DO ALL VISUAL TOKENS MATTER EQUALLY? OBJECT-EVIDENCE PRESERVING TOKEN MERGING FOR VISION-LANGUAGE RETRIEVAL

Anonymous authors
Paper under double-blind review

ABSTRACT

Multi-vector vision-language retrieval preserves fine-grained visual evidence through maximum-similarity late interaction, but dense image-side tokens make storage and scoring expensive. Existing token compression methods reduce this cost, yet they can remove or collapse object- and region-level evidence that future query tokens may need to select. We propose SaMer, an object-aware token merging framework that compresses image-side post-projector tokens into K representative centroids while preserving the original late-interaction interface. SaMer uses object annotations only during training as a merge prior to discourage cross-instance mixing, requires no ground-truth bounding boxes or detectors at inference time, and adapts only the shared projection layer with frozen vision and language backbones. With $K = 64$, SaMer removes more than 93% of image-side tokens and reduces ColPali storage by $16.09\times$, while improving Recall@1 on Flickr30K and MSCOCO. These gains arise because object-aware merging preserves query-selectable object evidence that pruning or feature-only pooling can remove or collapse. SaMer also outperforms compression baselines and shows stronger phrase-level grounding, suggesting that efficient multi-vector retrieval depends not only on reducing token count, but on preserving the evidence future query tokens need to select.

1 INTRODUCTION

Efficient multi-vector vision-language retrieval requires preserving the object-, attribute-, and relation-level evidence that query tokens can select under late interaction. Multi-vector retrieval has been widely adopted in text retrieval since ColBERT (Khattab & Zaharia, 2020; Santhanam et al., 2022b), which performs maximum-similarity (MaxSim) matching between contextualized query and document token embeddings. Recent vision-language retrievers such as ColPali (Faysse et al., 2024) extend this paradigm to visual inputs by storing image-side patch embeddings and comparing them with query tokens through MaxSim. Similarly, ColQwen2 (Faysse et al., 2024) builds on Qwen2-VL (Wang et al., 2024), whose visual tokenization supports flexible image representation. These models use MaxSim to compare each query token with all image-side tokens and select the most similar visual token as evidence, making retrieval sensitive to objects, attributes, and relations rather than only global image-text similarity (Li et al., 2019; Diao et al., 2021).

However, preserving token-level image representations makes retrieval expensive in storage and scoring. A multi-vector retriever stores hundreds to over a thousand image-side token embeddings per image, and retrieval needs MaxSim comparisons between query tokens and all stored image tokens. As the index grows, both memory footprint and scoring latency rise rapidly, limiting large-scale image search and retrieval-augmented visual question answering. This is especially pronounced in patch-based visual representations, where each image yields many local tokens stored and compared during MaxSim scoring, making patch-level embeddings a major bottleneck for multi-vector VLM retrievers (Dosovitskiy et al., 2021; Tang et al., 2022; Ma et al., 2025).

One strategy is to compress visual tokens through pruning or merging (Ryoo et al., 2021; Rao et al., 2021; Liang et al., 2022; Marin et al., 2023; Bolya et al., 2023). Recent multimodal acceleration methods reduce visual tokens for large vision-language models (Cao et al., 2023; Shang et al.,

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.

2025). Yet retrieval compression differs from token reduction because patch importance is query-dependent. For the same image, one query may depend on a small object, another on an attribute, and a third on a relation. If compression removes or incorrectly merges phrase-relevant evidence, late interaction can no longer recover the token supporting the match. This risk is acute under feature-based merging, where similar patches from different instances of the same category can collapse into one representation, losing the distinction a query depends on (Lee & Hong, 2024; Huang et al., 2025). For multi-vector retrieval, the goal is not simply to reduce redundant tokens, but to preserve object- and region-level evidence in a form that remains selectable under future MaxSim queries.

To this end, we propose **S**emantic-**a**ware **Mer**ging (SaMer), an object-aware token merging framework preserving object- and region-level semantic evidence for future MaxSim queries. SaMer compresses post-projector visual tokens into K merged tokens using feature-spatial soft assignment, forming each representative as a normalized weighted centroid while keeping the late-interaction interface. During training, object annotations serve not as an auxiliary grounding loss, but as a merge prior guiding which visual tokens should be grouped. This discourages cross-instance merging, helping compressed representatives retain object-level evidence for later MaxSim retrieval. At inference, SaMer needs no ground-truth bounding boxes (bbox) or detector, performing annotation-free feature-spatial merging while remaining pluggable into multi-vector retrievers. Finally, SaMer uses projection-only adaptation, freezing the vision encoder and language backbone while training only the shared projection layer to keep merged tokens compatible with MaxSim scoring.

Since SaMer is designed to preserve object-level structure under compression, retrieval accuracy alone may not reveal whether compressed tokens retain the phrase-level evidence used by late interaction. We thus complement retrieval evaluation with grounding-oriented metrics measuring whether query-token relevance stays concentrated in annotated phrase regions. With $K = 64$, SaMer removes over 93% of image-side tokens, reduces ColPali image-side storage by $16.09\times$, and improves Recall@1 from 77.0 to 82.4 on Flickr30K (Young et al., 2014) and from 47.4 to 51.6 on MSCOCO (Lin et al., 2014). Grounding results also show SaMer better preserves phrase-level visual evidence than pruning- and pooling-based compression baselines at the same token budget. These results support our view that the effectiveness of retrieval compression depends on whether object evidence stays accessible to query tokens after compression. Our contributions are:

- We frame visual-token compression for late-interaction retrieval as an evidence-preservation problem and identify object-instance collapse as a key failure mode of feature-based merging.
- We propose SaMer, an object-aware token merging method that uses training-time object annotations as a merge prior to discourage cross-instance merging, without requiring annotations at inference time.
- We show that SaMer at $K = 64$ removes over 93% of visual tokens and reduces ColPali storage by $16.09\times$, while outperforming pruning- and pooling-based compression baselines under matched budget and adaptation with stronger phrase-level grounding.

