BEYOND TOKEN CONTENT: VISUAL TOKEN RETIREMENT BY TRACKING EVOLVING INFLUENCE

Anonymous authors
Paper under double-blind review

ABSTRACT

Token compression inside the vision encoder can reduce computation in both subsequent vision blocks and the downstream language model, but a key challenge is determining when a token can be safely retired while its representation is still evolving. Existing criteria largely rank tokens by redundancy estimated from their current representations or local relations, using signals such as salience and representativeness, without directly assessing whether subsequent computation still depends on each token's continued participation. Through controlled interventions, we show that retaining an intermediate token state as frozen key–value memory does not substitute for allowing the token to continue updating and interacting across later blocks. We therefore recast encoder-internal compression as token retirement and propose Influence-Confirmed Token Retirement (ICTR), a training-free and instruction-agnostic procedure that progressively shortens the sequence to a fixed terminal budget. We estimate external influence without additional forward passes by analytically computing how removing a token from the current key–value set would alter other tokens' outputs and averaging the largest relative changes. As representations and the active set evolve, we recompute external influence and confirm candidates whose ranks remain low across consecutive blocks. Across 10 benchmarks with LLaVA-1.5-7B, the method retains over 96% of dense-model performance on average at the 64-token budget, after retiring 88.9% of visual tokens; at the 128-token budget, it achieves a $2.05\times$ prefill speedup and reduces the KV-cache footprint by 71.2%.

1 INTRODUCTION

Large vision-language models (LVLMs) transform images into visual tokens through a vision encoder and then use a large language model (LLM) for cross-modal understanding and reasoning. Compressing visual tokens within the vision encoder shortens the sequences processed by both subsequent vision blocks and the language model. Existing work has introduced token compression into visual encoding, selecting, merging, or removing intermediate tokens according to learned importance, attention responses, feature similarity, regional structure, or question relevance (Rao et al., 2021; Liang et al., 2022; Bolya et al., 2023; Zhang et al., 2026; Wang et al., 2026b; Gao et al., 2026). Many of these decisions rely primarily on representations or local relations at a particular layer.

Recent work has moved beyond single-layer representations to model cross-layer processes. FlowCut treats redundancy as emerging progressively with visual depth and guides progressive pruning using multi-criteria importance accumulated across layers (Tong et al., 2025). EvoCut tracks token evolution directions through the vision encoder and estimates importance from persistent deviations from group evolution directions (Lu et al., 2026). On the language-model side, TransPrune accumulates changes in the magnitude and direction of token representations across successive modules as evidence for token selection (Li et al., 2026a). Although these methods differ in both their signals and intervention stages, they ultimately reduce cross-layer observations to per-token estimates of importance or redundancy. These estimates do not directly quantify how removing a candidate token from the key–value source set changes the updates of the other active tokens.

Within the vision encoder, a visual token both carries its current representation and serves as a query and key–value source in subsequent layers. Removing a token therefore not only discards its existing representation, but also terminates its future updates and queries, alters the information accessible to

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.

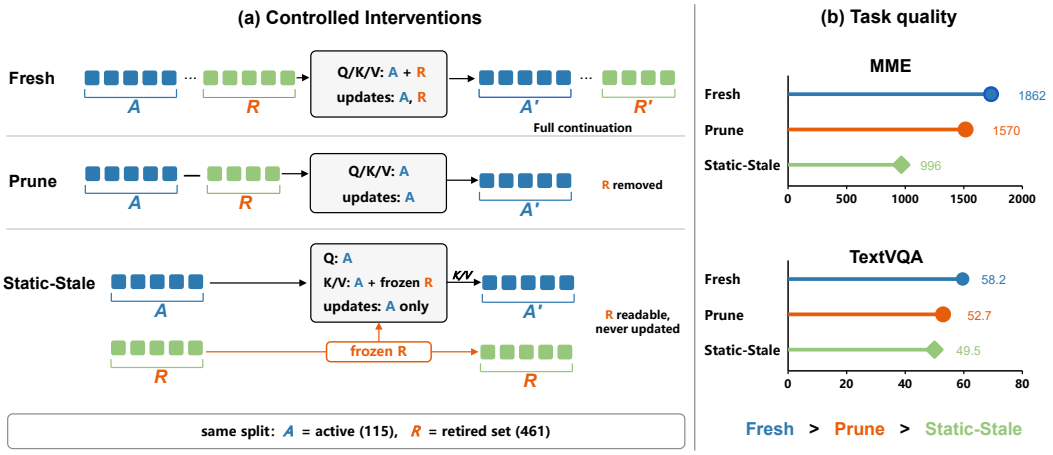

Figure 1: Controlled comparison of stored-state retention and continued token updating. (a) Treatment of tokens designated for removal under continued updating, direct removal, and frozen key–value retention, with the emitted token set fixed across settings. (b) MME and TextVQA scores for the corresponding settings.

other tokens, and changes their subsequent representation trajectories. This observation motivates us to track each token's key–value contribution to other tokens as encoding unfolds. We apply matched token-state and token-interaction interventions at different vision blocks and measure their effects on the final visual representations. Early and middle vision blocks are more sensitive, whereas the same interventions induce progressively smaller changes at greater depths. Token removability is therefore explicitly conditioned on encoder depth; the complete results are provided in the appendix. Depth dependence alone, however, does not reveal whether a token's utility resides in its stored content or in its continued participation in computation. Figure 1 fixes the set of tokens delivered at the encoder output and changes only how the remaining tokens participate in subsequent vision blocks. Compared with allowing these tokens to participate normally, deleting them early degrades performance. Retaining their intermediate states as frozen key–value sources does not recover this loss and, in some settings, performs even worse than direct deletion. Preserving a token's current representation is thus not equivalent to preserving its computational role in subsequent encoding. These results indicate that whether a token can leave the computation is not a static property attached to an individual representation; it depends on the token's cross-layer evolution and its continued interactions with the current active token set.

Based on this view, we recast visual-token compression as the retirement of computational nodes. We propose **Influence-Confirmed Token Retirement** (ICTR), which tracks each token's influence on other active tokens to determine when it can be removed from the vision encoder. We evaluate this influence by measuring the change in the attention-head outputs of other active tokens when a candidate is excluded from the key–value source set. This measurement jointly accounts for the attention assigned to the source and the discrepancy between its value and each recipient's current head output, without requiring a separate encoder forward pass for every candidate token. Aggregating the relative changes over the most affected recipients yields an external-influence score that remains sensitive to dependencies concentrated at only a few positions. Because retirement is irreversible and low influence at a single layer may be transient, we confirm only candidates that remain low-ranked by influence in two consecutive layers. After each retirement decision, influence is recomputed over the remaining sequence as the available source set changes. Retired tokens no longer participate in visual encoding: their states are neither updated nor retained as key–value memory. The resulting procedure requires neither additional training nor language instructions and progressively shortens the visual sequence to meet a prescribed final token budget.

We evaluate our method with LLaVA-1.5-7B on 10 image-understanding benchmarks and compare each compressed configuration with the uncompressed model under the same evaluation protocol. With final budgets of 192 and 128 visual tokens—removing 66.7% and 77.8%, respectively—the method retains 99.6% and 98.9% of average performance. At the 128-token budget, it reduces

measured prefill time by 51.2% (2.05× speedup) and the KV-cache footprint by 71.2%, including the overhead of external-influence computation and retirement control. These results show that determining token exit times from directly measured local influence within the active computation can substantially reduce downstream multimodal computation while preserving most capabilities.

Our contributions are as follows:

- We distinguish the stored representation of a visual token from its active computational role and show that sensitivity to token-level interventions varies markedly with visual depth, motivating the re-evaluation of removal decisions as encoder states evolve.
- We formulate encoder-internal compression as influence-confirmed token retirement. External influence measures the output changes induced in other active recipients by candidate-source removal, while consecutive-layer confirmation and re-evaluation over the active token set guide irreversible exits.
- We introduce a training-free, instruction-agnostic method that supports fixed token budgets. Without token merging, frozen key–value memory, or modifications to the downstream language model, it preserves high LVLM performance across moderate to aggressive compression settings.

2 Related Work

2.1 Visual Token Reduction within Vision Encoders

Token reduction within visual encoders predates LVLMs. Research on efficient Vision Transformers (ViTs) established several basic mechanisms, including learned pruning and adaptive halting, attention-guided reorganization and sampling, and similarity-based token merging (Rao et al., 2021; Yin et al., 2022; Liang et al., 2022; Fayyaz et al., 2022; Bolya et al., 2023). Although developed mainly for vision-only objectives, these methods show that intermediate visual sequences can be shortened progressively during encoding. Recent LVLM studies extend this idea, with most methods focusing on which tokens should remain. Attention and similarity provide common retention signals, while newer criteria incorporate spatial structure or task relevance. PosPrune distributes retained tokens across semantically rich regions, whereas QuietPrune maps the textual query into the vision Transformer to identify query-relevant token groups (Wang et al., 2026b; Gao et al., 2026). VisionDrop instead favors visual-only attention to avoid unreliable cross-modal guidance (Xu et al., 2026a). Going beyond token-wise ranking, SCoRe treats the retained tokens as a set and jointly optimizes salience and coverage through a representativeness objective (Xu et al., 2026b).

