamara L Berg. Modeling context in referring expressions. In *European conference on computer vision*, pp. 69–85. Springer, 2016.

Qianhao Yuan, Qingyu Zhang, Yanjiang Liu, Jiawei Chen, Yaojie Lu, Hongyu Lin, Jia Zheng, Xianpei Han, and Le Sun. Shortv: Efficient multimodal large language models by freezing visual tokens in ineffective layers. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 329–339. IEEE, 2025.

Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, et al. 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.

Weili Zeng, Ziyuan Huang, Kaixiang Ji, and Yichao Yan. Skip-vision: Efficient and scalable acceleration of vision-language models via adaptive token skipping. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 21384–21397. IEEE, 2025.

Jun Zhang, Desen Meng, Zhengming Zhang, Zhenpeng Huang, Tao Wu, and Limin Wang. p-mod: Building mixture-of-depths mllms via progressive ratio decay. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3705–3715, 2025a.

Pu Zhang, Yuwei Li, Xingyuan Xian, and Guoming Tang. Zspaprune: Zero-shot prompt-aware token pruning for vision-language models. *arXiv preprint arXiv:2510.17197*, 2025b.

Qizhe Zhang, Aosong Cheng, Ming Lu, Renrui Zhang, Zhiyong Zhuo, Jiajun Cao, Shaobo Guo, Qi She, and Shanghang Zhang. Beyond text-visual attention: Exploiting visual cues for effective token pruning in vlms. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 20857–20867, 2025c.

Qizhe Zhang, Mengzhen Liu, Lichen Li, Ming Lu, Yuan Zhang, Junwen Pan, Qi She, and Shanghang Zhang. Beyond attention or similarity: Maximizing conditional diversity for token pruning in mllms. *Advances in Neural Information Processing Systems*, 38:25438–25468, 2026.

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

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, et al. H2o: Heavy-hitter oracle for efficient generative inference of large language models. *Advances in Neural Information Processing Systems*, 36:34661–34710, 2023.

APPENDIX OVERVIEW

This appendix provides reproducibility details and supplementary analyses supporting the main results. Appendix A specifies the evaluation datasets and splits, model and inference settings, routing configurations, and position-index handling used throughout the experiments. Appendix B examines the routing-design choices behind Reroute, including the early-stage PDrop schedule, routing-frequency and layout sensitivity, and deferred-state staleness and recovery. Appendix C provides additional evidence beyond the main grounding results, including general VQA preservation, OCR-heavy evaluation on Qwen2.5-VL, boundary cases on Nüwa and Qwen3-VL, and qualitative grounding and token-selection visualizations. Together, these analyses document the experimental protocol and further characterize the robustness, recovery behavior, cross-task performance, and boundary conditions of Reroute.

A EXPERIMENTAL PROTOCOL AND CONFIGURATIONS

This section provides the implementation and evaluation details needed to reproduce the main results. We first describe the datasets, evaluation splits, model settings, and grounding post-processing conventions. We then specify the routing configurations, cross-model schedule normalization, and position-index handling used throughout the experiments.

A.1 EVALUATION DATASETS AND SPLITS

For visual grounding, we evaluate on the RefCOCO-family grounding suite in `lmms-eval`, including RefCOCO, RefCOCO+, and RefCOCOg (Kazemzadeh et al., 2014; Yu et al., 2016). The evaluated splits include RefCOCO (`refcoco_val`, `refcoco_testA`, `refcoco_testB`), RefCOCO+ (`refcoco+_val`, `refcoco+_testA`, `refcoco+_testB`), and RefCOCOg (`refcocog_val`, `refcocog_test`). These benchmarks evaluate whether a model can localize the object referred to by a natural-language expression. The model prediction is parsed as a bounding box and evaluated using the standard IoU-based grounding protocol implemented by `lmms-eval`.

To verify that visual-token reduction does not improve grounding at the expense of general multimodal understanding, we also evaluate on a general VQA suite. The core suite includes MME (Fu et al., 2023), GQA (Hudson & Manning, 2019), POPE (Li et al., 2023b), TextVQA (Singh et al., 2019), ScienceQA (Lu et al., 2022), and MMBench (Liu et al., 2024b). When available, we additionally report results on AI2D (Kembhavi et al., 2016), MMMU-Val (Yue et al., 2024), VQAv2-Lite (Goyal et al., 2017), MMStar (Chen et al., 2024c), and SEEDBench-Lite (Li et al., 2023a). These tasks cover general perception, object recognition, hallucination evaluation, OCR-oriented question answering, scientific reasoning, and broader multimodal reasoning.

For OCR- and document-centric evaluation, we additionally report OCRBench (Liu et al., 2024c), DocVQA (Mathew et al., 2021), ChartQA (Masry et al., 2022), and InfographicVQA (InfoVQA) (Mathew et al., 2022). OCRBench provides a broad evaluation of text-centric visual understanding, covering text recognition and multiple OCR-related reasoning tasks. DocVQA evaluates question answering over document images, while ChartQA focuses on visual and logical reasoning over charts. InfographicVQA evaluates question answering over infographics that combine text, layout, graphical elements, and data visualizations. Together, these benchmarks probe the preservation of fine-grained textual and structured visual evidence under visual-token reduction.

A.2 MODEL AND INFERENCE SETTINGS

We evaluate Reroute and its pruning baselines on LLaVA-1.5-7B, Qwen2.5-VL-7B, and Qwen3.5-9B. We additionally include Qwen3-VL-8B for architecture-level discussion and supplementary comparisons. Unless otherwise noted, each benchmark uses the default generation configuration provided by `lmms-eval`. All experiments are conducted on a single NVIDIA RTX 4090 GPU.

For visual grounding, prompt and post-processing conventions follow the corresponding model family. For LLaVA-1.5, we use the default `lmms-eval` prompt and post-processing pipeline. For Qwen-family models, we follow the official/Nuwa prompt format and the community-suggested grounding post-processing convention. In particular, Qwen2.5-VL bounding boxes are normalized by

Table 5: LLaVA-1.5 routing configurations at Avg. = 192. All schedules use keep-index positional handling.

Backbone	Setting	Layers	Routing layers
LLaVA-1.5-7B	PDrop (L8)	32	$\{8, 16, 24\}$
LLaVA-1.5-7B	PDrop (L2)	32	$\{2, 7, 15, 23\}$
LLaVA-1.5-7B	PDrop (L2) + Reroute	32	$\{2, 7, 15, 23\}$
LLaVA-1.5-7B	FastV (L3)	32	$\{3\}$
LLaVA-1.5-7B	FastV (L3) + Reroute	32	$\{3, 7, 15, 23\}$

Table 6: Cross-model routing-layer conventions used for Reroute.