2 Related Works

2.1 Vision-language retrieval and late interaction

Vision-language retrieval has been widely studied through contrastive image-text representation learning and multimodal pretraining, including ALIGN (Jia et al., 2021), ALBEF (Li et al., 2021), BLIP (Li et al., 2022), and BLIP-2 (Li et al., 2023). While these models provide strong image-text representations, they can underrepresent local evidence when retrieval depends on fine-grained objects, attributes, or regions. Late-interaction retrieval addresses this by preserving token-level matching, as in COIL (Gao et al., 2021), SPLADE (Formal et al., 2021b;a), PLAID (Santhanam et al., 2022a), WARP (Scheerer et al., 2025), and multi-vector vision-language retrievers such as ColPali and ColQwen2. However, storing and scoring many visual token embeddings per candidate is costly.

2.2 VISUAL TOKEN REDUCTION AND RETRIEVAL-SIDE COMPRESSION

Visual token reduction has been explored through pruning and merging methods such as LLaVA-PruMerge (Shang et al., 2025), LVPruning (Sun et al., 2025), VisionZip (Yang et al., 2025), ZipVL (He et al., 2024), LiteLVLM (Lee & Choi, 2026), and ToMe (Bolya et al., 2023). These methods mainly reduce transformer computation inside generative LVLMs or vision encoders, whereas SaMer compresses post-projector retrieval embeddings stored in the multi-vector index and used by late-interaction scoring. Recent work has studied retrieval-side compression for multi-vector vision-language retrievers. Prior work compares pruning and merging strategies for ColPali and ColQwen2 (Ma et al., 2025), while HPC-ColPali (Bach, 2025) combines quantization, attention-guided pruning, and binary encoding. SAP (Liu et al., 2026) prunes structural anchor patches, and H-Pool (Qin et al., 2026) performs modality-agnostic fixed-budget pooling for multi-vector indexes. Unlike these, SaMer uses object annotations during adaptation to shape the merge assignment and discourage cross-instance mixing, requiring no detector at inference.

3 PRELIMINARIES

3.1 MULTI-VECTOR LATE INTERACTION

Multi-vector vision-language retrieval represents images and text queries as sets of token embeddings and computes retrieval scores through late interaction. Given an image I, the vision encoder and projection layer produce N visual tokens, denoted as $V(I) = \{v_i\}_{i=1}^{N}$, where $v_i \in \mathbb{R}^D$. Similarly, a text query q is represented as M query tokens $Q(q) = \{q_j\}_{j=1}^{M}$, where $q_j \in \mathbb{R}^D$. Both image and text tokens are projected into the same D-dimensional retrieval space.

The retrieval score is defined by MaxSim late interaction. Each query token is matched to the most similar visual token, and the query-image score is computed as

$$S(q, I) = \frac{1}{M} \sum_{j=1}^{M} \max_{i \in [N]} q_j^\top v_i. \tag{1}$$

This scoring allows query tokens to match objects, attributes, or regions in the image, making multi-vector retrieval effective for fine-grained visual evidence. However, since every query token should be compared with every visual token, the number of visual tokens N becomes the bottleneck for both storage and computation.

3.2 POST-PROJECTOR TOKEN COMPRESSION

In multi-vector vision-language retrieval, each image is represented by N post-projector visual tokens $V = \{v_i\}_{i=1}^{N}$, where each token has been mapped into the shared image-text retrieval space. These tokens are stored in the retrieval index and used by MaxSim scoring. Therefore, full-token retrieval requires $O(ND)$ storage per image for D-dimensional tokens and computes MN query-image token similarities per image, corresponding to $O(MND)$ dot-product cost for a query with M tokens. Reducing the number of image-side post-projector tokens can directly lower both index storage and scoring cost while preserving the encoder and late-interaction retrieval interface. Unlike token pruning, which discards visual evidence before future queries are known, token merging aims to aggregate the token set into a smaller set of representative tokens.

4 METHODS

Our framework has three steps, all designed to keep the original late-interaction interface while replacing the image side with compressed representatives. First, SaMer compresses the image-side post-projector tokens into K representative tokens using feature-spatial soft assignment. Second, during training only, object annotations guide which tokens may be merged by penalizing assignments that mix different object instances. Third, SaMer adapts only the shared projection layer so that the compressed tokens remain compatible with MaxSim scoring, while the vision encoder and language backbone stay frozen. Figure 1 illustrates the overall process.

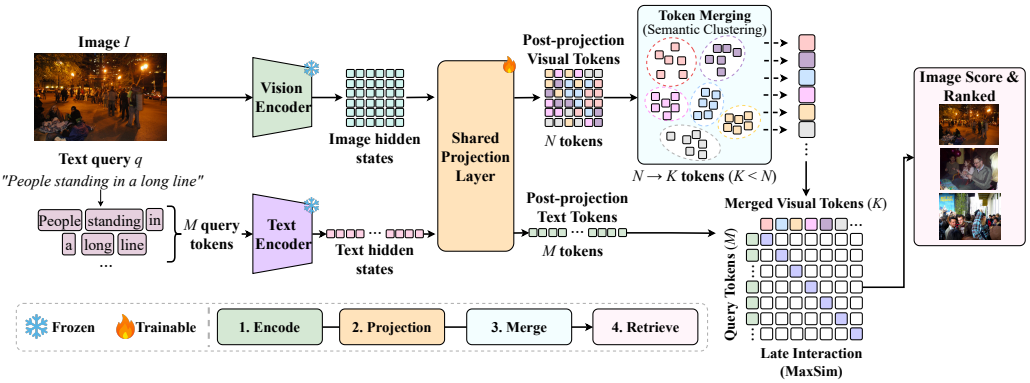


Figure 1: Overview of SaMer. Frozen vision and language encoders feed a shared trainable projection layer, and image-side post-projector tokens are merged from N to K representatives before MaxSim scoring. Training uses object-aware merging with projection-only adaptation, while inference is annotation-free and preserves the original late-interaction interface.