Other work concentrates on what happens to the information carried by removed tokens. LaCo and VScan reduce intermediate sequences through structured aggregation and merging, while FiCoCo-V transfers information from discarded tokens to correlated survivors (**?**Zhang et al., 2026; Han et al., 2026). IPCV addresses the effect of early pruning on bidirectional visual attention more directly: it temporarily reconstructs pruned tokens during attention and approximates their key–value contributions to limit feature distortion (Chen et al., 2025).

The closest line of work studies when reduction decisions become reliable. FlowCut shows that isolated layers can produce contradictory redundancy estimates and therefore accumulates current and historical importance signals across the encoder (Tong et al., 2025). SAINT adapts similarity-based pruning to different stages of token evolution, while EvoCut derives importance from token trajectories across multiple encoder layers (Jeddi et al., 2025; Lu et al., 2026). For video LLMs, EarlyTom further demonstrates the system-level benefit of moving frame merging into the vision encoder rather than compressing only after visual encoding (Wang et al., 2026a). Our work centers on this timing question, defining token retirement in terms of ongoing computational dependence.

2.2 Visual Token Compression after Visual Encoding

Encoder-internal compression modifies the token sequence while visual representations are still evolving. Methods applied to the final encoder output instead work with completed visual features, constructing a compact token set from encoder-side attention and using feature similarity or diversity to determine how the remaining information is merged or retained (Shang et al., 2025;

Yang et al., 2025; Zhang et al., 2025a). Other approaches compress visual tokens within the LLM, where interactions with the instruction and changes in hidden states provide additional evidence for token selection. LLM-side compression may be applied once in an early layer or progressively under layer- and input-dependent schedules, using cross-modal attention, feature similarity, or token recycling to limit information loss (Chen et al., 2024; Zhang et al., 2025b; Xing et al., 2025; Ye et al., 2025). Token dynamics provide another criterion: TransPrune and V2Drop measure token transitions or inter-layer variation (Li et al., 2026a; Chen et al., 2026), whereas VisiPruner relates the pruning schedule to distinct phases of cross-modal interaction (Fan et al., 2025). Dual-stage designs combine encoder-output compression with subsequent pruning inside the LLM (Singh et al., 2026). Although these methods reduce the visual sequence processed by all or part of the language model, the preceding vision-encoder computation remains dense.

3 METHOD

3.1 RETIREMENT DURING VISUAL ENCODING

Figure 2 presents *Influence-Confirmed Token Retirement*, which progressively removes visual tokens from an otherwise unchanged visual encoder. At each scheduled decision, the method operates on the tokens that remain active after previous retirements. For every retirement candidate, we estimate the change in the attention outputs of the other active tokens that would result from excluding it as a key–value source. This source-removal effect is computed directly from the attention weights, value vectors, and head outputs of the current block, without evaluating each candidate through a separate encoder forward pass.

A candidate is not retired solely because it has little influence at one layer. We rank the candidates by their effects on the most affected recipients and compare their rankings across two consecutive observations. Only tokens that remain weak sources at both observations are confirmed for retirement. Confirmed candidates are retired toward the current cumulative budget, while any shortfall is carried to subsequent decisions. Repeating this process progressively reduces the visual sequence to the prescribed terminal budget M. Retired tokens take no further part in encoding and are not retained as frozen key–value memory.

Let $\mathcal{A}_\ell$ contain the active visual-token identities when block ℓ is evaluated. Encoder-specific special tokens, when present, remain in the sequence but are not retirement candidates. If decision t retires $\mathcal{R}_t \subseteq \mathcal{A}_{\ell_t}$ after block ℓ_t, the active visual set entering the next block is therefore given by:

$$\mathcal{A}_{\ell_t+1} = \mathcal{A}_{\ell_t} \setminus \mathcal{R}_t. \tag{1}$$

The surviving states keep their sequence order. Both their representations and the sources available to them continue to change in later blocks, so every retirement decision uses a fresh measurement on the current sequence. We next describe how this measurement is obtained and how consecutive observations are converted into a retirement decision.

3.2 MEASURING EXTERNAL INFLUENCE

Consider recipient i in attention head h of block ℓ. Its output over the sources visible under the encoder's attention mask can be written as:

$$\mathbf{y}_{\ell,h,i} = \sum_{j \in \mathcal{K}_\ell(i)} \alpha_{\ell,h,i,j} \mathbf{v}_{\ell,h,j}, \tag{2}$$

where $\mathcal{K}_\ell(i)$ is the visible source set, $\alpha_{\ell,h,i,j}$ is the attention assigned by i to source j, and $\mathbf{v}_{\ell,h,j}$ is the corresponding value vector. Defining the sources through the encoder's own attention mask makes the measurement independent of a particular visual backbone or the presence of a global summary token.

Following the source-deletion analysis of IWP (Lee et al., 2026), we measure the local effect of a candidate source by the change in a recipient's normalized attention output when that source is excluded. For a candidate $s \in \mathcal{A}_\ell$, let $\mathbf{y}^{(-s)}_{\ell,h,i}$ denote the output obtained after removing s from the key–value source set while holding the current queries, keys, and values fixed. Since the logits of

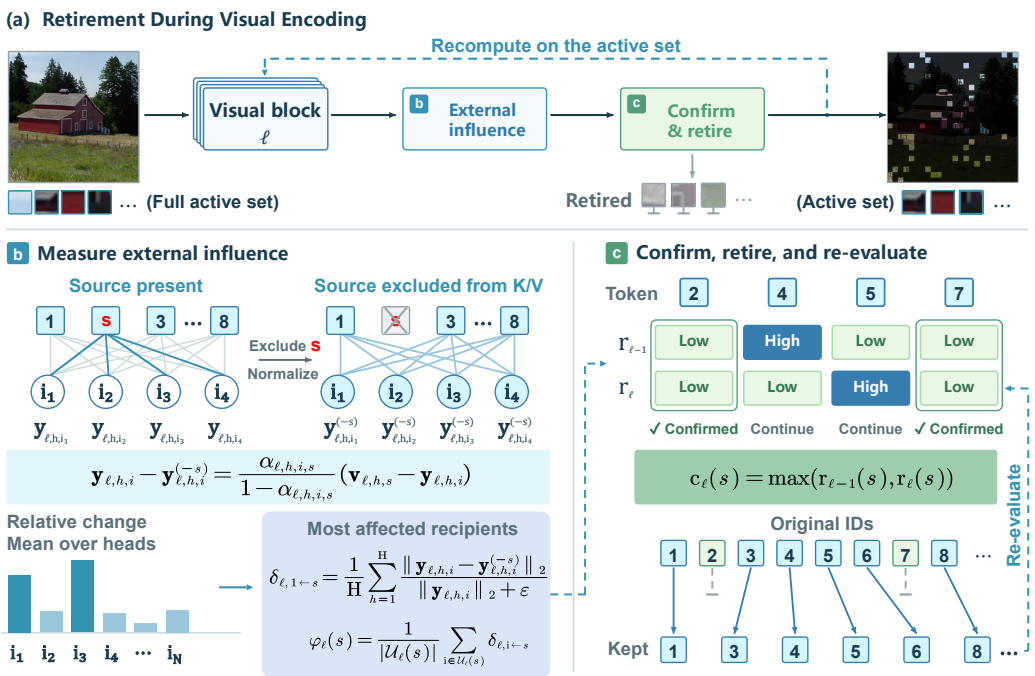

Figure 2: **Overview of Influence-Confirmed Token Retirement.** (a) Confirmed tokens are progressively removed from the active visual sequence. (b) External influence $\phi_\ell(s)$ measures the normalized change in other tokens' attention outputs when candidate source s is excluded, without additional encoder forward passes. (c) Candidates ranked low at consecutive observations are confirmed and retired under the target budget; influence is then recomputed on the shortened sequence.

the remaining sources are unchanged, renormalizing their attention weights yields:

$$\mathbf{y}_{\ell,h,i} - \mathbf{y}^{(-s)}_{\ell,h,i} = \frac{\alpha_{\ell,h,i,s}}{1-\alpha_{\ell,h,i,s}} \left(\mathbf{v}_{\ell,h,s} - \mathbf{y}_{\ell,h,i}\right). \tag{3}$$

The resulting change depends jointly on the attention assigned to the source and the discrepancy between its value and the recipient's current output. Because all terms are available from the current attention computation, the effects of all candidate sources can be evaluated without additional encoder forward passes.