Backbone	Setting	Layers	Routing layers	Rationale
LLaVA-1.5-7B	PDrop + Reroute	32	$\{2, 7, 15, 23\}$	Early PDrop-aligned schedule from the LLaVA ablation
LLaVA-1.5-7B	FastV + Reroute	32	$\{3, 7, 15, 23\}$	Main-table FastV setting uses $K = 3$
Qwen2.5-VL-7B	PDrop + Reroute	28	$\{2, 6, 13, 20\}$	Layer-normalized early PDrop-aligned schedule
Qwen2.5-VL-7B	FastV + Reroute	28	$\{3, 6, 13, 20\}$	Layer-normalized FastV-aligned schedule
Qwen3-VL-8B	Both	36	$\{3, 8, 17, 26\}$	Starts after DeepStack injection layers
Qwen3.5-9B	Both	32	$\{3, 7, 15, 23\}$	Snapped to compatible full-attention layers in the hybrid-attention backbone

pixel coordinates, which is already handled by `lmms-eval`, whereas Qwen3-VL and Qwen3.5 use $[0, 1000]$ coordinate normalization.

A.3 Routing configurations

We report the routing-layer configurations used in our experiments. LLaVA-1.5 contains 32 decoder layers and uses 576 visual tokens under the standard evaluation setting. Table 5 summarizes the LLaVA-1.5 configurations used in the main Avg. = 192 comparison.

PDrop introduces a configuration ambiguity because different routing-layer schedules can target the same average-token budget. We consider two conventions: a standard PDrop-style schedule that distributes routing decisions approximately evenly across decoder depth, e.g., $\{8, 16, 24\}$ for LLaVA-1.5, and an early-routing convention that starts near layer 2 and continues at later stages, e.g., $\{2, 7, 15, 23\}$. We denote these configurations as PDrop (L8) and PDrop (L2), respectively. Appendix B.1 shows that PDrop (L2) gives stronger grounding results, so we use the early-routing convention as the main PDrop-aligned schedule for Reroute.

For cross-model experiments, we preserve the same scheduling principle while normalizing routing layers to model depth and respecting architecture-specific constraints. Table 6 summarizes the cross-model routing-layer conventions used for Reroute.

A.4 Position-index handling for physical pruning

When visual tokens are physically pruned, the remaining tokens can either be re-indexed into a contiguous position sequence or keep their original visual position indices (Chien et al., 2025; Huang et al., 2026; Bagrov et al., 2025). Table 7 compares these two choices under FastV on LLaVA-1.5-7B. Re-indexing substantially degrades both general VQA and RefCOCO grounding, indicating that spatial position preservation is critical for localization after token removal.

Therefore, for all physical-pruning baselines, we use the keep-index convention: kept visual tokens preserve their original position indices after pruning. This makes the pruning baselines stronger and avoids conflating pruning failures with position-indexing artifacts. Reroute does not physically remove deferred tokens, so it preserves the original sequence layout by construction.

B Routing Schedule Analysis

This section analyzes the schedule-design choices behind the main Reroute configurations. Since Reroute is training-free and reuses the same attention-score ranking interface as pruning baselines, its main design degrees of freedom are the routing layers and per-stage retention ratios. We therefore study three related questions: whether PDrop should start early or follow its standard depth-spaced

Table 7: Re-indexing vs. preserved indexing under FastV on LLaVA-1.5-7B. Grounding results are reported using ACC@0.5.

Schedule	Avg.	PE handling	GQA	MMBench	RefCOCO A	RefCOCO B
FastV (K=2)	306	Re-index	5.3	16.6	3.9	3.3
FastV (K=2)	306	Keep-index	58.9	61.9	45.0	37.0
FastV (K=3)	192	Re-index	40.8	34.8	1.9	1.1
FastV (K=3)	192	Keep-index	55.0	58.6	27.8	22.3

Table 8: Schedule ablation on RefCOCO-family visual grounding at Avg. = 192 (66.7% reduction). All values report ACC@0.5. We compare the standard PDrop schedule, denoted PDrop (L8), with the early-routing schedule, denoted PDrop (L2), on the two backbones used for the main grounding analysis.

Model	Schedule	Avg.	RefCOCO			RefCOCO+			RefCOCOg	
			val	A	B	val	A	B	val	test
LLaVA-1.5-7B	PDrop (L8)	192	26.2	31.9	19.9	22.4	27.1	16.5	21.8	20.8
LLaVA-1.5-7B	PDrop (L2)	192	39.8	45.1	33.4	33.5	39.5	25.9	33.2	32.8
Qwen2.5-VL-7B	PDrop (L8)	192	33.4	39.3	28.5	28.9	34.6	23.3	32.7	33.1
Qwen2.5-VL-7B	PDrop (L2)	192	55.9	59.2	52.0	48.4	53.0	43.6	53.6	52.9

schedule, how many reroute stages are useful, and where these stages should be placed across decoder depth.

B.1 Early-stage routing for PDrop

For the full RefCOCO-family evaluation, we report ACC@0.5 as the primary grounding metric in Table 8. Across both evaluated backbones, PDrop (L2) consistently improves over PDrop (L8) at the same Avg. = 192 budget. This result supports using early-stage routing as the main PDrop-aligned schedule when comparing pruning against Reroute.

B.2 Frequency sweep and additional routing-layout configurations

We provide additional details for the routing-schedule ablations reported in the main paper. Table 9 lists the exact configurations used for the frequency sweep, where the first routing layer is fixed and the number of routing stages is varied. Table 10 reports additional routing-layout configurations with a fixed number of routing stages, covering early, uniform, late, and clustered schedules. In both cases, each routing stage uses the same per-stage retention ratio, $r_i\% = 0.5$. The realized average token ratio is reported explicitly because it can vary with the routing layout.

The frequency sweep shows that increasing the number of routing stages generally improves GQA and RefCOCO performance at a fixed average token ratio, with the gains saturating under denser routing. The additional routing-layout configurations show that performance can vary with the routing schedule, while closely related early configurations remain comparable. These results complement the main-text ablation by demonstrating the benefit of repeated reassessment and providing additional evidence across a broader range of routing schedules.

B.3 Deferred-state staleness and recovery

Reroute retains deferred visual tokens for later reassessment without updating their hidden states during bypass. Although this avoids deferred-token computation, the stored representations can diverge from the layer-matched full-compute states, introducing *hidden-state staleness*.

Table 11 examines the accuracy–computation trade-off underlying this design. ASkip continues FFN updates during deferral, while Full-compute supplies re-admitted tokens with layer-matched states from an additional full forward. These additional updates can improve accuracy, but Reroute retains 96.1–99.3% of Full-compute accuracy on GQA and MMBench and 84.2–97.2% on RefCOCO testA

Table 9: **Frequency-sweep configurations for reroute scheduling on LLaVA-1.5-7B.** All settings fix the first routing layer at layer 2 and use per-stage retention ratio $r_i\% = 0.5$. The sweep varies only the number of reroute stages. RefCOCO A/B denote testA/testB Acc@0.5.