4.1 Post-projector Visual Token Merging

SaMer reduces the number of visual tokens while keeping the late-interaction interface unchanged. Given the full post-projector visual token set $V = \{v_i\}_{i=1}^{N}$, SaMer assigns these tokens to K representatives, where $K \ll N$, and constructs one merged token for each representative. These post-projector embeddings are the vectors stored in the retrieval index and compared with query tokens by MaxSim. Therefore, compressing them directly reduces index storage and query-time similarity computation without modifying the vision encoder or the retrieval scoring function.

We construct the compressed representation with feature-spatial centroid merging. Each visual token v_i is associated with a spatial coordinate $p_i \in [0,1]^2$, and each representative k maintains a feature centroid μ_k and a spatial centroid s_k initialized from K uniformly spaced seed tokens in the flattened image-token grid. SaMer assigns tokens to representatives using a distance that combines feature similarity and spatial coherence:

$$d(i,k) = \underbrace{\left(1 - v_i^\top \mu_k\right)}_{\text{feature similarity}} + \gamma \underbrace{\|p_i - s_k\|_2^2}_{\text{spatial coherence}}, \tag{2}$$

where γ controls the strength of the spatial term. The feature term groups semantically similar tokens, while the spatial term discourages visually similar but spatially distant regions from being merged too aggressively.

SaMer converts this distance $d(i,k)$ into soft assignment weights $a_{i,k}$ and forms each representative as the corresponding weighted centroid with ℓ_2 normalization:

$$\begin{aligned} a_{i,k} &= \mathrm{softmax}_k\left(-\frac{d(i,k)}{\tau_s}\right), \\ r_k &= \mathrm{normalize}_2\left(\frac{\sum_{i=1}^{N} a_{i,k} v_i}{\sum_{i=1}^{N} a_{i,k}}\right). \end{aligned} \tag{3}$$

Here, $a_{i,k}$ is the assignment weight from token i to representative k, and τ_s is the soft-assignment temperature. Unlike pruning, the centroid construction in Eq. 3 does not discard visual tokens but aggregates them into compact representatives. The compressed representation $R(I) = \{r_k\}_{k=1}^{K}$ is scored with the same late-interaction rule as Eq. 1, except that MaxSim is taken over merged tokens:

$$S_K(q,I) = \frac{1}{M}\sum_{j=1}^{M} \max_{k \in [K]} q_j^\top r_k. \tag{4}$$

Thus, training-free SaMer acts as a cache-time compression method where the query encoder, image encoder, projection layer, and scoring rule stay unchanged, and only the stored image-side representation is replaced with K merged tokens.

4.2 OBJECT-AWARE MERGE PRIOR

When object-level annotations are available during training, SaMer uses them as a weak merge prior rather than an auxiliary grounding loss. The goal is to make merged representatives more object-consistent, since visually similar regions may belong to different object instances and a purely feature-based assignment may collapse them into one representative. For each visual token v_i, we assign a bbox label b_i using its spatial coordinate p_i: tokens outside all bboxes receive a background label, tokens inside a bbox receive the corresponding object instance label, and tokens inside multiple boxes are assigned to the smallest bbox. These labels are used only during training to define the merge prior; background and context tokens are not removed from the compressed representation.

To build the object-aware prior, we first estimate which bbox labels dominate each representative. We do this with a stop-gradient hard assignment $c_i = \arg\min_{k\in[K]} d(i,k)$, computed from the feature-spatial distance before applying the object-aware penalty. This hard assignment is not used to form the final merged token; it is used only to estimate the bbox-label distribution of each representative, so gradients do not backpropagate through c_i:

$$\begin{aligned} P_k(b) &= \frac{\sum_{i=1}^{N} \mathbf{1}[c_i = k]\mathbf{1}[b_i = b]}{\sum_{i=1}^{N} \mathbf{1}[c_i = k] + \epsilon}, \\ P_{\text{inst}}(i,k) &= 1 - P_k(b_i). \end{aligned} \tag{5}$$

This penalty is small when token i shares the bbox label with most tokens in representative k, and large when assigning i would mix evidence from different object instances.

After estimating this prior, SaMer returns to soft assignment to construct the actual training-time merged tokens. The object-aware penalty is added to the feature-spatial distance, so cross-instance assignments receive lower weight:

$$\begin{aligned} a_{i,k} &= \text{softmax}_k\left(-\frac{d(i,k) + P_{\text{inst}}(i,k)}{\tau_s}\right), \\ r_k^{\text{soft}} &= \text{normalize}_2\left(\frac{\sum_{i=1}^{N} a_{i,k} v_i}{\sum_{i=1}^{N} a_{i,k}}\right). \end{aligned} \tag{6}$$

where the softmax is taken over representatives $k = 1, \ldots, K$ for each token i. During training, SaMer uses centroids r_k^{soft} in Eq. 6 for compressed MaxSim scoring. The penalty reshapes the soft assignment weights, through which gradients flow to the projected embeddings and merged centroids. At inference time, SaMer uses no bboxes or object labels. Instead, it performs bbox-free feature-spatial soft assignment, with the object-aware prior reflected in the adapted projection space.

4.3 COMPRESSION-AWARE PROJECTION-ONLY ADAPTATION

Training-free SaMer can be directly applied to an existing multi-vector retriever, but small-K compression creates a mismatch between the representation used by the original retriever and the compressed representation used by SaMer. The projection layer is trained assuming all N visual tokens are available for late-interaction scoring, whereas SaMer scores only K merged tokens. We therefore adapt only the image-text shared projection layer using the compressed score S_K, while keeping the vision encoder and language backbone frozen. This aligns the retrieval space with compressed tokens without changing the backbone representations or introducing a new retrieval architecture.