Recipients and attention heads may have different output scales. Accordingly, we normalize the change by the original output magnitude and average it over the H heads to obtain:

$$\bar{\delta}_{\ell,i\leftarrow s} = \frac{1}{H}\sum_{h=1}^{H} \frac{\left\|\mathbf{y}_{\ell,h,i} - \mathbf{y}^{(-s)}_{\ell,h,i}\right\|_2}{\|\mathbf{y}_{\ell,h,i}\|_2 + \varepsilon}, \qquad i \neq s, \tag{4}$$

where $\varepsilon > 0$ prevents division by zero. The source's own output is excluded because the quantity of interest is its effect on the rest of the active computation.

A source can be critical to a small group of tokens even when its average effect over the entire sequence is weak. To preserve such localized dependencies, let $\mathcal{U}_\ell(s)$ contain the fraction η of active recipients with the largest $\bar{\delta}_{\ell,i\leftarrow s}$. Aggregating the effects over these recipients defines the external influence of s as:

$$\phi_\ell(s) = \frac{1}{|\mathcal{U}_\ell(s)|} \sum_{i\in\mathcal{U}_\ell(s)} \bar{\delta}_{\ell,i\leftarrow s}. \tag{5}$$

Thus, $\phi_\ell(s)$ is small only when excluding s causes little immediate change even among the recipients that depend on it most. The derivation and numerically stable evaluation of the source-removal effect are provided in Appendix B.1.

3.3 CONFIRMING RETIREMENT ACROSS LAYERS

External influence is measured in the state of a particular block. Its scale can change with depth as token representations evolve and earlier retirements alter the active sequence. We therefore compare candidates by their relative order within each observation rather than by applying one absolute influence threshold across the encoder. For an active candidate s, the normalized ascending rank is defined as:

$$r_\ell(s) = \frac{|\{u \in \mathcal{A}_\ell : \phi_\ell(u) < \phi_\ell(s)\}|}{\max(|\mathcal{A}_\ell| - 1, 1)}. \tag{6}$$

Lower values indicate weaker external influence. Equal influence values share the same minimum rank; the complete tie convention is given in Appendix B.2.

Retirement requires consistent evidence from two consecutive observations. For a token that remains active at block ℓ, let $r_{\ell-1}(s)$ denote the rank stored under the same identity from block $\ell - 1$. We define its confirmation score as:

$$c_\ell(s) = \max(r_{\ell-1}(s), r_\ell(s)) . \tag{7}$$

The maximum is low only when the token ranks near the bottom at both layers. This prevents a transient drop in influence at either observation from being treated as sufficient evidence for irreversible removal.

At decision t, the budget schedule determines a low-rank window $\theta_t \in (0, 1]$. Requiring a token to lie within that window at both observations defines the eligible set:

$$\mathcal{E}_t = \{s \in \mathcal{A}_{\ell_t} : r_{\ell_t-1}(s) \le \theta_t,\ r_{\ell_t}(s) \le \theta_t\} . \tag{8}$$

Let ν_t be the number of retirements still required at this decision after accounting for any quota deferred from earlier layers. We retire the eligible tokens with the smallest confirmation scores, and defer any unmet quota rather than relaxing the confirmation rule immediately. Accordingly, the retirement set is:

$$\mathcal{R}_t = \mathrm{Bottom}_{\min(\nu_t, |\mathcal{E}_t|)}\big(\mathcal{E}_t; c_{\ell_t}\big). \tag{9}$$

At the final decision, the remaining requirement is filled in ascending confirmation-score order so that every input reaches the same terminal budget M. This deadline rule controls the comparison budget; it is not part of the influence measurement.

Once $\mathcal{R}_t$ is removed, the next block receives the shortened sequence defined by Equation equation 1. Influence is then recomputed from that block's own attention, so a survivor is judged under the representations and source set that actually exist after earlier retirements. Appendix B.2 specifies the fixed-budget allocation, and Algorithm 1 gives the complete procedure.

4 EXPERIMENTS

Model and benchmarks. We evaluate LLaVA-1.5-7B with CLIP ViT-L/14-336 at 336×336 input resolution. Our benchmark suite covers general and compositional visual question answering, text-rich and scientific reasoning, hallucination, and broad multimodal perception through GQA (Hudson & Manning, 2019), VQAv2 (Goyal et al., 2017), TextVQA (Singh et al., 2019), SQA (Lu et al., 2022), POPE (Li et al., 2023), MME (Fu et al., 2023), MMBench and MMBench-CN (Liu et al., 2024), VizWiz (Gurari et al., 2018), and MM-Vet (Yu et al., 2024). We report the primary metric defined for each benchmark. Detailed descriptions of the benchmark splits and evaluation metrics are provided in Appendix C.

Baselines and protocol. We compare against representative training-free visual-token reduction methods: ToMe (Bolya et al., 2023), SparseVLM (Zhang et al., 2025b), PyramidDrop (PDrop) (Xing et al., 2025), VisionZip (Yang et al., 2025), FlowCut (Tong et al., 2025), PruneSID-Dyn (Fang et al., 2026), ApET (Ma et al., 2026), and SCTS (Li et al., 2026b). These methods reduce visual tokens inside the vision encoder, at its output, or within the language model. The efficiency analysis additionally compares against FastV (Chen et al., 2024).

Implementation details. Influence-Confirmed Token Retirement requires no training or parameter updates. We use a single retirement configuration across all benchmarks and token budgets. The retirement window, hyperparameters, benchmark metrics, and timing protocol are provided in Appendix D.1.

Table 1: Performance comparison on LLaVA-1.5-7B under different terminal visual-token budgets. Dense uses all 576 visual tokens, and Avg. denotes the mean benchmark-wise performance normalized to Dense.

Method	GQA	MMB	MMB_{CN}	MME	POPE	SQA	VQA_{V2}	VQA_{Text}	VizWiz	MM-Vet	Avg.
Dense Reference, 576 Tokens (100%)											
Vanilla	61.9	64.7	58.1	1862	86.3	69.5	78.5	58.2	50.2	31.1	100%
Retain 192 Tokens (↓ 66.7%)											
ToMe (ICLR23)	54.3	60.5	–	1563	72.4	65.2	68.0	52.1	–	27.9	88.6%
SparseVLM (ICML25)	57.6	62.5	53.7	1721	83.6	68.9	75.6	56.1	50.5	31.5	96.5%
PDrop (CVPR25)	57.3	62.9	56.8	1766	82.3	68.8	75.1	56.5	51.1	30.5	96.9%
VisionZip (CVPR25)	59.3	63.0	57.3	1778	85.2	68.7	76.6	57.3	51.8	31.7	98.6%
FlowCut (NeurIPS26)	60.1	63.2	**57.8**	**1836**	86.1	68.6	77.1	57.5	51.8	**34.3**	**100.2%**
PruneSID-Dyn (ICLR26)	**60.2**	**63.8**	–	1797	**87.1**	69.1	76.8	56.9	**55.5**	–	99.9%
ICTR (Ours)	60.0	63.0	57.5	1816	85.8	**69.2**	**77.3**	**57.8**	52.0	32.5	99.6%
Retain 128 Tokens (↓ 77.8%)											
ToMe (ICLR23)	52.4	53.3	–	1343	62.8	59.6	63.0	49.1	–	27.2	81.2%
SparseVLM (ICML25)	56.0	60.0	51.1	1696	80.5	67.1	73.8	54.9	51.4	30.0	93.9%
PDrop (CVPR25)	57.1	61.6	56.3	1664	82.3	68.3	72.9	56.6	51.0	30.8	95.8%
VisionZip (CVPR25)	57.6	62.0	56.2	1759	83.2	68.9	75.6	56.8	51.6	32.1	97.6%
FlowCut (NeurIPS26)	58.5	62.1	56.5	**1792**	85.2	68.6	76.0	57.3	52.2	32.5	98.5%
PruneSID-Dyn (ICLR26)	**58.9**	**62.6**	–	1760	**86.9**	68.8	75.4	55.1	**56.0**	–	98.5%
ICTR (Ours)	**58.9**	61.8	**56.8**	1776	85.9	**69.2**	**76.2**	**57.4**	52.1	**33.3**	**98.9%**
Retain 64 Tokens (↓ 88.9%)											
ToMe (ICLR23)	48.6	43.7	–	1138	52.5	50.0	57.1	45.3	–	24.1	71.0%
SparseVLM (ICML25)	52.7	56.2	46.1	1505	75.1	62.2	68.2	51.8	50.1	23.3	85.9%
PDrop (CVPR25)	47.5	58.8	50.5	1561	55.9	68.6	69.2	50.6	50.7	30.7	87.7%
VisionZip (CVPR25)	55.1	60.1	55.3	1687	77.0	69.0	72.4	55.5	52.8	31.5	95.0%
FlowCut (NeurIPS26)	55.6	60.8	**55.4**	**1744**	80.2	69.1	72.8	55.6	53.2	**32.3**	96.3%
PruneSID-Dyn (ICLR26)	57.2	59.7	–	1734	84.1	68.1	73.8	54.2	**57.0**	–	**96.7%**
ICTR (Ours)	**57.5**	**60.9**	54.5	1680	**85.7**	**69.8**	**74.0**	**56.0**	51.2	30.9	96.3%

4.1 Main results

Tables 1–3 summarize the results across three model architectures. On LLaVA-1.5-7B, ICTR remains competitive with the strongest baselines across all budgets and retains 96.3% of dense performance with only 64 visual tokens. Its robustness under tighter budgets supports our premise that, when capacity is limited, preserving tokens that remain influential to the evolving active set is preferable to relying on a static ranking. This trend is more pronounced on LLaVA-NeXT-7B: despite starting from 2,880 visual tokens, ICTR achieves the best normalized average at all three budgets, with a larger advantage at smaller terminal budgets. These results demonstrate that progressive retirement generalizes beyond the fixed-resolution LLaVA-1.5 setting to substantially longer visual sequences.