Setting	#Stages	Routing layers	Avg. ratio	GQA	RefCOCO A	RefCOCO B
N1	1	$\{2\}$	0.5312	58.94	44.95	36.90
N2	2	$\{2, 16\}$	0.5312	59.03	44.56	36.31
N3	3	$\{2, 11, 20\}$	0.5312	59.24	45.48	37.49
N4	4	$\{2, 9, 16, 23\}$	0.5312	59.56	48.49	37.45
N6	6	$\{2, 7, 12, 17, 22, 27\}$	0.5312	59.72	50.93	39.33
N8	8	$\{2, 6, 10, 14, 18, 22, 26, 30\}$	0.5312	59.79	52.22	41.39
Dense	30	$\{2, 3, \ldots, 31\}$	0.5312	59.88	53.51	41.18

Table 10: **Additional routing-layout configurations on LLaVA-1.5-7B.** All settings use four routing stages with per-stage retention ratio $r_i\% = 0.5$ and vary their placement across decoder depth. The realized average token ratio is reported explicitly because it changes with the routing layout. RefCOCO A/B denote testA/testB Acc@0.5.

Placement	Routing layers	Avg. ratio	GQA	RefCOCO A	RefCOCO B
Early	$\{2, 5, 9, 13\}$	0.531	60.05	53.07	41.28
Uniform	$\{4, 12, 20, 28\}$	0.563	59.31	46.88	37.74
Late	$\{9, 15, 21, 27\}$	0.641	60.61	53.81	40.29
Very late	$\{15, 19, 23, 27\}$	0.734	60.53	59.41	47.93
Clustered early	$\{2, 4, 7, 11\}$	0.531	59.89	53.08	41.06
Clustered late	$\{20, 23, 26, 29\}$	0.813	60.52	59.02	47.91

without the extra deferred-state computation. Reroute thus preserves much of the accuracy available from fresher representations while retaining the computational savings of deferral.

Reroute mitigates staleness through re-entry rather than continuous updates to inactive tokens. Figure 6 shows that deferred representations can partially recover toward the full-compute trajectory after normal decoder computation resumes, as observed for the layer-15 and layer-23 cohorts. This recovery occurs through subsequent decoder updates without recomputing the skipped layers, allowing Reroute to recover visual evidence without maintaining fully updated states throughout deferral.

The placement ablation in Figure 5(b) further informs the timing of recovery. Placing the second routing stage in the early-to-middle decoder provides the greatest benefit, whereas substantially delayed re-entry reduces performance toward the single-stage FastV baseline. This pattern is consistent with accumulated staleness limiting the effectiveness of later recovery and supports Reroute's use of repeated reassessment to provide timely re-entry opportunities.

C Supplementary Results and Visualizations

This section provides additional evidence beyond the main grounding tables. We first report general VQA results to verify that Reroute does not sacrifice broad multimodal capability. We then discuss Qwen3-VL as an architecture-sensitive case study, followed by qualitative grounding and token-selection visualizations.

C.1 General VQA preservation

We report general VQA benchmark results in Tables 12, 3, 13, 14, and 15. These results serve as a preservation test rather than the primary source of improvement. Since many VQA benchmarks can be answered from coarse or globally salient visual evidence, the key question is whether Reroute maintains broad multimodal capability under the same average-token budget.

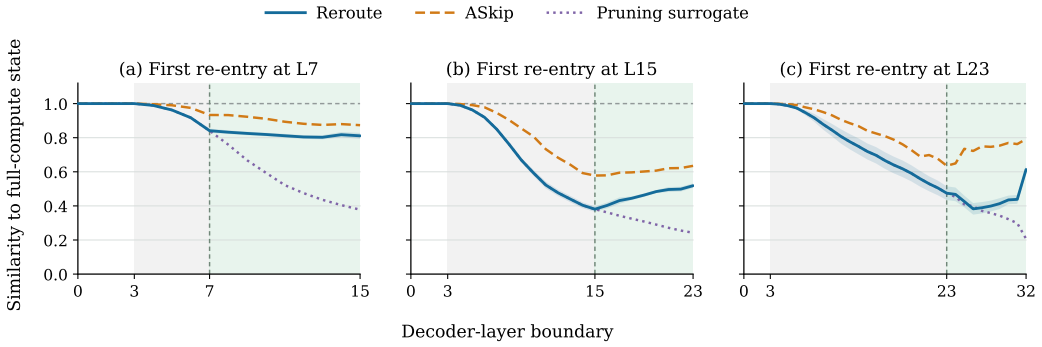


Figure 6: **Representation recovery after first re-admission.** Visual tokens initially deferred at layer 3 are grouped by first re-entry at layers 7, 15, and 23 under the approximately 128-token budget on LLaVA-1.5-7B. Similarity is measured against the corresponding full-compute hidden state at the same decoder-layer boundary. Curves average token similarity within each of five RefCOCO+ control images and then average equally across images; the blue band shows between-image ± 1 standard deviation. ASkip replays the Reroute masks and applies FFN updates to deferred tokens, while the pruning surrogate holds the layer-3 routing-input state fixed. Gray denotes deferral, green denotes the first active stage, vertical dashed lines mark first re-entry, and the horizontal dashed line at 1.0 denotes exact agreement with the full-compute state. After re-admission, Reroute partially recovers toward the full-compute trajectory.

Table 11: **Effect of deferred-state freshness on accuracy.** Reroute leaves deferred states unchanged until re-entry, while ASkip and Full-compute introduce progressively more computation to maintain fresher representations. Fresher states recover additional accuracy, especially under tighter token budgets, whereas Reroute preserves most of this accuracy without the additional deferred-state computation. The final column reports Reroute accuracy normalized by Full-compute.

Nominal Avg. T	Metric	Full-compute	ASkip	Reroute	Reroute / Full
192	GQA	59.94	59.68	58.94	98.3%
	MMBench	61.77	61.68	61.34	99.3%
	RefCOCO A	50.96	51.81	49.53	97.2%
128	GQA	58.40	57.81	57.25	98.0%
	MMBench	60.74	59.45	59.79	98.4%
	RefCOCO A	43.10	42.44	39.60	91.9%
64	GQA	56.56	54.97	54.38	96.1%
	MMBench	58.25	56.36	57.39	98.5%
	RefCOCO A	27.70	27.91	23.32	84.2%

Across LLaVA-1.5-7B and Qwen-family backbones, Reroute is generally comparable to, and often improves over, its corresponding pruning baseline. The trend is clearest for PDrop + Reroute, where multi-stage routing gives deferred tokens later opportunities to re-enter computation; for example, on Qwen2.5-VL-7B, the average-ratio score improves over PDrop at all three reduction levels. FastV + Reroute is less stable under extreme reduction because FastV depends on a single early scoring layer, but remains broadly comparable under moderate budgets. Overall, these results indicate that Reroute's grounding gains do not come from sacrificing general VQA performance. Instead, recoverable routing provides a robust alternative to irreversible pruning, especially when paired with multi-stage schedules.