Token merging stays non-parametric during adaptation, adding no learnable merge parameters. However, because the merged centroids are derived from projected visual embeddings through soft assignment and weighted averaging, gradients from the compressed MaxSim score still update the projection layer. Thus, the projection learns to produce features that stay effective after merging. When object annotations are available, the object-aware prior shapes the soft assignments without an auxiliary loss, and SaMer is optimized only with a multi-positive InfoNCE retrieval loss (Oord et al., 2018) over compressed MaxSim scores:

$$\mathcal{L}_{ret} = -\log \frac{\sum_{I^+\in\mathcal{P}(q)} \exp(S_K(q, I^+)/\tau)}{\sum_{I\in\mathcal{B}} \exp(S_K(q, I)/\tau)}. \tag{7}$$

Table 1: Retriever-only results across four benchmarks. SaMer compresses the image-side representation to $K = 64$ tokens and is compared with compression baselines applied to the same backbone when available. Shaded rows indicate SaMer variants. **Bold** and underline indicate the best and second-best results within each method group or backbone family, respectively.

Method	Flickr30K			MSCOCO			ImageCoDe			DocVQA		
	R@1	R@5	nDCG@10	R@1	R@5	nDCG@10	R@1	R@5	nDCG@10	R@1	R@5	nDCG@10
Single-vector												
CLIP	58.2	82.9	73.7	30.8	56.0	47.7	1.4	7.0	5.7	6.7	13.8	11.8
OpenCLIP	76.5	93.5	87.0	48.5	73.2	64.4	2.2	9.1	7.3	15.3	26.8	23.1
MetaCLIP	62.8	85.8	77.2	35.7	61.7	52.8	2.1	7.6	6.3	5.4	14.4	11.5
EVA-CLIP	**79.9**	94.7	**89.2**	51.0	74.6	66.1	2.5	8.8	7.5	12.9	25.1	19.8
DFN-CLIP	79.6	**94.9**	89.0	**53.4**	**77.8**	**69.0**	1.9	**9.8**	**7.7**	12.4	27.7	21.8
SigLIP2	79.8	94.2	88.8	52.1	75.2	66.9	**2.6**	9.4	7.5	**15.5**	**29.1**	**24.3**
VLM-based												
VLM2Vec-v2	41.7	68.8	59.0	20.2	44.1	36.5	1.0	5.1	4.4	4.0	10.4	8.5
GME	**74.6**	**91.8**	**85.2**	**47.3**	**71.1**	**62.7**	**2.3**	**9.1**	**7.7**	**41.5**	**60.3**	**53.3**
VisRAG	69.6	90.0	82.2	42.5	67.5	58.9	1.7	7.0	6.0	34.8	51.2	45.6
Multi-vector												
ColPali	77.0	92.9	86.9	47.4	71.7	63.3	5.4	16.0	13.2	**51.0**	**66.5**	**60.7**
HPC	54.0	77.8	69.0	30.6	52.5	45.4	3.6	11.3	9.0	26.4	39.7	34.9
H-Pool	73.7	91.4	84.7	44.9	69.6	60.9	5.6	15.6	12.9	45.0	61.9	55.5
SAP	68.3	88.6	80.8	39.6	63.7	55.4	3.8	12.5	10.1	40.4	55.2	50.2
SaMer (w/o FT)	73.5	91.1	84.5	45.1	69.7	61.1	5.2	**17.0**	14.1	44.6	62.1	55.6
SaMer (w/ FT)	**82.4**	**96.3**	**90.9**	**51.6**	**75.5**	**67.1**	**5.9**	**17.0**	**14.4**	45.7	60.5	55.6
ColQwen2	73.6	91.3	84.7	43.3	68.0	59.5	4.7	14.6	12.2	**53.2**	**67.9**	**63.0**
HPC	67.1	88.7	80.4	40.1	64.8	56.5	4.4	13.9	11.6	39.9	55.2	49.6
H-Pool	71.8	90.8	83.6	42.2	67.2	58.6	4.7	14.2	12.3	50.1	64.8	59.3
SAP	69.3	89.9	82.2	40.7	65.8	57.2	4.8	13.7	11.8	45.5	59.4	54.7
SaMer (w/o FT)	71.3	90.4	83.3	42.1	67.0	58.4	4.8	15.6	**13.0**	47.2	63.9	57.3
SaMer (w/ FT)	**79.3**	**94.7**	**88.7**	**47.5**	**72.3**	**63.5**	**5.1**	**16.2**	12.9	48.8	62.5	58.1

Here, $\mathcal{P}(q)$ is the positive-image set for query q, $\mathcal{B}$ the batch candidate set, and τ the temperature. At inference, K merged visual tokens are cached, and retrieval uses the same late-interaction rule over the compressed image representation.

5 Experiments

5.1 Datasets

We evaluate SaMer with retriever-only evaluation on Flickr30K, MSCOCO, ImageCoDe (Krojer et al., 2022), and DocVQA (Mathew et al., 2021). Flickr30K and MSCOCO evaluate natural-image text-to-image retrieval, where queries often refer to objects, attributes, and relations in natural scenes. Since SaMer is adapted only on the Flickr30K-Entities (Plummer et al., 2015) training split, we treat Flickr30K as in-domain and MSCOCO as cross-dataset. ImageCoDe evaluates compositional retrieval over visually similar images, where small object, attribute, or relation differences can determine the correct image, testing whether compression preserves subtle query-dependent evidence among near-miss candidates. DocVQA provides a document-domain reference, where retrieval often depends on sparse OCR tokens and layout cues rather than natural-object evidence. It therefore serves as a boundary case for object-centric compression in a domain that requires broad textual and spatial coverage. We report Recall@1 (R@1), Recall@5 (R@5), and nDCG@10.