Qwen2.5-VL-7B exhibits a different pattern. Qwen's learned spatial merger requires each 2×2 patch group to be retired atomically, reducing the flexibility of per-token retirement and contributing to the larger performance gap on this backbone. Appendix D.2 provides the complete per-budget analysis and further discussion.

We present visualizations of the layer-wise retirement process in Appendix E. Notably, in most cases, the prescribed retirement budget is fully met within the designated layers.

4.2 Ablation study

We ablate three core designs of ICTR on LLaVA-1.5-7B: the complete source-removal effect, active-set re-evaluation, and cross-layer confirmation. All variants follow the same retirement schedule and terminal token budget.

Ablation on influence measurement. Replacing the complete source-removal effect with the attention coefficient alone leads to increasingly larger degradation as the budget tightens. At 64 tokens, the coefficient-only variant performs worse on all five benchmarks and loses 1.88 normalized points on average. This result shows that attention strength alone is insufficient under aggressive compression, while the value–output discrepancy provides additional information for identifying consequential sources.

Table 2: Performance comparison on LLaVA-NeXT-7B across terminal visual-token budgets. Avg. denotes the mean benchmark-wise performance normalized to Dense.

Method	GQA	MMB	POPE	VQA_{V2}	VQA_{Text}	Avg.
Dense Reference, 2880 Tokens (100%)						
Vanilla	**64.2**	**67.9**	**86.5**	**81.8**	**61.3**	**100%**
Retain 640 Tokens (↓ 77.8%)						
VisionZip	61.3	66.3	86.3	79.1	60.2	97.6%
PRUNESID	61.6	64.2	86.3	78.5	57.9	96.1%
ICTR	63.2	66.8	86.3	79.2	60.5	**98.4%**
Retain 320 Tokens (↓ 88.9%)						
VisionZip	59.3	63.1	82.1	76.2	58.9	93.9%
PRUNESID	60.5	63.0	83.1	76.6	54.4	93.1%
ICTR	62.3	65.1	85.8	76.4	59.6	**96.5%**
Retain 160 Tokens (↓ 94.4%)						
VisionZip	55.5	60.1	74.8	71.4	56.2	88.1%
PRUNESID	58.9	60.8	76.9	73.8	54.0	89.7%
ICTR	59.4	62.4	82.2	74.0	57.3	**92.7%**

Table 3: Performance comparison on Qwen2.5-VL-7B across visual-token retention ratios. Avg. denotes the mean benchmark-wise performance normalized to Dense.

Method	MMB	MME	POPE	SQA	VQA_{Text}	Avg.
Dense Reference (100%)						
Vanilla	82.8	2304	86.1	86.2	84.9	100%
Token Pruning Rate (↓ 66.7%)						
HoloV	78.3	2093	85.0	79.8	78.9	93.9%
SCTS	78.6	2317	86.4	85.5	77.3	**97.2%**
ICTR	79.3	2276	82.2	82.8	73.9	94.6%
Token Pruning Rate (↓ 77.8%)						
HoloV	76.5	2043	82.3	79.8	70.3	90.4%
SCTS	78.1	2218	85.9	84.0	75.5	**95.3%**
ICTR	78.2	2201	82.2	78.9	67.3	91.2%
Token Pruning Rate (↓ 88.9%)						
HoloV	72.4	2006	80.7	79.5	61.8	86.7%
SCTS	73.9	2026	81.5	82.9	71.0	**90.3%**
ICTR	73.5	2053	82.2	73.0	59.1	85.5%

Table 4: Component ablations on LLaVA-1.5-7B. *Coefficient only* removes the value–output discrepancy, *frozen scores* disables active-set re-evaluation, and *single observation* removes cross-layer confirmation.

Variant	GQA	MMB	MME	POPE	VQA_{Text}	Avg.
Retain 192 Tokens						
Complete ICTR	60.0	63.0	1816	85.8	57.8	**98.11**
Coefficient only	60.1	62.2	1788	85.8	57.6	97.53 (−0.58)
Frozen scores	59.8	58.8	1695	84.8	57.6	95.15 (−2.96)
Single observation	60.0	62.5	1800	85.8	57.6	97.72 (−0.40)
Retain 128 Tokens						
Complete ICTR	58.9	61.8	1776	85.9	57.4	**96.84**
Coefficient only	59.0	61.6	1771	85.0	56.4	96.21 (−0.64)
Frozen scores	58.8	57.9	1604	84.4	56.9	93.24 (−3.60)
Single observation	59.2	61.1	1767	85.5	57.0	96.40 (−0.45)
Retain 64 Tokens						
Complete ICTR	57.5	60.9	1680	85.7	56.0	**94.55**
Coefficient only	56.2	59.6	1652	84.5	54.6	92.67 (−1.88)
Frozen scores	56.1	56.4	1593	83.4	54.4	90.69 (−3.86)
Single observation	56.8	60.4	1667	85.4	55.6	93.83 (−0.73)

Ablation on active-set re-evaluation. Freezing the initial influence ordering causes the largest and most consistent degradation, reducing the normalized average by 2.96–3.86 points. As tokens retire, both the available source set and survivor representations evolve, making active-set re-evaluation necessary to keep retirement decisions aligned with the current computation. The resulting decision drift is visualized in Appendix D.3.

Ablation on cross-layer confirmation. Using a single observation preserves similar average performance but changes 18.0–32.6% of the final survivor identities. Cross-layer confirmation therefore mainly stabilizes the retirement trajectory by requiring a token to remain weak across successive blocks before irreversible removal.

Together, these ablations show that complete influence measurement identifies consequential sources, active-set re-evaluation tracks their changing roles, and cross-layer confirmation improves the consistency of retirement decisions.

We further examine whether the selected retired tokens indeed induce smaller downstream perturbations through a controlled ablation at layer 16. All branches share the same ICTR prefix, retire the same number of tokens using ICTR, current-layer influence scores, or random selection, and perform no further retirement. As shown in Fig. 3, both influence-based strategies produce lower

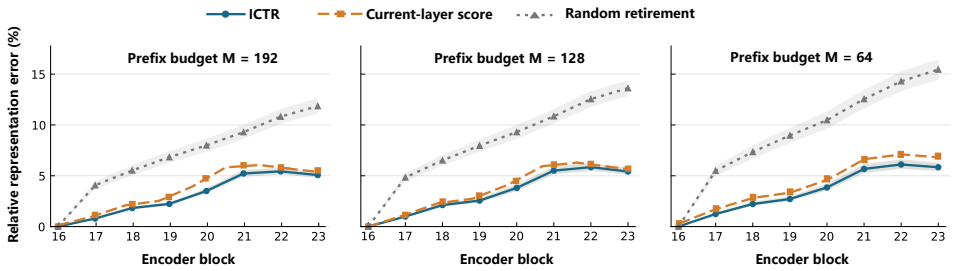

Figure 3: Effect of retirement selection on subsequent encoder representations. Relative representation error is measured against the no-removal continuation on tokens surviving all branches. Curves show means over 100 images, with five random trials averaged per image; shaded bands indicate pointwise 95% bootstrap confidence intervals.

subsequent representation errors than random retirement, indicating that the influence score identifies tokens whose removal causes less disruption to later representations. ICTR further yields a modest reduction in final error over using current-layer scores alone.

4.3 Efficiency analysis

The stage at which tokens are removed determines how much subsequent computation benefits from compression. Table 5 shows that ICTR reduces prefill time by 51.2% and the KV-cache footprint by 71.2% relative to Dense, yielding a $2.048\times$ speedup. Its controller increases vision-stage time by 15.3% over Dense, but progressive retirement remains 6.7% faster in this stage than output-stage VisionZip. Compared with FastV, making the compact sequence available before language-model processing reduces post-vision time by 13.8% and the KV-cache footprint by 13.4%. ICTR therefore achieves the lowest end-to-end prefill time, 6.6% below VisionZip and 7.3% below FastV, showing that its bounded control cost is recovered by reducing the computation that follows.

Table 5: Efficiency comparison at a budget of 128 visual tokens. Vision-time savings are measured relative to Dense; negative values indicate overhead.