C.2 OCR-HEAVY EVALUATION ON QWEN2.5-VL

We further evaluate Reroute on OCR-heavy benchmarks that require preserving small, localized visual evidence under aggressive token reduction. Table 16 reports results on OCRBench, DocVQA,

Table 12: VQA performance comparison on LLaVA-1.5-7B under the original LLaVA / Nüwa official-codebase format. Prior-method rows are paper-reported results, while rows marked with * are reproduced in the official Nüwa codebase. Best and second-best results within each average-token block are shown in **bold** and underlined, respectively.

Method	Source	GQA	MMB	MMMU	MME	TextVQA	POPE	SQA	Avg. Ratio
Average Token 576									
Vanilla	CVPR'24	**61.9**	**64.7**	**36.3**	**1862**	**58.2**	**85.9**	**69.5**	**100.0%**
Average Token 192 ↓ 66.7%									
FastV	ECCV'24	52.7	61.2	34.3	1612	52.5	64.8	67.3	89.0%
PDrop	CVPR'25	57.1	63.2	34.1	1766	56.1	82.3	**70.2**	96.0%
SparseVLM	ICML'25	57.6	62.5	33.8	1721	56.1	83.6	69.1	95.5%
VisionZip	CVPR'25	59.3	63.0	**36.6**	1782	57.3	85.3	68.9	98.1%
Nüwa	ICLR'26	60.9	**64.3**	35.5	1834	57.4	**86.4**	68.2	98.8%
Nüwa+Reroute*	Ours	**61.1**	63.6	35.7	**1848**	**57.8**	86.0	68.6	**99.0%**
Average Token 128 ↓ 77.8%									
FastV	ECCV'24	49.6	56.1	34.9	1490	50.6	59.6	60.2	83.7%
PDrop	CVPR'25	56.0	61.1	34.2	1664	55.1	82.3	**69.9**	94.2%
SparseVLM	ICML'25	56.0	60.0	33.8	1696	54.9	80.5	67.1	93.1%
VisionZip	CVPR'25	57.6	62.0	**37.9**	1761	56.8	83.2	68.9	97.4%
PruMerge	ICCV'25	57.8	59.6	36.2	1712	54.3	81.5	67.6	94.7%
Nüwa	ICLR'26	60.2	**63.4**	35.8	**1828**	57.0	**85.5**	67.8	98.2%
Nüwa+Reroute*	Ours	**60.4**	63.2	36.1	1827	**57.3**	85.3	68.2	**98.4%**
Average Token 64 ↓ 88.9%									
FastV	ECCV'24	46.1	48.0	34.0	1256	47.8	59.6	51.1	76.4%
PDrop	CVPR'25	41.9	33.3	26.5	1092	45.9	55.9	69.2	70.6%
SparseVLM	ICML'25	53.8	60.1	35.4	1589	53.4	77.5	**69.8**	92.2%
VisionZip	CVPR'25	55.1	60.1	36.2	1690	**55.5**	77.0	69.0	93.8%
PruMerge	ICCV'25	55.4	59.6	35.8	1616	52.0	75.7	69.5	92.1%
Nüwa	ICLR'26	58.3	**62.0**	**36.4**	**1706**	54.9	**83.0**	67.5	**95.7%**
Nüwa+Reroute*	Ours	**58.5**	61.9	35.9	1688	55.1	82.6	67.7	95.4%

Vanilla baseline ; Other methods ; Base methods ; Reroute variants .

*: reproduced in the official Nüwa codebase with Reroute applied. Unavailable results are denoted by "–". MMB denotes MMBench; SQA denotes ScienceQA. Avg. Ratio is the mean of per-benchmark performance ratios relative to the corresponding full-token baseline across the seven reported benchmarks.

ChartQA, and InfoVQA using Qwen2.5-VL-7B under the same matched reduction settings as the grounding experiments.

Reroute improves both FastV and PDrop across all four benchmarks and all evaluated token budgets. For FastV, the average ratio increases from 53.6% to 62.8% at 66.7% reduction and from 33.5% to 44.1% at 77.8% reduction. The gains are particularly pronounced for PDrop, where Reroute improves the average ratio from 61.8% to 69.5%, 39.7% to 52.6%, and 22.7% to 33.4% across the three reduction levels. At 77.8% reduction, for example, PDrop+Reroute improves OCRBench from 20.7 to 35.6 and ChartQA from 39.48 to 54.68.

These results show that the benefits of Reroute extend beyond spatial grounding to OCR-heavy settings. The consistent improvements across FastV and PDrop indicate that recoverable routing is particularly useful when aggressive token reduction must preserve fine-grained textual and structured visual evidence.

Table 13: VQA performance comparison on Qwen2.5-VL-7B. Best and second-best results within each average-token block are shown in **bold** and underlined, respectively.

Method	Source	GQA	POPE	MMB	SQA	TextVQA	AI2D	MMMU	MMStar	SEED	VQAv2-lite	MME	Avg. Ratio
Average Token (100%)													
Vanilla	Qwen'25	**60.8**	**87.8**	**83.7**	**87.8**	**81.0**	**82.6**	**48.7**	**62.3**	**79.8**	**78.6**	**2318**	**100.0%**
Average Token Reduction ↓ 66.7%													
FastV	ECCV'24	54.0	84.6	77.8	82.8	80.1	72.5	**49.6**	50.3	66.9	72.9	2178	92.0%
PDrop	CVPR'25	55.7	84.7	78.6	**85.0**	81.3	**75.0**	48.4	52.6	68.7	75.8	2213	93.8%
FastV+Reroute	Ours	55.9	85.2	77.3	82.5	77.9	73.2	47.3	50.6	68.1	76.3	2184	92.3%
PDrop+Reroute	Ours	**57.0**	**85.8**	**79.0**	84.8	**82.2**	74.8	46.7	**53.5**	**69.5**	**77.6**	**2243**	**94.4%**
Average Token Reduction ↓ 77.8%													
FastV	ECCV'24	48.2	79.5	66.5	77.1	74.2	67.5	44.9	41.0	61.0	64.2	1994	83.0%
PDrop	CVPR'25	50.7	80.5	72.8	**82.0**	77.5	**71.5**	**47.0**	46.8	62.0	67.6	2056	87.4%
FastV+Reroute	Ours	52.1	80.8	68.1	78.8	72.1	68.1	45.0	42.4	59.4	72.4	2002	84.9%
PDrop+Reroute	Ours	**53.8**	**83.1**	**75.4**	81.2	**80.3**	71.1	45.8	**47.5**	**62.6**	**74.1**	**2078**	**89.4%**
Average Token Reduction ↓ 88.9%													
FastV	ECCV'24	36.3	49.3	32.0	73.2	13.6	64.7	41.2	30.6	43.0	37.7	1127	56.1%
PDrop	CVPR'25	44.9	69.3	**67.4**	**79.4**	53.5	**68.0**	**45.4**	39.7	53.7	58.2	1755	76.9%
FastV+Reroute	Ours	42.2	58.3	36.3	74.1	36.7	64.3	41.7	30.8	45.3	52.3	1247	63.5%
PDrop+Reroute	Ours	**49.3**	**77.0**	66.0	77.2	**65.0**	67.8	44.4	**41.0**	**57.2**	**68.6**	**1874**	**81.3%**

Vanilla baseline ; Base methods ; Reroute variants .