5.2 Baselines

We compare against single-vector, VLM-based, multi-vector, and retrieval-side compression baselines. Single-vector baselines: CLIP (Radford et al., 2021), OpenCLIP (Cherti et al., 2023), MetaCLIP (Xu et al., 2024), EVA-CLIP (Sun et al., 2023), DFN-CLIP (Fang et al., 2023), and SigLIP2 (Tschannen et al., 2025), which use global image-query embeddings. VLM-based baselines: VLM2Vec-v2 (Meng et al., 2026), GME (Zhang et al., 2024), and VisRAG (Yu et al., 2025),

with stronger multimodal backbones. Multi-vector baselines: ColPali and ColQwen2. Retrieval-side compression baselines include H-Pool, HPC, and SAP, which reduce image-side retrieval embeddings through pooling, attention-based pruning, or structural anchor pruning. We apply SaMer to the multi-vector retrievers in training-free and adapted settings. For controlled comparisons, compression baselines use the same backbones and $K = 64$ visual tokens when applicable, following their method-specific settings. Unless otherwise stated, SaMer uses $K = 64$, adapts the shared projection layer for 3 epochs on Flickr30K-Entities, and keeps the backbones frozen.

5.3 Results

Table 1 presents retriever-only performance across natural-image, compositional-image, and document-image benchmarks. SaMer is most effective on natural-image retrieval, where redundant visual tokens coexist with query-specific object and region evidence. On Flickr30K, SaMer improves ColPali R@1 from 77.0 to 82.4 and ColQwen2 R@1 from 73.6 to 79.3, suggesting that compact object-aware representatives can improve retrieval under aggressive token reduction. It also outperforms H-Pool, the strongest external compression baseline, whose R@1 is 73.7 with ColPali and 71.8 with ColQwen2. The same trend holds on MSCOCO, where SaMer improves ColPali R@1 from 47.4 to 51.6 and outperforms H-Pool, SAP, and HPC. These results suggest that feature-similarity pooling alone may be less effective than object-aware merging with projection-only adaptation for preserving object evidence under MaxSim retrieval.

SaMer performs strongly on ImageCoDe, where successful retrieval requires distinguishing visually similar images using subtle object, attribute, and relation cues. With ColPali, SaMer improves R@1 from 5.4 to 5.9 and nDCG@10 from 13.2 to 14.4, supporting our view that efficient compression should preserve query-selectable object evidence rather than only retain globally salient or visually similar patches. DocVQA serves as a boundary case beyond the object-centric retrieval setting targeted by SaMer: document images require broad OCR and layout coverage, so aggressive token compression can remove sparse textual evidence rather than redundant visual patches. SaMer is therefore not optimized for this setting, but remains competitive with compressed baselines using a compact image-side representation.

6 Analysis of SaMer

6.1 Ablation Study

Is the Gain Only from Adaptation? One possible explanation for SaMer's gains is that projection-only adaptation alone improves the compressed retrieval space. To separate adaptation from the merge design, Table 2 compares SaMer with other compression methods trained under the same adaptation setting. All methods use the same frozen backbone, the same projection-only update, and the same $K = 64$ visual token budget.

Table 2: Adapted compression results at $K = 64$ under projection-only adaptation. **Bold** and underline denote the best and second-best results.

Method	Flickr30K		MSCOCO		ImageCoDe	
	R@1	nDCG@10	R@1	nDCG@10	R@1	nDCG@10
ColPali						
H-Pool	79.2	88.7	<u>49.7</u>	<u>65.3</u>	**5.9**	<u>13.8</u>
HPC	67.7	81.1	37.8	53.4	<u>4.7</u>	11.8
SAP	<u>80.2</u>	<u>89.5</u>	47.3	62.9	4.6	12.6
SaMer	**82.4**	**90.9**	**51.6**	**67.1**	**5.9**	**14.4**
ColQwen2						
H-Pool	77.0	87.0	46.1	62.1	<u>5.0</u>	<u>12.9</u>
HPC	77.9	87.8	45.3	61.7	4.9	12.7
SAP	<u>78.2</u>	<u>88.2</u>	<u>46.4</u>	<u>62.5</u>	5.0	12.4
SaMer	**79.3**	**88.7**	**47.5**	**63.5**	**5.1**	**12.9**

SaMer achieves the best R@1 and nDCG@10 on natural-image and compositional retrieval metrics for ColPali and ColQwen2. This suggests that the improvement is not a fine-tuning effect. If projection-only adaptation were the sole explanation, we would expect the gap between SaMer and the other adapted compression variants to be smaller under the same training setup. Instead, the results suggest that SaMer remains stronger because its merge rule better controls which visual evidence survives compression.

Compression Budget Analysis. Table 3 reports retrieval performance as the token budget K varies, and Figure 2 visualizes the trade-off between token budget and R@5. Retrieval performance improves as K increases from 32 to 64, but the gain becomes small beyond that point. On

Table 3: Compression budget analysis on Flickr30K and MSCOCO as K varies. $K = 64$ provides a favorable trade-off between retrieval quality and image-side compression.

K	Comp. ratio (%)	Flickr30K			MSCOCO		
		R@1	R@5	nDCG@10	R@1	R@5	nDCG@10
512	50.29	**83.7**	**96.8**	**91.7**	52.2	76.1	67.5
256	75.15	83.5	96.7	91.6	**52.4**	**76.2**	**67.7**
128	87.57	83.1	96.5	91.3	52.3	76.0	67.6
64	93.79	82.4	96.3	90.9	51.6	75.5	67.1
32	96.89	81.3	95.8	90.3	51.3	75.0	66.4

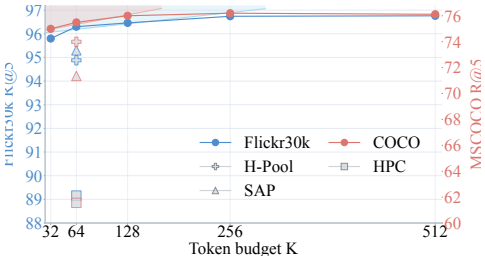


Figure 2: Token budget vs R@5. Solid lines show SaMer across token budgets, with baselines marked at $K = 64$. SaMer shows diminishing returns beyond $K = 64$ and outperforms baselines under the same budget.