Method	K	**Vision (s)**	**ΔVision (s)**	**After vision (s)**	**Prefill (s)**	**Speedup**	**KV (MiB)**
Dense	576	1.225	0.000	7.187	8.412	1.000×	314.5
FastV	128	1.302	-0.077	3.127	4.430	1.899×	104.5
VisionZip	128	1.513	-0.288	2.884	4.397	1.913×	**90.5**
ICTR	128	1.412	-0.187	**2.696**	**4.108**	**2.048**×	**90.5**

5 Conclusion

In this work, we investigate when visual tokens can leave the vision encoder without disrupting later computation. Controlled interventions show that preserving an intermediate state cannot replace a token's continued updates and interactions. ICTR therefore measures external influence, confirms low-influence candidates across layers, and progressively retires them without training or textual guidance. On LLaVA-1.5-7B, ICTR retains 96.3% of dense performance with only 64 of 576 visual tokens. It also transfers to LLaVA-NeXT and halves prefill time at the 128-token budget. Progressive retirement therefore recovers its control cost by reducing the visual and language-model computation that follows. These results show that tracking evolving influence enables reliable encoder-internal compression under tight token budgets.

AI USE STATEMENT

In this work, we used generative AI tools to edit and polish the manuscript for clarity, grammar, and readability, and to assist in identifying and fixing bugs in the software code used in this work. We did not use generative AI tools for tasks classified as requiring disclosure under the ICLR 2027 AI Policy for Authors.

All AI-assisted modifications to the manuscript were manually reviewed by the authors to ensure that they preserved the intended technical meaning and did not introduce unsupported claims or factual inaccuracies. AI-assisted code modifications were manually inspected and tested by the authors for correctness, and the relevant experiments were rerun to verify the resulting implementation and results. We take full responsibility for the final content of this work, including all text, claims, code, results, and other artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

Code for reproducing the experiments is provided in the supplementary material. The technical appendices provide the source-removal derivation, fixed-budget procedure, benchmark protocols, and experimental configuration.

REFERENCES

Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your ViT but faster. In *International Conference on Learning Representations*, 2023.

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 *Computer Vision – ECCV 2024*, pp. 19–35. Springer, 2024.

Yuan Chen, Zichen Wen, Yuzhou Wu, Xuyang Liu, Shuang Chen, Junpeng Ma, Weijia Li, Conghui He, and Linfeng Zhang. IPCV: Information-preserving compression for MLLM visual encoders. *arXiv preprint arXiv:2512.18747*, 2025.

Yingqi Fan, Anhao Zhao, Jinlan Fu, Junlong Tong, Hui Su, Yijie Pan, Wei Zhang, and Xiaoyu Shen. VisiPruner: Decoding discontinuous cross-modal dynamics for efficient multimodal LLMs. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 18885–18902. Association for Computational Linguistics, 2025.

Zhengyao Fang, Pengyuan Lyu, Chengquan Zhang, Guangming Lu, Jun Yu, and Wenjie Pei. Prune redundancy, preserve essence: Vision token compression in VLMs via synergistic importance–diversity. In *International Conference on Learning Representations*, 2026.

Mohsen Fayyaz, Soroush Abbasi Koohpayegani, Farnoush Rezaei Jafari, Sunando Sengupta, Hamid Reza Vaezi Joze, Eric Sommerlade, Hamed Pirsiavash, and Jürgen Gall. Adaptive token sampling for efficient vision transformers. In *Computer Vision – ECCV 2022*, pp. 396–414. Springer, 2022.

Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, Yunsheng Wu, and Rongrong Ji. MME: A comprehensive evaluation benchmark for multimodal large language models. *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.

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/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6904–6913, 2017.

Danna Gurari, Qing Li, Abigale J. Stangl, Anhong Guo, Chi Lin, Kristen Grauman, Jiebo Luo, and Jeffrey P. Bigham. VizWiz grand challenge: Answering visual questions from blind people. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3608–3617, 2018.

Yuhang Han, Xuyang Liu, Zihan Zhang, Pengxiang Ding, Junjie Chen, Honggang Chen, Donglin Wang, Qingsen Yan, and Siteng Huang. Filter, correlate, compress: Training-free token reduction for MLLM acceleration. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 4601–4609, 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.

Ahmadreza Jeddi, Negin Baghbanzadeh, Elham Dolatabadi, and Babak Taati. Similarity-aware token pruning: Your VLM but faster. *arXiv preprint arXiv:2503.11549*, 2025.

Dong-Jae Lee, Sunghyun Baek, and Junmo Kim. IWP: Token pruning as implicit weight pruning in large vision language models. *arXiv preprint arXiv:2604.00757*, 2026.

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.

Xueting Li, Qi Liu, Chenghao Xu, Xu Yang, Guangtao Lyu, Jiahua Li, and Cheng Deng. Semantically comprehensive token pruning in lvlms via maximizing concept coverage. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 27829–27846. Association for Computational Linguistics, 2026b.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne 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. Association for Computational Linguistics, 2023.

Youwei Liang, Chongjian Ge, Zhan Tong, Yibing Song, Jue Wang, and Pengtao Xie. EViT: Expediting vision transformers via token reorganizations. In *International Conference on Learning Representations*, 2022.

Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, Kai Chen, and Dahua Lin. MMBench: Is your multi-modal model an all-around player? In *Computer Vision – ECCV 2024*, pp. 216–233. Springer, 2024.

Hongyu Lu, Feng Zhang, Wenwei Jin, Huanling Hu, Pengfei Zhang, Yao Hu, Jiawei Li, and Shikai Jiang. EvoCut: Multi-layer evolution-aware visual token compression for efficient large vision-language models. *arXiv preprint arXiv:2606.01756*, 2026.

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

Qiankun Ma, Ziyao Zhang, Haofei Wang, Zhen Song, Jie Chen, and Hairong Zheng. ApET: Approximation-error guided token compression for efficient VLMs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 26306–26316, 2026.

Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. DynamicViT: Efficient vision transformers with dynamic token sparsification. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

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.

Aditya Kumar Singh, Hitesh Kandala, Pratik Prabhanjan Brahma, Zicheng Liu, and Emad Barsoum. DUET-VLM: Dual stage unified efficient token reduction for VLM training and inference. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17651–17660, 2026.

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.

Jintao Tong, Wenwei Jin, Pengda Qin, Anqi Li, Yixiong Zou, Yuhong Li, Yuhua Li, and Ruixuan Li. FlowCut: Rethinking redundancy via information flow for efficient vision-language models. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Hesong Wang, Xin Jin, Lu Lu, Chenhaowen Li, Jian Chen, Qiang Liu, and Huan Wang. EarlyTom: Early token compression completes fast video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 40559–40568, 2026a.

Ziyang Wang, Mengwei Li, Hao Yin, Wenhao Liu, and Zilei Wang. PosPrune: Visual token pruning with positional bias correction for efficient large vision-language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 10485–10493, 2026b.

Long Xing, Qidong Huang, Xiaoyi Dong, Jiajie Lu, Pan Zhang, Yuhang Zang, Yuhang Cao, Conghui He, Jiaqi Wang, Feng Wu, and Dahua Lin. Conical visual concentration for efficient large vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14593–14603, 2025.

Rui Xu, Yunke Wang, Yong Luo, and Bo Du. Rethinking visual token reduction in LVLMs under cross-modal misalignment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 27323–27331, 2026a.

Tong Xu, Hailong Shi, and Xingyu Gao. SCoRe: Salience-coverage reduction for vision token pruning in vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24686–24695, 2026b.

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

Xubing Ye, Yukang Gan, Yixiao Ge, Xiao-Ping Zhang, and Yansong Tang. ATP-LLaVA: Adaptive token pruning for large vision language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24972–24982, 2025.

Hongxu Yin, Arash Vahdat, Jose M. Alvarez, Arun Mallya, Jan Kautz, and Pavlo Molchanov. A-ViT: Adaptive tokens for efficient vision transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10809–10818, 2022.

Weihao Yu, Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Zicheng Liu, Xinchao Wang, and Lijuan Wang. MM-vet: Evaluating large multimodal models for integrated capabilities. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 57730–57754. PMLR, 2024.

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

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, 2025a.

Yuan Zhang, Chun-Kai Fan, Junpeng Ma, Wenzhao Zheng, Tao Huang, Kuan Cheng, Denis A. Gudovskiy, Tomoyuki Okuno, Yohei Nakata, Kurt Keutzer, and Shanghang Zhang. SparseVLM: Visual token sparsification for efficient vision-language model inference. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 74840–74857. PMLR, 2025b.

CONTENTS OF TECHNICAL APPENDICES

A Motivation analysis **15**

B Method Details **15**

B.1 Source-removal identity and influence evaluation 15

B.2 Ranks and fixed-budget allocation 16

B.3 Complete procedure 16

C Benchmarks and Metrics **16**

D Experiment **18**

D.1 Retirement configuration 18

D.2 Detailed Analysis of Main Results 18

D.3 Additional ablation analysis 19

E More Visualization Results **19**

A MOTIVATION ANALYSIS

We complement the task results with a matched intervention study on all 5,000 images in the COCO 2017 validation set. Figure 4 shows that applying the same state or interaction intervention at progressively deeper visual blocks produces increasingly smaller changes in the final visual representation. The observed depth dependence motivates treating retirement as an evolving decision: the consequence of removing a token from continued participation changes as the encoder states evolve.