Avg. Ratio is the mean score ratio against the vanilla baseline, computed over subsets where both scores are available. Schedule details are reported in the appendix.

Table 14: VQA performance comparison on Qwen3-VL-8B. Best and second-best results within each average-token block are shown in **bold** and underlined, respectively.

Method	Source	GQA	POPE	MMB	SQA	TextVQA	AI2D	MMMU	MMStar	SEED	VQAv2-lite	MME	Avg. Ratio
Average Token (100%)													
Vanilla	Qwen'25	**61.7**	**88.9**	**84.4**	**94.5**	**81.6**	**83.8**	**52.2**	**61.8**	**80.4**	**80.0**	**2374**	**100.0%**
Average Token Reduction ↓ 66.7%													
FastV	ECCV'24	54.1	85.4	**76.6**	**87.3**	76.7	**74.8**	49.2	49.1	72.7	70.6	2012	89.8%
PDrop	CVPR'25	56.9	**87.6**	75.4	87.0	71.0	74.4	**50.3**	**50.1**	**77.6**	74.2	2008	90.8%
FastV+Reroute	Ours	54.7	85.4	71.3	81.2	73.0	71.0	48.3	43.7	71.7	71.8	1902	86.5%
PDrop+Reroute	Ours	**57.5**	87.1	75.8	86.9	**77.8**	74.6	49.6	48.4	76.8	**75.1**	**2039**	**91.4%**
Average Token Reduction ↓ 77.8%													
FastV	ECCV'24	49.4	80.7	68.0	**83.7**	68.5	69.5	48.8	43.0	69.3	63.4	1756	82.7%
PDrop	CVPR'25	51.7	83.5	67.9	80.8	57.6	68.9	47.9	**43.6**	70.7	62.5	1812	81.9%
FastV+Reroute	Ours	50.4	79.1	62.7	77.3	64.2	67.2	48.0	40.0	64.0	62.1	1659	79.0%
PDrop+Reroute	Ours	**53.6**	**84.2**	**70.3**	81.6	**71.7**	**70.2**	**49.1**	42.5	**71.9**	**69.3**	**1839**	**85.4%**
Average Token Reduction ↓ 88.9%													
FastV	ECCV'24	37.3	56.9	29.6	73.8	16.0	64.9	44.9	30.7	48.9	40.8	1195	57.5%
PDrop	CVPR'25	45.3	73.8	52.6	75.5	33.5	65.3	46.8	35.4	58.4	51.6	1525	69.6%
FastV+Reroute	Ours	39.8	57.9	29.0	73.3	20.0	64.5	44.1	30.1	47.9	42.7	1177	58.1%
PDrop+Reroute	Ours	**49.3**	**77.5**	**60.0**	**77.1**	**60.6**	**66.8**	**46.9**	**38.6**	**63.8**	**60.6**	**1661**	**77.4%**

Vanilla baseline ; Base methods ; Reroute variants .

Avg. Ratio is the mean score ratio against the vanilla baseline, computed over subsets where both scores are available. Schedule details are reported in the appendix.

C.3 Boundary Cases of Recoverable Routing

While Reroute improves grounding in most evaluated settings, the magnitude of its benefit varies across pruning pipelines and architectures. We examine two informative boundary cases: the relatively small gains when Reroute is integrated into Nüwa, and the degradation of Qwen3-VL (Bai et al., 2025a) grounding under moderate token reduction.

Table 15: VQA performance comparison on Qwen3.5-9B-Hybrid. Best and second-best results within each average-token block are shown in **bold** and underlined, respectively.

Method	Source	GQA	POPE	MMB	SQA	TextVQA	AI2D	MMMU	MMStar	SEED	VQAv2-lite	MME	Avg. Ratio
Average Gated Attention Token (100%)													
Vanilla	Qwen'26	**61.7**	**89.6**	**84.0**	**93.5**	**82.8**	**85.2**	**48.1**	**46.8**	**77.0**	**79.6**	**2378**	**100.0%**
Average Gated Attention Token Reduction ↓ 66.7%													
FastV	ECCV'24	52.8	81.4	69.9	82.8	48.0	72.0	45.3	35.4	70.1	62.8	1823	82.5%
PDrop	CVPR'25	56.6	85.5	74.2	86.7	65.0	74.6	44.3	37.4	72.1	68.6	1924	87.9%
FastV+Reroute	Ours	58.4	84.7	76.9	87.2	77.4	75.3	44.3	38.9	71.1	**77.2**	1929	91.1%
PDrop+Reroute	Ours	**59.6**	**87.2**	**80.0**	**89.9**	**79.6**	**77.3**	**48.1**	**40.0**	**72.7**	75.4	**2042**	**93.9%**
Average Gated Attention Token Reduction ↓ 77.8%													
FastV	ECCV'24	47.5	74.7	58.9	78.3	28.0	69.2	43.2	31.2	61.8	53.1	1547	72.5%
PDrop	CVPR'25	51.7	80.1	65.7	79.3	46.2	71.4	44.3	33.2	64.9	58.6	1688	78.9%
FastV+Reroute	Ours	56.4	82.5	72.6	82.0	71.7	72.2	**45.0**	35.1	65.1	72.3	1798	86.2%
PDrop+Reroute	Ours	**57.3**	**83.9**	**75.6**	**84.9**	**74.6**	**74.3**	44.6	**37.7**	**71.7**	**74.4**	**1932**	**89.6%**
Average Gated Attention Token Reduction ↓ 88.9%													
FastV	ECCV'24	38.4	55.7	33.2	73.1	9.8	66.8	43.3	20.1	46.9	38.9	1263	57.1%
PDrop	CVPR'25	44.8	69.0	51.7	75.3	24.3	68.9	42.3	27.7	56.6	47.3	1431	67.5%
FastV+Reroute	Ours	51.0	68.7	64.0	78.5	56.0	67.3	42.8	29.5	56.8	64.5	1539	76.1%
PDrop+Reroute	Ours	**54.9**	**79.7**	**70.8**	**80.8**	**68.3**	**71.9**	**45.1**	**34.7**	**64.9**	**70.8**	**1711**	**84.4%**

Vanilla baseline ; Base methods ; Reroute variants .

Avg. Ratio is the mean score ratio against the vanilla baseline, computed over subsets where both scores are available. Schedule details are reported in the appendix.

Table 16: **OCR-heavy evaluation on Qwen2.5-VL-7B.** Results on OCRBench, DocVQA, ChartQA, and InfoVQA under matched visual-token reduction. These tasks stress small, localized visual evidence and structured text-rich reasoning. Avg. Ratio is normalized to the full-token vanilla baseline.