Flickr30K, R@5 rises from 95.8 at $K = 32$ to 96.3 at $K = 64$, and then increases only to 96.8 at $K = 512$. MSCOCO shows the same pattern, increasing from 75.0 at $K = 32$ to 75.5 at $K = 64$, and reaching 76.1 at $K = 512$. This suggests that a small set of merged tokens is sufficient to preserve most retrieval performance on these benchmarks.

Figure 2 compares SaMer with competing compression methods at the same $K = 64$ budget. Under this controlled setting, SaMer lies above H-Pool, SAP, and HPC on both Flickr30K and MSCOCO, indicating a better quality-budget trade-off at the same token budget. Taken together, these results make $K = 64$ a favorable operating point, balancing strong retrieval quality with substantial image-side compression. We use $K = 64$ throughout the main experiments.

Table 4: The object-aware merge prior improves both retrieval performance and object consistency.

Variant	Merge Components			Flickr30K			MSCOCO			Grounding		
	Feature	Spatial	Obj. prior	R@1	R@5	nDCG@10	R@1	R@5	nDCG@10	BoxMass	RegionHit	Coverage IoU
(A)	✓	✗	✗	80.7	95.6	89.8	49.8	73.7	65.4	47.6	67.9	13.0
(B)	✓	✓	✗	80.4	95.5	89.7	49.8	73.9	65.5	47.8	68.0	13.1
SaMer	✓	✓	✓	**82.4**	**96.3**	**90.9**	**51.6**	**75.5**	**67.1**	**54.2**	**68.3**	**16.4**

Merge Component Study. Table 4 analyzes each merge component's contribution. Feature-only merging groups visually similar tokens, but adding spatial coherence alone provides little benefit. Flickr30K R@1 changes from 80.7 to 80.4, with MSCOCO R@1 nearly unchanged. This suggests that local smoothness is insufficient for retrieval-oriented compression because query-relevant evidence can depend on object identity rather than spatial proximity. In other words, spatial coherence helps organize nearby tokens, but does not by itself make the merged representation object-aware.

The object-aware merge prior provides the main gain, raising Flickr30K R@1 to 82.4 and MSCOCO R@1 to 51.6 while also improving grounding quality. This suggests that the gain comes not only from projection adaptation but also from enforcing stronger object-instance consistency in feature-spatial merging. BoxMass, RegionHit, and CoverageIoU measure alignment between phrase relevance and annotated object regions; their definitions are provided in Appendix C. BoxMass increases from 47.8 to 54.2 and CoverageIoU from 13.1 to 16.4, indicating that the prior improves retrieval by preserving phrase-level evidence within the correct object regions.

6.2 Grounding Comparison

Figure 3 evaluates whether compressed representations preserve phrase-level visual evidence. Compared with full ColPali, compression baselines show mixed behavior: some improve weak localization, but their relevance can remain spread across nearby objects or background regions. This suggests that reducing visual tokens can still leave enough signal to reach the general target area, but not necessarily preserve object-specific evidence for precise phrase grounding. SaMer shows a more consistent pattern. Training-free SaMer improves RegionHit from 10.5 to 62.8 and Cover-

ageIoU from 2.1 to 11.7, indicating that feature-spatial merging alone helps the representation retain spatial access to the correct phrase region even without projection adaptation.

Object-aware projection-only adaptation further improves relevance concentration. SaMer raises BoxMass from 41.3 to 54.2, surpassing the full-token baseline of 51.8, while improving RegionHit to 68.3 and CoverageIoU to 16.4. This distinction matters because a method may reach the target region while still assigning substantial relevance to surrounding context, nearby objects, or visually similar background patches. The BoxMass gain shows that SaMer does not merely overlap the relevance map with the target bbox; it concentrates more phrase-specific evidence inside the annotated object region. With the gains in RegionHit and CoverageIoU, this suggests that the object-aware prior helps merged tokens preserve both localization and evidence concentration under aggressive compression. Qualitative grounding examples are provided in Appendix E.

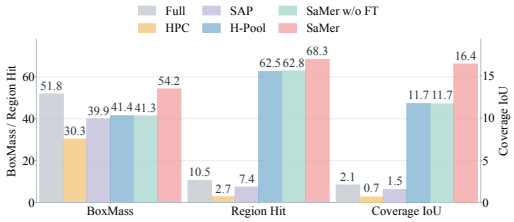


Figure 3: Grounding comparison of ColPali, compression baselines, and SaMer. BoxMass and RegionHit use the left axis, and CoverageIoU the right. SaMer improves localization and evidence concentration under compression.

6.3 Efficiency Analysis

Table 5: Storage and cached-embedding retrieval efficiency on Flickr30K and MSCOCO.

Variant	Flickr30K					MSCOCO				
	Tokens/ img	Storage/ 1M (GB)	MaxSim Ops (10^{12})	Time (s)	QPS	Tokens/ img	Storage/ 1M (GB)	MaxSim Ops (10^{12})	Time (s)	QPS
ColPali										
Full	1030.0	263.7	36.87	7.994	625.5	1030.0	263.7	845.51	160.879	154.2
SaMer	64.0	16.4	2.29	1.841	2716.3	64.0	16.4	52.54	17.636	1406.7
ColQwen2										
Full	238.4	61.0	8.01	3.619	1381.6	127.4	32.6	97.50	78.837	314.7
SaMer	64.0	16.4	2.15	1.869	2674.7	64.0	16.4	48.99	25.585	969.7

Table 5 summarizes the storage and retrieval efficiency of SaMer at the $K = 64$ setting under the same late-interaction scoring interface. Storage is estimated for one million images using FP16 embeddings with $D = 128$, while MaxSim Ops denotes scalar multiplications in query–image token dot products over the evaluation set and Time denotes total retrieval evaluation time. By compressing only the cached image-side representations, SaMer reduces both index storage and MaxSim computation without changing the retrieval scoring rule. For ColPali, reducing the image representation from 1,030 to 64 tokens yields a $16.09\times$ reduction in storage and $16.1\times$ fewer MaxSim operations on Flickr30K and MSCOCO, resulting in QPS improvements of $4.3\times$ and $9.1\times$, respectively. For ColQwen2, the corresponding storage reductions are $3.73\times$ and $1.99\times$, with approximately $3.7\times$ and $2.0\times$ fewer MaxSim operations and $1.9\times$ and $3.1\times$ higher QPS. Although the throughput gains are smaller than the reductions in MaxSim operations, SaMer consistently reduces storage and scoring cost while improving retrieval throughput under the same late-interaction formulation.