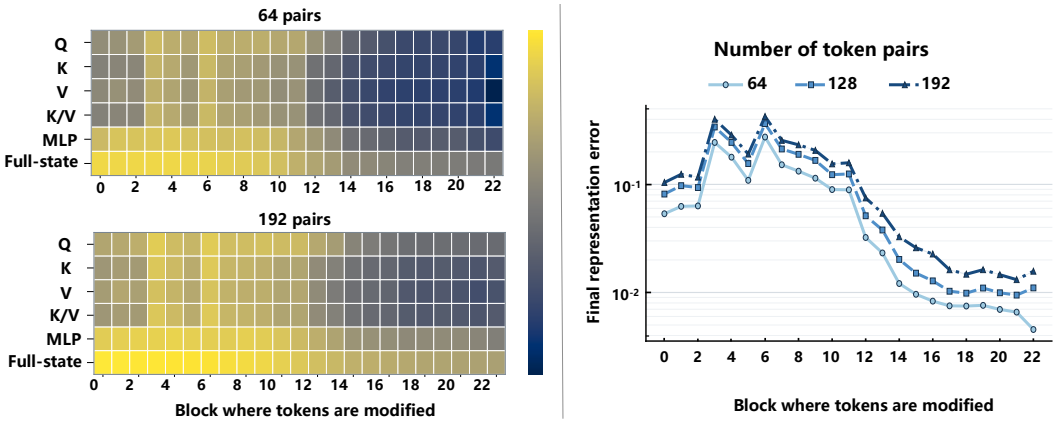

Figure 4: Layerwise final-representation error under matched token-pair interventions on COCO val2017. (a) Relative final-representation error when 64 or 192 token pairs share Q, K, V, K/V, MLP output, or full token state at one visual block. (b) Final-representation error from K/V compaction with 64, 128, or 192 pairs across visual blocks.

B METHOD DETAILS

B.1 SOURCE-REMOVAL IDENTITY AND INFLUENCE EVALUATION

For a visible source $s \in \mathcal{K}_\ell(i)$, excluding its key and value leaves the logits of the remaining sources unchanged. Their new attention weights are therefore $\alpha_{\ell,h,i,j}/(1-\alpha_{\ell,h,i,s})$ for $j \neq s$. When the remaining visible set is nonempty, the output after excluding s can be written as:

$$\begin{aligned}\mathbf{y}_{\ell,h,i}^{(-s)} &= \sum_{j \in \mathcal{K}_\ell(i)\setminus\{s\}} \frac{\alpha_{\ell,h,i,j}}{1-\alpha_{\ell,h,i,s}} \mathbf{v}_{\ell,h,j} \\ &= \frac{\mathbf{y}_{\ell,h,i} - \alpha_{\ell,h,i,s}\mathbf{v}_{\ell,h,s}}{1-\alpha_{\ell,h,i,s}}.\end{aligned} \tag{10}$$

Subtracting this expression from $\mathbf{y}_{\ell,h,i}$ gives Equation equation 3. An invisible source is assigned zero effect. The identity concerns the current attention head output; subsequent state evolution is assessed through fresh observations in later blocks.

For numerical stability, the denominator introduced by source removal is clamped at ε. The relative effect in Equation equation 4 is therefore evaluated from the attention weights, values, and head outputs as:

$$\bar{\delta}_{\ell,i\leftarrow s} = \frac{1}{H}\sum_{h=1}^{H} \frac{\alpha_{\ell,h,i,s}}{\max(1-\alpha_{\ell,h,i,s},\varepsilon)} \frac{\|\mathbf{v}_{\ell,h,s} - \mathbf{y}_{\ell,h,i}\|_2}{\|\mathbf{y}_{\ell,h,i}\|_2 + \varepsilon}. \tag{11}$$

These calculations use float32 arithmetic. Pairwise distances are computed from squared norms and inner products, with negative squared distances caused by round-off clipped to zero. Candidate sources are processed in chunks to limit intermediate memory. Head averaging precedes recipient selection, and each source's self-recipient is excluded before the largest relative effects are aggregated. The attention quantities come from the block's normal attention computation; retirement is applied only after that block, including its residual and feed-forward updates, has finished.

B.2 Ranks and fixed-budget allocation

Rank convention. Equation equation 6 assigns the minimum ascending rank to equal influence values. The previous-layer rank of a surviving token is retrieved by its original identity; ranks stored at that observation are not recomputed after the subsequent retirement changes the active set.

Decision window. Let L be the number of executed visual blocks, b_{o} the first observation block, and b_{d} the final retirement block, with $1 \le b_{\mathrm{o}} < b_{\mathrm{d}} \le L$. Block b_{o} initializes the rank history. The following $T = b_{\mathrm{d}} - b_{\mathrm{o}}$ blocks provide decisions at $\ell_t = b_{\mathrm{o}} + t$, for $t = 1, \dots, T$. Before this window and after its deadline, the encoder runs without influence scoring or further retirement. Window placement is an encoder-specific experimental setting, not a fixed depth prescribed by the strategy.

Integer allocation. For an initial active visual-token count $N = |\mathcal{A}_1|$ and terminal count $1 \le M < N$, a total of $G = N - M$ tokens must retire. We allocate this count across the T decisions using linearly increasing weights. For $T > 1$, the weight assigned to decision t is defined as:

$$\omega_t = 1 - \lambda + 2\lambda \frac{t-1}{T-1}, \qquad t = 1, \dots, T, \tag{12}$$

where $\lambda \in [0, 1)$ controls the slope; for $T = 1$, set $\omega_1 = 1$. Let P_t be the planned cumulative retirement count and g_t the planned increment. These quantities are computed as:

$$P_t = \mathrm{round}\left(G \frac{\sum_{a=1}^{t} \omega_a}{\sum_{a=1}^{T} \omega_a} \right), \qquad g_t = P_t - P_{t-1}, \qquad P_0 = 0. \tag{13}$$

Rounding is to the nearest integer, with half-integer ties rounded to the nearest even integer, and P_T is set to G. Rounding cumulative counts keeps the increments consistent with the terminal budget.

Let $\widehat{P}_{t-1}$ denote the number of tokens actually retired before decision t, with $\widehat{P}_0 = 0$. The outstanding quota ν_t and low-rank threshold θ_t are computed as:

$$\nu_t = \max(0, P_t - \widehat{P}_{t-1}), \qquad \theta_t = \min\left(1, \frac{\kappa g_t}{n_{\ell_t}}\right), \tag{14}$$

where $\kappa > 0$ controls the low-rank candidate window. The quota includes deferred retirements, whereas the threshold uses the planned increment g_t and current active visual-token count $n_{\ell_t} = |\mathcal{A}_{\ell_t}|$. At the deadline, confirmed candidates are selected first and any remaining requirement is filled in ascending confirmation-score order. This allocation fixes the comparison budget; the influence measurement and consecutive-layer confirmation determine which tokens are selected before the deadline rule is applied.

B.3 Complete procedure

Algorithm 1 describes the procedure for one input sequence. The operator $\mathrm{Bottom}_z(\mathcal{S}; f)$ returns the z elements of $\mathcal{S}$ with the smallest values of f. Ties are resolved by current sequence order, and $z = 0$ returns the empty set. Token identities are preserved when gathering surviving states.

C Benchmarks and Metrics

We evaluate on ten benchmark variants using the data preparation and prompt formats of LLaVA-1.5. Table 6 summarizes their splits and primary metrics.

Open-ended visual question answering. GQA evaluates compositional reasoning over objects, attributes, and relations; we use its 12,578-question balanced test-dev split (Hudson & Manning, 2019). VQAv2 evaluates general visual question answering on test-dev2015 (Goyal et al., 2017). TextVQA evaluates reading and reasoning over text in natural images; we use its 5,000-question v0.5.1 validation split (Singh et al., 2019). VizWiz evaluates visual question answering in an assistive setting; we use its 8,000-question test split (Gurari et al., 2018).