Method	Src.	OCRBench	DocVQA	ChartQA	InfoVQA	Avg. Ratio
Vanilla	Qwen'25	83.10	91.98	81.60	70.11	100.0%
Average Token Reduction ↓ 66.7%						
FastV	ECCV'24	48.70	44.91	49.76	32.28	53.6%
PDrop	CVPR'25	41.40	61.65	60.76	39.11	61.8%
FastV+Reroute	Ours	**51.50**	60.54	58.36	36.34	62.8%
PDrop+Reroute	Ours	45.30	**70.26**	**68.08**	**44.74**	**69.5%**
Average Token Reduction ↓ 77.8%						
FastV	ECCV'24	29.80	24.76	28.36	25.43	33.5%
PDrop	CVPR'25	20.70	41.00	39.48	28.79	39.7%
FastV+Reroute	Ours	34.60	42.04	39.60	28.52	44.1%
PDrop+Reroute	Ours	**35.60**	**49.25**	**54.68**	**32.97**	**52.6%**
Average Token Reduction ↓ 88.9%						
FastV	ECCV'24	4.10	9.42	12.76	23.00	15.9%
PDrop	CVPR'25	6.20	20.27	20.88	25.16	22.7%
FastV+Reroute	Ours	4.20	13.22	15.16	24.18	18.1%
PDrop+Reroute	Ours	**19.30**	**30.07**	**32.20**	**26.78**	**33.4%**

Vanilla baseline ; Base methods ; Reroute variants .

Nüwa combines spatially preserving pre-decoder aggregation with text-guided decoder pruning (Huang et al., 2026). We apply Reroute only to the second-stage pruning while leaving the aggregation stage unchanged. On RefCOCO-family grounding, Nüwa+Reroute remains comparable to Nüwa at 192 and 128 average tokens, and improves the average ratio from 54.6% to 55.8% at

Table 17: **OCRBench evaluation on LLaVA-1.5-7B.** We compare Nüwa and Nüwa+Reroute under matched average-token budgets. Results are obtained using the official Nüwa codebase.

Method	Avg. Tokens	OCRBench
Vanilla	576	31.4
Nüwa	192	23.9
Nüwa + Reroute	192	**29.7**
Nüwa	128	20.3
Nüwa + Reroute	128	**28.6**
Nüwa	64	15.7
Nüwa + Reroute	64	**26.6**

Vanilla baseline ; Base methods ; Reroute variants .

64 tokens (Table 1(a)). We attribute these smaller marginal gains to Nüwa's first-stage aggregation, which already preserves much of the grounding-relevant visual evidence before decoder pruning, leaving less information for Reroute to recover.

OCR-heavy evaluation further supports this interpretation. On OCRBench, Reroute improves Nüwa by 5.8, 8.3, and 10.9 accuracy points at average budgets of 192, 128, and 64 tokens, respectively (Table 17). These larger gains are consistent with fine-grained textual evidence remaining susceptible to mismatched selection in the second stage, leaving greater scope for later reassessment and recovery.

Qwen3-VL provides a complementary architecture-sensitive case. As shown in Table 18, PDrop+Reroute decreases the average grounding ratio from 69.5% to 57.2% at 66.7% token reduction and from 38.9% to 36.1% at 77.8% reduction. Under the most aggressive 88.9% reduction, however, Reroute improves the average ratio from 12.9% to 19.1%.

One possible explanation follows the same intuition as the Nüwa case. With stronger visual evidence available in Qwen3-VL (Meng et al., 2024), task-relevant information may be distributed more broadly across candidate tokens, leaving less differentiation for reassessment to exploit under moderate reduction. As the token budget becomes tighter, however, the risk of excluding useful evidence increases, consistent with the benefit of recoverability re-emerging at 88.9% reduction.

These results show that the benefit of recoverable routing varies with how much task-relevant visual evidence remains vulnerable to decoder-side pruning, with larger gains on OCR-heavy evaluation and under the tightest Qwen3-VL token budget.

C.4 Additional qualitative results

We provide additional qualitative visualizations for RefCOCO-family grounding in Figures 9, 10, and 11. These examples cover LLaVA-1.5, Qwen2.5-VL, and Qwen3.5, and compare pruning baselines with their corresponding Reroute variants under matched token budgets. We further visualize selected visual-token masks for LLaVA-1.5 and Qwen2.5-VL in Figures 7 and 8, showing which image regions remain active under the routing policy.

Table 18: Performance comparison on RefCOCO-series visual grounding benchmarks using Qwen3-VL-8B. Best and second-best results within each average-token block are shown in **bold** and underlined, respectively.

Method	Source	RefCOCO			RefCOCO+			RefCOCOg		Avg.
		val	A	B	val	A	B	val	test	Ratio
Average Token (100%)										
Vanilla	Qwen'25	**89.3**	**90.4**	**84.7**	**83.3**	**85.0**	**76.3**	**87.2**	**87.2**	**100.0%**
Average Token Reduction ↓ 66.7%										
FastV	ECCV'24	49.8	55.1	54.6	42.3	47.6	46.6	53.5	53.6	59.1%
PDrop	CVPR'25	**63.4**	**64.8**	**60.6**	**55.5**	**57.3**	**51.2**	**61.6**	**61.0**	**69.5%**
FastV+Reroute	Ours	46.8	50.8	43.0	39.2	43.8	35.5	44.0	43.5	50.6%
PDrop+Reroute	Ours	58.1	53.8	46.8	50.5	47.2	39.0	48.7	48.0	57.2%
Average Token Reduction ↓ 77.8%										
FastV	ECCV'24	27.4	33.9	31.5	22.6	29.3	**28.1**	31.5	**31.9**	34.7%
PDrop	CVPR'25	**37.1**	**40.5**	**32.0**	**31.6**	**34.1**	27.4	**32.1**	31.5	**38.9%**
FastV+Reroute	Ours	27.0	30.4	24.9	22.1	24.8	20.2	24.0	23.6	28.8%
PDrop+Reroute	Ours	36.3	36.2	29.6	30.3	30.7	24.2	29.8	29.2	36.1%
Average Token Reduction ↓ 88.9%										
FastV	ECCV'24	6.3	7.6	4.2	4.1	4.1	3.5	5.1	5.3	5.9%
PDrop	CVPR'25	11.9	15.0	8.9	10.6	12.3	8.1	10.7	10.5	12.9%
FastV+Reroute	Ours	6.6	8.2	4.5	3.9	4.4	3.9	5.2	5.5	6.1%
PDrop+Reroute	Ours	**17.8**	**21.5**	**15.8**	**14.1**	**17.7**	**12.9**	**15.9**	**15.1**	**19.1%**

Vanilla baseline ; Base methods ; Reroute variants .

For RefCOCO and RefCOCO+, A/B denote testA/testB. Avg. Ratio is the mean score ratio against the vanilla baseline, computed over subsets where both scores are available. Schedule details are reported in the appendix.