7 Conclusion

In this work, we present SaMer, an object-aware token merging framework for efficient multi-vector vision-language retrieval. Motivated by the view that image-side compression should preserve query-selectable evidence rather than merely reduce token count, SaMer compresses post-projector image tokens into K representative centroids. Using object annotations only during training, SaMer discourages cross-instance merging while requiring no bboxes or detectors at inference time. With projection-only adaptation, SaMer aligns compressed visual representations with the existing retrieval space, keeping the vision and language backbones frozen. Experiments show that $K = 64$ merged tokens substantially reduce storage and MaxSim scoring cost while keeping strong retrieval performance and phrase-level grounding. Overall, our results suggest that efficient late-interaction retrieval depends not only on token count, but also on preserving query-selectable evidence.

Reproducibility statement

To support reproducibility, we provide implementation code, execution scripts, and configuration files for SaMer training, cache construction, inference, and evaluation in the supplementary material. Appendix B describes the main implementation details, including the token budget, merging hyperparameters, training schedule, batch size, optimization settings, and hardware. The main paper and appendix further specify the dataset and evaluation settings, definitions of the grounding metrics, and the SaMer procedures used during training and inference. For controlled comparisons, competing compression methods use the same backbone and a visual-token budget of $K = 64$ when applicable. We verified that the reported results are consistent with the corresponding experimental settings.

AI Use Statement

Generative AI tools were used solely to assist with language editing, including wording and grammar, and with code debugging during implementation. All AI-assisted suggestions were independently reviewed and verified by the authors. Generative AI tools were not used to formulate research ideas, design the methodology or experiments, interpret the results, or develop scientific claims. These aspects of the work were developed and verified independently by the authors. The authors take full responsibility for the content of the paper.

References

Duong Bach. Hierarchical patch compression for colpali: Efficient multi-vector document retrieval with dynamic pruning and quantization. *arXiv preprint arXiv:2506.21601*, 2025.

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

Qingqing Cao, Bhargavi Paranjape, and Hannaneh Hajishirzi. Pumer: Pruning and merging tokens for efficient vision language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, pp. 12890–12903, 2023.

Mehdi Cherti, Romain Beaumont, Ross Wightman, Mitchell Wortsman, Gabriel Ilharco, Cade Gordon, Christoph Schuhmann, Ludwig Schmidt, and Jenia Jitsev. Reproducible scaling laws for contrastive language-image learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2818–2829, June 2023.

Haiwen Diao, Ying Zhang, Lin Ma, and Huchuan Lu. Similarity reasoning and filtration for image-text matching. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 1218–1226, 2021.

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

Alex Fang, Albin Madappally Jose, Amit Jain, Ludwig Schmidt, Alexander Toshev, and Vaishaal Shankar. Data filtering networks. *arXiv preprint arXiv:2309.17425*, 2023.

Manuel Faysse, Hugues Sibille, Tony Wu, Bilel Omrani, Gautier Viaud, Céline Hudelot, and Pierre Colombo. Colpali: Efficient document retrieval with vision language models. *arXiv preprint arXiv:2407.01449*, 2024.

Thibault Formal, Carlos Lassance, Benjamin Piwowarski, and Stéphane Clinchant. Splade v2: Sparse lexical and expansion model for information retrieval. *arXiv preprint arXiv:2109.10086*, 2021a.

Thibault Formal, Benjamin Piwowarski, and Stéphane Clinchant. Splade: Sparse lexical and expansion model for first stage ranking. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 2288–2292, 2021b.

Luyu Gao, Zhuyun Dai, and Jamie Callan. Coil: Revisit exact lexical match in information retrieval with contextualized inverted list. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 3030–3042, 2021.

Yefei He, Feng Chen, Jing Liu, Wenqi Shao, Hong Zhou, Kaipeng Zhang, and Bohan Zhuang. Zipvl: Efficient large vision-language models with dynamic token sparsification. *arXiv preprint arXiv:2410.08584*, 2024.

Hsiang-Wei Huang, Wenhao Chai, Kuang-Ming Chen, Cheng-Yen Yang, and Jenq-Neng Hwang. Tosa: Token merging with spatial awareness. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 9654–9660. IEEE, 2025.

Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International conference on machine learning*, pp. 4904–4916. PMLR, 2021.

Omar Khattab and Matei Zaharia. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 39–48, 2020.

Benno Krojer, Vaibhav Adlakha, Vibhav Vineet, Yash Goyal, Edoardo Ponti, and Siva Reddy. Image retrieval from contextual descriptions. *arXiv preprint arXiv:2203.15867*, 2022.

Dong Hoon Lee and Seunghoon Hong. Learning to merge tokens via decoupled embedding for efficient vision transformers. *Advances in Neural Information Processing Systems*, 37:54079–54104, 2024.

Sangin Lee and Yukyung Choi. Clip tricks you: Training-free token pruning for efficient pixel grounding in large vision-language models. *arXiv preprint arXiv:2605.13178*, 2026.