Algorithm 1 Influence-Confirmed Token Retirement

Require: Initial token states, active visual identities $\mathcal{A}_1$, L visual blocks, budget M, window $[b_{\mathrm{o}}, b_{\mathrm{d}}]$, and $\eta, \kappa, \lambda, \varepsilon$

$N \leftarrow |\mathcal{A}_1|$, $G \leftarrow N - M$, $T \leftarrow b_{\mathrm{d}} - b_{\mathrm{o}}$, $\widehat{P}_0 \leftarrow 0$
Construct $\{P_t, g_t\}_{t=1}^{T}$ using equation 12–equation 13
for $\ell = 1, \ldots, L$ **do**
  Run block ℓ on the current active states
  if $b_{\mathrm{o}} \leq \ell \leq b_{\mathrm{d}}$ **then**
    Compute fresh ϕ_ℓ from this block's attention using equation 11 and equation 5
    Compute r_ℓ using equation 6; store ranks by token identity
  end if
  if $b_{\mathrm{o}} < \ell \leq b_{\mathrm{d}}$ **then**
    $t \leftarrow \ell - b_{\mathrm{o}}$, $\ell_t \leftarrow \ell$
    Compute ν_t and θ_t using equation 14
    Compute c_ℓ and $\mathcal{E}_t$ using equation 7 and equation 8
    $\mathcal{R}_t \leftarrow \mathrm{Bottom}_{\min(\nu_t, |\mathcal{E}_t|)}(\mathcal{E}_t; c_\ell)$
    if $t = T$ **then**
      $\xi \leftarrow G - \widehat{P}_{t-1} - |\mathcal{R}_t|$
      $\mathcal{R}_t \leftarrow \mathcal{R}_t \cup \mathrm{Bottom}_{\xi}(\mathcal{A}_\ell \setminus \mathcal{R}_t; c_\ell)$
    end if
    $\widehat{P}_t \leftarrow \widehat{P}_{t-1} + |\mathcal{R}_t|$
    Update $\mathcal{A}_{\ell+1}$ using equation 1
    Gather surviving states in their original order
  else
    $\mathcal{A}_{\ell+1} \leftarrow \mathcal{A}_\ell$
  end if
end for
return Surviving states through the encoder's output interface

Table 6: Benchmarks, evaluation splits, and primary metrics.

Benchmark	Split or version	Evaluation focus	Primary metric
GQA	testdev-balanced	Compositional visual reasoning	GQA accuracy
MMBench	English DEV 20230712	Broad multimodal multiple choice	CircularEval accuracy
MMBench-CN	Chinese DEV 20231003	Chinese multimodal multiple choice	CircularEval accuracy
MME	Official 14-task set	Perception and cognition	Total MME score
POPE	COCO random/popular/adversarial	Object hallucination	Macro F1
SQA	ScienceQA test (4,241 questions)	Scientific knowledge and reasoning	Accuracy
VQAv2	test-dev2015	General visual question answering	VQA accuracy
TextVQA	v0.5.1 validation	Reading and reasoning over scene text	VQA accuracy
VizWiz	Test	Assistive, real-world visual QA	VQA accuracy
MM-Vet	v1	Integrated vision-language capabilities	MM-Vet score

Multiple-choice and broad-coverage evaluation. MMBench evaluates fine-grained perception and reasoning with English and Chinese multiple-choice questions and uses CircularEval to account for option order. We use the English DEV 20230712 set with 1,176 questions and the Chinese DEV 20231003 set with 1,164 questions (Liu et al., 2024). SQA evaluates scientific knowledge and reasoning across natural, social, and language science. Following the LLaVA-1.5 protocol, we report accuracy on all 4,241 ScienceQA test questions, including 2,017 image questions and 2,224 text-only questions (Lu et al., 2022).

Specialized diagnostics. MME contains 2,374 yes/no questions across ten perception tasks and four cognition tasks (Fu et al., 2023). POPE evaluates object hallucination with 8,910 COCO-based questions constructed by random, popular, and adversarial sampling (Li et al., 2023). MM-Vet v1 contains 218 open-ended questions that combine recognition, OCR, knowledge, spatial reasoning, mathematics, and language generation (Yu et al., 2024).

Table 7: Backbone-specific retirement configurations. All block indices are one-based.

Setting	LLaVA-1.5-7B	Qwen2.5-VL-7B
Vision encoder	CLIP ViT-L/14-336	Qwen2.5-VL ViT
Executed visual blocks L	23	32
Retirement candidate	Patch token	2×2 patch group
Initial candidate count N	576	Image-dependent (256–1280)
Observation block b_{o}	11	15
Final retirement block b_{d}	21	29
Terminal candidate budget M	$\{64, 128, 192\}$	$\{1/9, 2/9, 1/3\}N$
Recipient fraction η	0.10	0.10
Candidate-window factor κ	2	2
Allocation slope λ	0.5	0.5
Candidate-source chunk size	32	32 groups
Numerical constant ε	10^{-6}	10^{-6}

D EXPERIMENT

D.1 RETIREMENT CONFIGURATION

Table 7 records the configuration used for the CLIP-based experiments. All block indices are one-based. The first observation is made in block 11, retirement decisions follow blocks 12–21, and the target token count is reached after block 21. For Qwen2.5-VL, the corresponding blocks are 15, 16–29, and 29. Within each backbone, the configuration is shared across all benchmarks and retention rates. This configuration is shared across all benchmarks and token budgets.

The CLIP class token remains active and is included among the recipients used to measure patch-source influence, but it is not a retirement candidate. The encoder supplies patch features after its 23rd block to the original downstream projection module.

Qwen2.5-VL uses a learned spatial merger that maps each 2×2 group of patch tokens to one visual token. Removing an individual patch would break the pretrained merger input structure. We therefore treat each merger group as one atomic retirement candidate: its four patch sources are scored and retired jointly, and each surviving group is passed intact to the learned merger. Because Qwen2.5-VL uses dynamic image resolution, its initial candidate count is image-dependent; the three terminal budgets retain $1/3$, $2/9$, or $1/9$ of the original merger groups.

D.2 DETAILED ANALYSIS OF MAIN RESULTS

Table 1 compares all methods under matched terminal visual-token budgets on LLaVA-1.5-7B. ICTR retains 99.6%, 98.9%, and 96.3% of the dense-model average at 192, 128, and 64 tokens, respectively. It remains competitive with the strongest baselines at the two larger budgets, achieves the highest average at 128 tokens, and matches FlowCut at 64 tokens while retaining only 11.1% of the original visual sequence. The limited degradation under increasingly severe compression is consistent with our central motivation: when retained capacity is ample, several criteria can preserve the dominant visual content; under a tight budget, preserving the few tokens on which the evolving computation still depends becomes decisive.

The same behavior extends to the higher-resolution LLaVA-NeXT-7B setting in Table 2. Starting from 2,880 visual tokens, ICTR retains 98.4%, 96.5%, and 92.7% of the dense-model average at budgets of 640, 320, and 160 tokens, respectively, outperforming both VisionZip and PRUNESID at every budget. Its margin over the strongest competing result increases from 0.8 percentage points at 640 tokens to 3.0 points at 160 tokens. These results show that progressive, influence-confirmed retirement transfers beyond the fixed-resolution LLaVA-1.5 encoder and remains effective when the initial visual sequence is substantially longer.

Qwen2.5-VL-7B presents a different architectural regime (Table 3). ICTR retains 94.6%, 91.2%, and 85.5% of the dense-model average across the three retention ratios. It outperforms HoloV at the first two ratios but trails SCTS, with the gap increasing under stronger compression. This difference is closely related to where and how compression is applied. HoloV and SCTS select tokens only after dense visual encoding, whereas ICTR retires sources while their representations are still evolving, thereby reducing computation within the remaining vision-encoder blocks.

Qwen2.5-VL further imposes a granularity constraint through its learned spatial merger: each 2×2 patch group must be retired as a single atomic candidate (Table 7). This coupling prevents the independent per-token retirement used on CLIP-based encoders and reduces the flexibility of influence-based selection. Consequently, the measured influence of an individual patch cannot always be translated into an equivalent token-level retirement decision, which likely contributes to the larger performance gap on this backbone.

This distinction makes the Qwen comparison substantially more demanding than a conventional post-encoder selection problem. Matching the reported retention ratio equates only the final token budget; it does not equate the amount of dense computation completed before compression. HoloV and SCTS preserve the full visual-encoder path and select from completed features. ICTR must instead decide whether a source can be removed before its representation has finished evolving. Each irreversible decision changes the attention source set seen by all subsequent visual blocks, so an early error propagates through the remaining encoder computation. In return, retirement shortens both the remaining vision-encoder sequence and the visual sequence processed downstream. Qwen's atomic merger-group constraint further restricts this already harder decision problem by replacing four independent patch decisions with one joint decision. The Qwen results therefore characterize the accuracy–efficiency trade-off of encoder-internal compression under merger compatibility, rather than an isolated ranking of post-encoder token-selection quality.

D.3 Additional ablation analysis

Figure 5 traces the retirement trajectories at the 64-token budget. After retirement begins, the survivor set produced by frozen scoring rapidly diverges from that of complete ICTR, whereas the variants that continue to evaluate the current active sequence remain substantially closer. The growing discrepancy shows how an early ranking mismatch propagates through subsequent irreversible decisions.