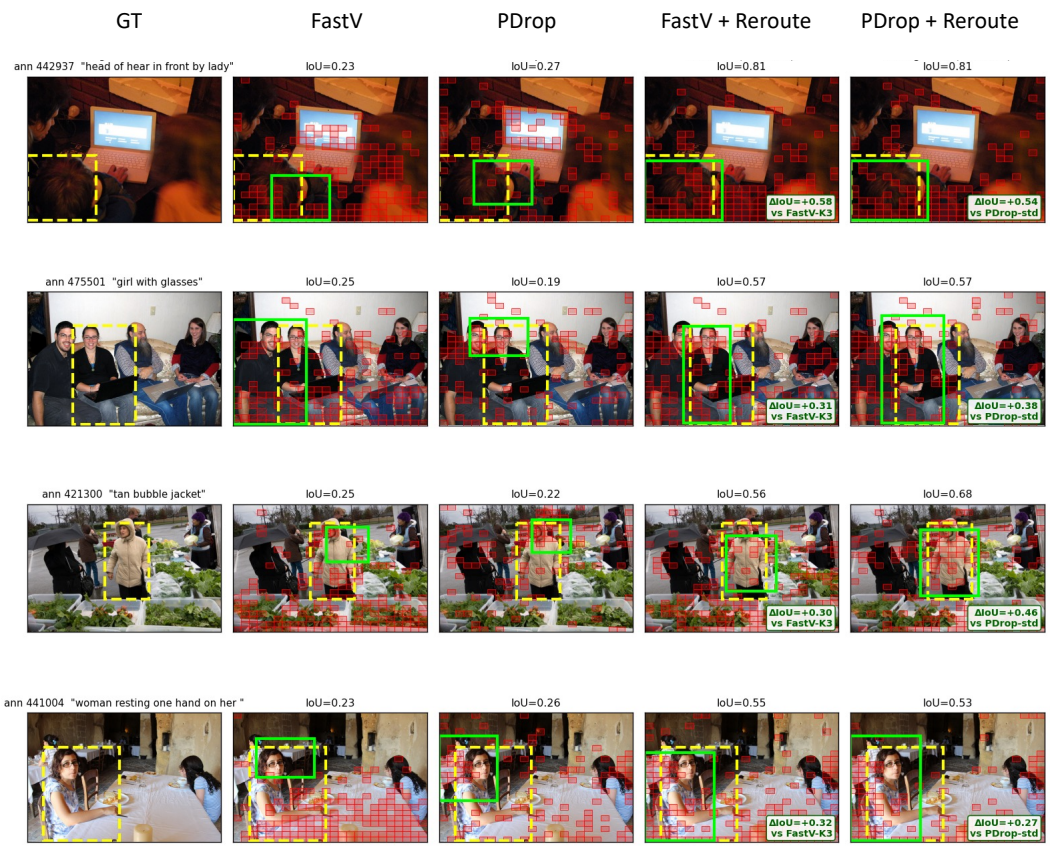


Figure 7: Selected visual-token masks on LLaVA-1.5. The highlighted regions indicate the visual tokens selected by the routing policy under the corresponding token budget.

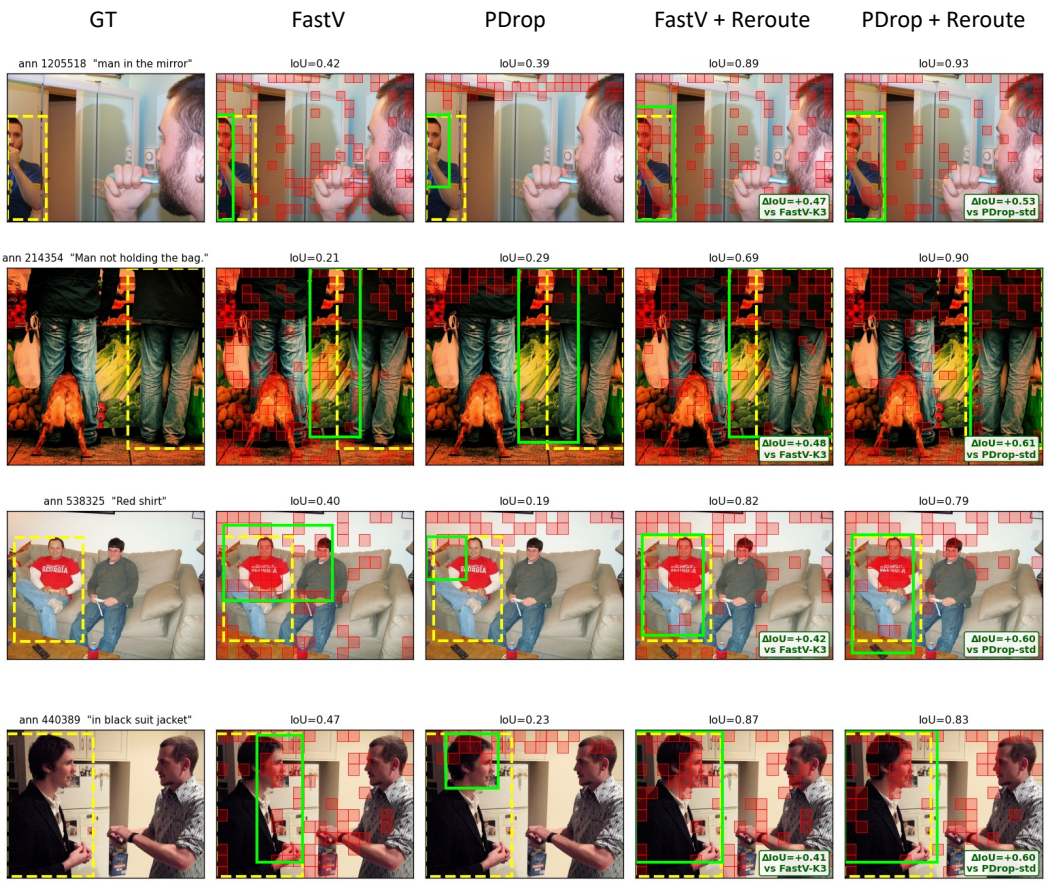


Figure 8: Selected visual-token masks on Qwen2.5-VL. The highlighted regions indicate the visual tokens selected by the routing policy under the corresponding token budget.