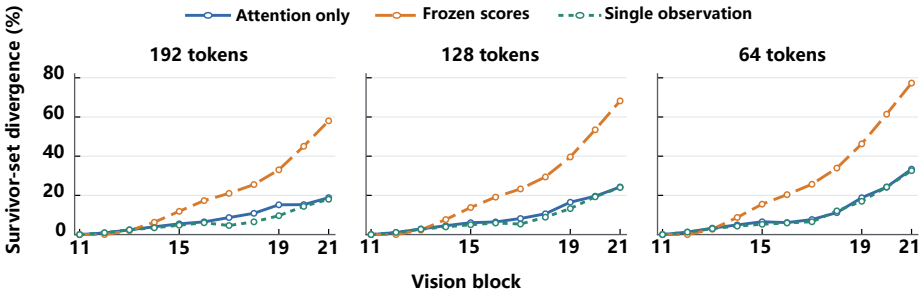

Figure 5: Evolution of survivor-set divergence from complete ICTR at budgets of 192, 128, and 64 tokens, averaged over 398 GQA images. Frozen scoring rapidly diverges after retirement begins, revealing the compounding effect of reusing an influence ordering computed before the active sequence changed.

E More Visualization Results

We visualize the tokens retained at each layer throughout the progressive retirement process, as shown in Fig 6 and Fig 7.

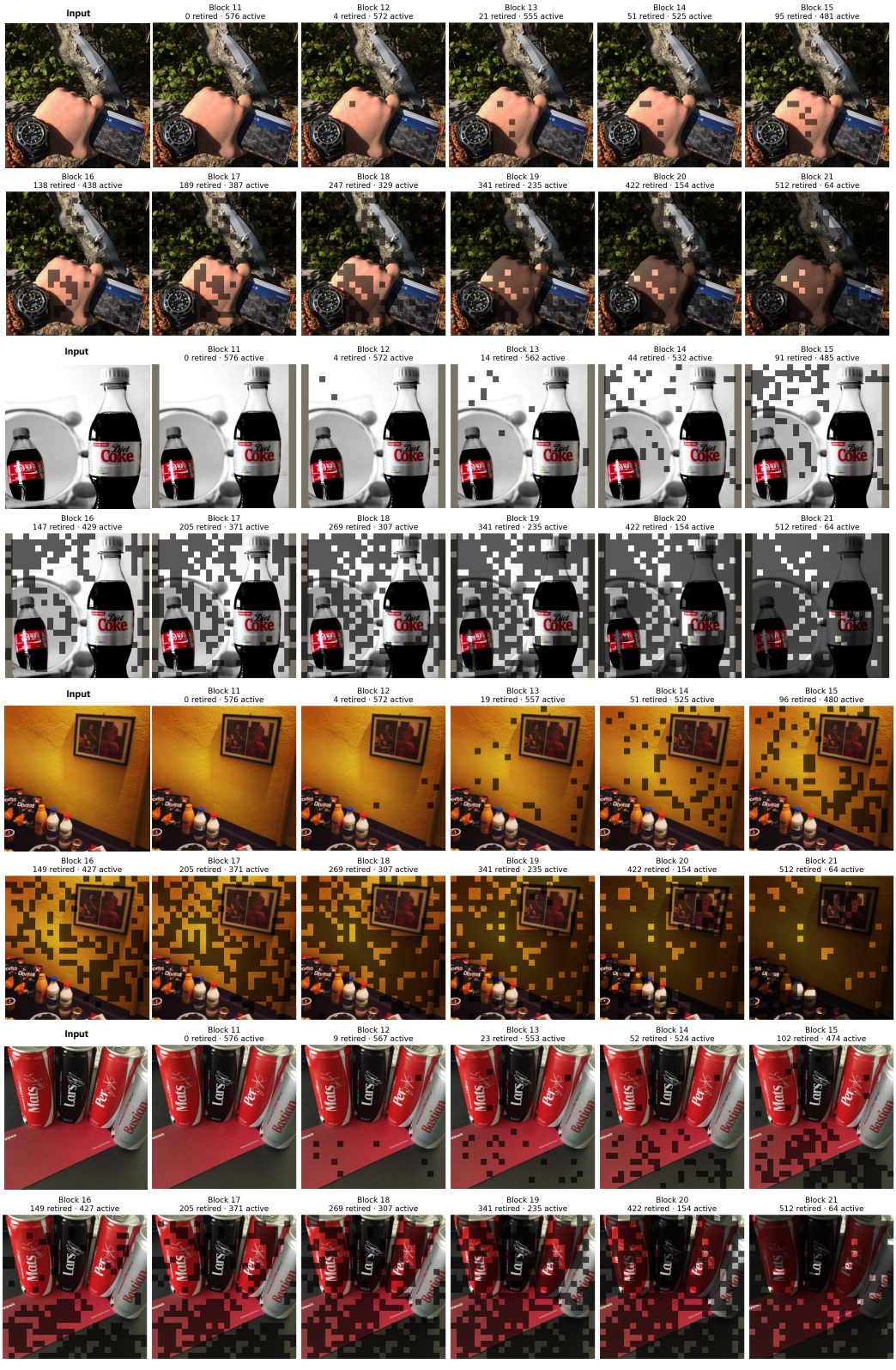

Figure 6: Additional qualitative examples of progressive token retirement at the 64-token budget. Darkened patches denote retired tokens, and the text above each panel reports the cumulative numbers of retired and active tokens after the corresponding vision block.

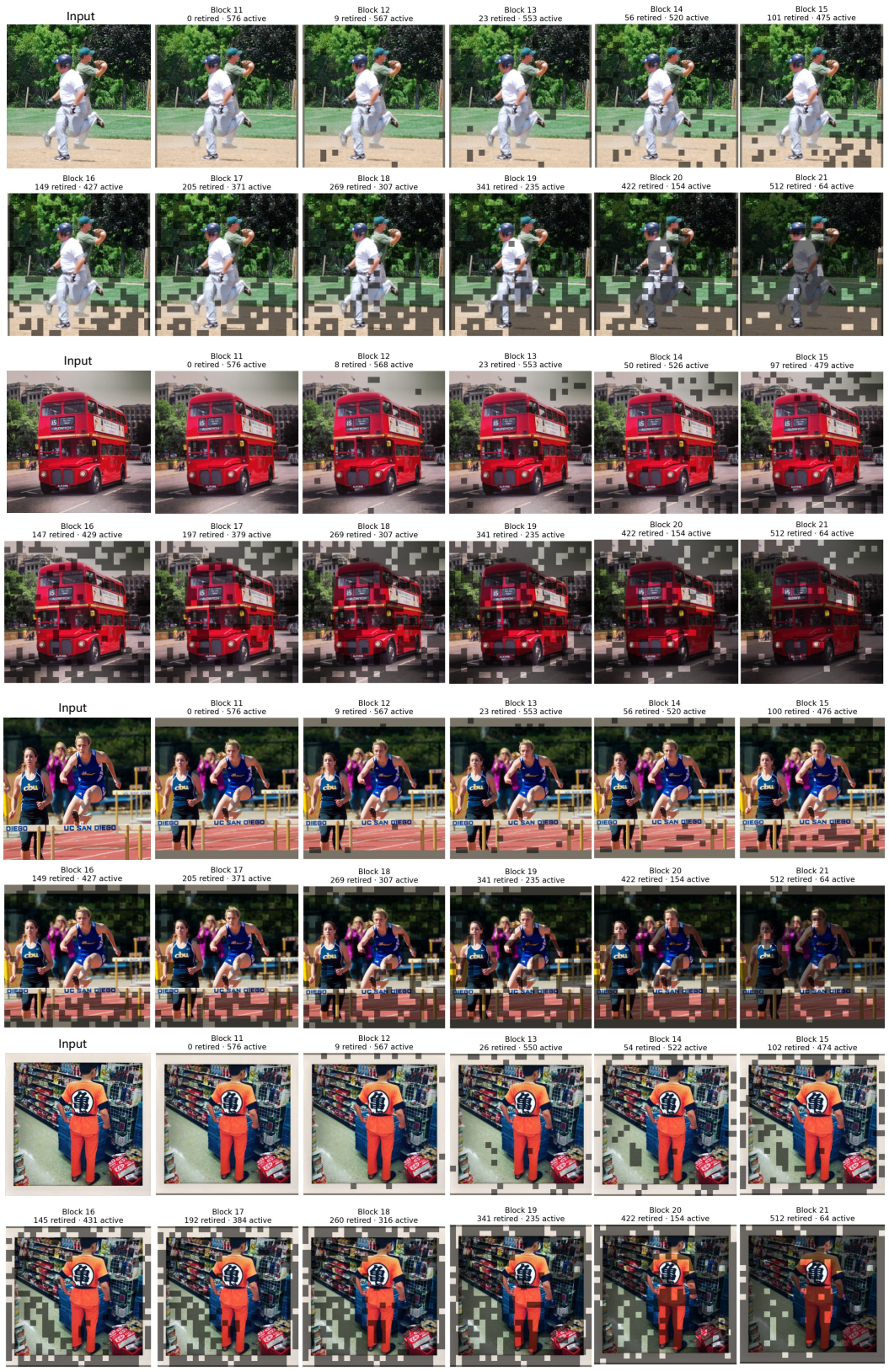

Figure 7: Additional qualitative examples of progressive token retirement at the 64-token budget. Darkened patches denote retired tokens, and the text above each panel reports the cumulative numbers of retired and active tokens after the corresponding vision block.