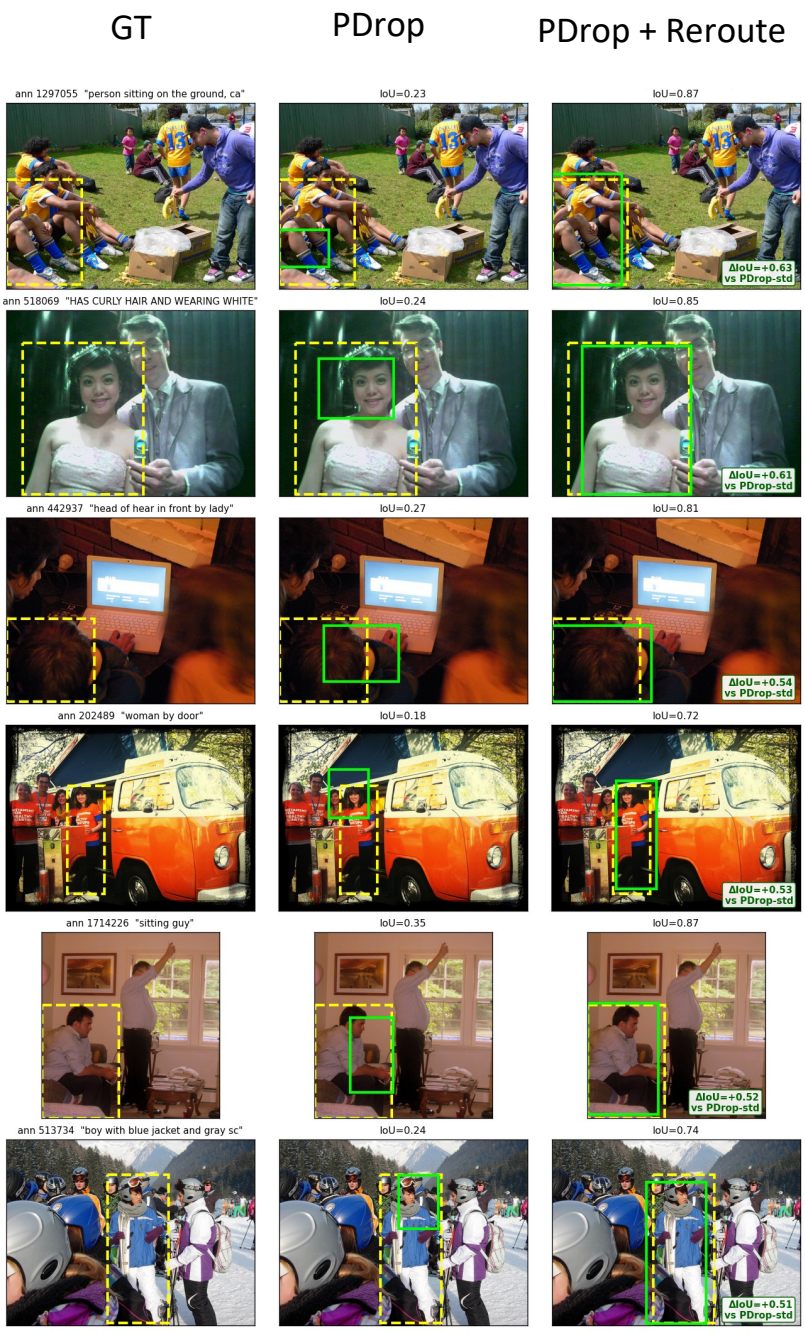


Figure 9: Additional RefCOCO-family grounding examples on LLaVA-1.5. We compare pruning baselines with their corresponding Reroute variants under matched token budgets.

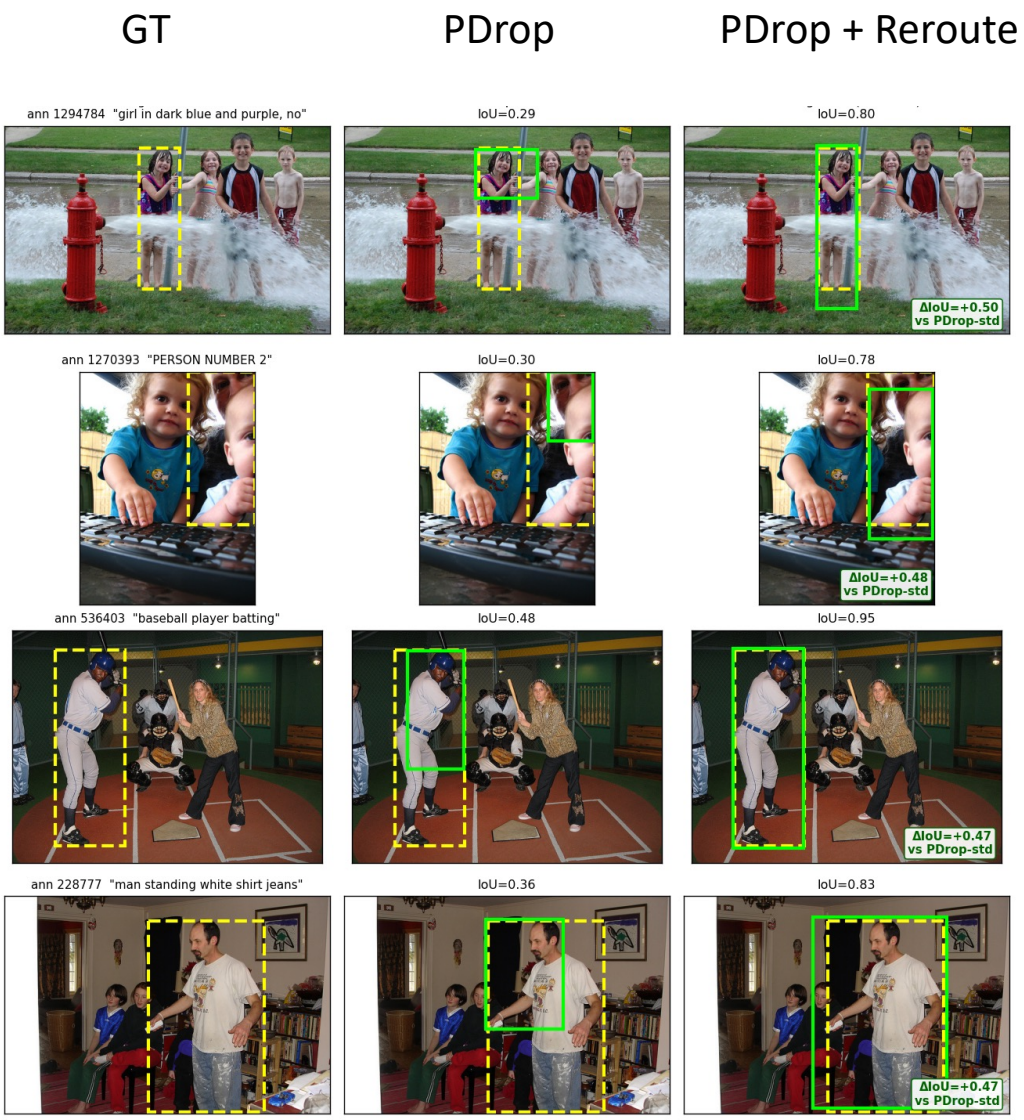


Figure 10: Additional RefCOCO-family grounding examples on Qwen2.5-VL. Reroute preserves deferred visual evidence and improves grounding over matched pruning baselines.



Figure 11: Additional RefCOCO-family grounding examples on Qwen3.5. We compare pruning baselines and their Reroute variants under matched gated-attention token budgets.