Junnan Li, Ramprasaath Selvaraju, Akhilesh Gotmare, Shafiq Joty, Caiming Xiong, and Steven Chu Hong Hoi. Align before fuse: Vision and language representation learning with momentum distillation. *Advances in neural information processing systems*, 34:9694–9705, 2021.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International conference on machine learning*, pp. 12888–12900. PMLR, 2022.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pp. 19730–19742. PMLR, 2023.

Kunpeng Li, Yulun Zhang, Kai Li, Yuanyuan Li, and Yun Fu. Visual semantic reasoning for image-text matching. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4654–4662, 2019.

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. In *International Conference on Learning Representations*, 2022.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars (eds.), *Computer Vision – ECCV 2014*, pp. 740–755, Cham, 2014. Springer International Publishing. ISBN 978-3-319-10602-1.

Chenxi Liu, Junhua Mao, Fei Sha, and Alan Yuille. Attention correctness in neural image captioning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31, 2017.

Zhuchenyang Liu, Ziyu Hu, Yao Zhang, and Yu Xiao. Structural anchor pruning: Training-free multi-vector compression for visual document retrieval, 2026. URL https://arxiv.org/abs/2601.20107.

Yubo Ma, Jinsong Li, Yuhang Zang, Xiaobao Wu, Xiaoyi Dong, Pan Zhang, Yuhang Cao, Haodong Duan, Jiaqi Wang, Yixin Cao, and Aixin Sun. Towards storage-efficient visual document retrieval: An empirical study on reducing patch-level embeddings. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 19568–19580, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.1003. URL https://aclanthology.org/2025.findings-acl.1003/.

Dmitrii Marin, Jen-Hao Rick Chang, Anurag Ranjan, Anish Prabhu, Mohammad Rastegari, and Oncel Tuzel. Token pooling in vision transformers for image classification. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2023.

Minesh Mathew, Dimosthenis Karatzas, and C. V. Jawahar. Docvqa: A dataset for vqa on document images. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2021.

Rui Meng, Ziyan Jiang, Ye Liu, Mingyi Su, Xinyi Yang, Yuepeng Fu, Can Qin, Raghuveer Thirukovalluru, Xuan Zhang, Zeyuan Chen, Ran Xu, Caiming Xiong, Yingbo Zhou, Wenhu Chen, and Semih Yavuz. VLM2vec-v2: Advancing multimodal embedding for videos, images, and visual documents. *Transactions on Machine Learning Research*, 2026. ISSN 2835-8856. URL https://openreview.net/forum?id=TpU38jbKIJ.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Bryan A. Plummer, Liwei Wang, Chris M. Cervantes, Juan C. Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. *arXiv preprint arXiv:1505.04870*, 2015.

Hanxiang Qin, Alexander Martin, Rohan Jha, Chunsheng Zuo, Reno Kriz, and Benjamin Van Durme. Multi-vector index compression in any modality. *arXiv preprint arXiv:2602.21202*, 2026.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

Navid Rajabi and Jana Kosecka. Q-groundcam: Quantifying grounding in vision language models via gradcam. *arXiv preprint arXiv:2404.19128*, 2024.

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, pp. 13937–13949, 2021.

Anna Rohrbach, Marcus Rohrbach, Ronghang Hu, Trevor Darrell, and Bernt Schiele. Grounding of textual phrases in images by reconstruction. In *European Conference on Computer Vision*, pp. 817–834. Springer, 2016.

Michael S Ryoo, AJ Piergiovanni, Anurag Arnab, Mostafa Dehghani, and Anelia Angelova. Tokenlearner: Adaptive space-time tokenization for videos. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=z-l1kpDXs88.

Keshav Santhanam, Omar Khattab, Christopher Potts, and Matei Zaharia. Plaid: an efficient engine for late interaction retrieval. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pp. 1747–1756, 2022a.

Keshav Santhanam, Omar Khattab, Jon Saad-Falcon, Christopher Potts, and Matei Zaharia. Colbertv2: Effective and efficient retrieval via lightweight late interaction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 3715–3734, 2022b.

Jan Luca Scheerer, Matei Zaharia, Christopher Potts, Gustavo Alonso, and Omar Khattab. Warp: An efficient engine for multi-vector retrieval. In *Proceedings of the 48th international ACM SIGIR conference on research and development in information retrieval*, pp. 2504–2512, 2025.

Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: visual explanations from deep networks via gradient-based localization. *International journal of computer vision*, 128(2):336–359, 2020.

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.

Quan Sun, Yuxin Fang, Ledell Wu, Xinlong Wang, and Yue Cao. Eva-clip: Improved training techniques for clip at scale. *arXiv preprint arXiv:2303.15389*, 2023.

Yizheng Sun, Yanze Xin, Hao Li, Jingyuan Sun, Chenghua Lin, and Riza Theresa Batista-Navarro. Lvpruning: An effective yet simple language-guided vision token pruning approach for multi-modal large language models. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 4299–4308, 2025.

Yehui Tang, Kai Han, Yunhe Wang, Chang Xu, Jianyuan Guo, Chao Xu, and Dacheng Tao. Patch slimming for efficient vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12165–12174, 2022.

Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, Olivier Hénaff, Jeremiah Harmsen, Andreas Steiner, and Xiaohua Zhai. Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786*, 2025.

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

Hu Xu, Saining Xie, Xiaoqing Tan, Po-Yao Huang, Russell Howes, Vasu Sharma, Shang-Wen Li, Gargi Ghosh, Luke Zettlemoyer, and Christoph Feichtenhofer. Demystifying CLIP data. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=5BCFlnfE1g.

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.

Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2:67–78, 2014.

Shi Yu, Chaoyue Tang, Bokai Xu, Junbo Cui, Junhao Ran, Yukun Yan, Zhenghao Liu, Shuo Wang, Xu Han, Zhiyuan Liu, and Maosong Sun. VisRAG: Vision-based retrieval-augmented generation on multi-modality documents. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=zG459X3Xge.

Jianming Zhang, Sarah Adel Bargal, Zhe Lin, Jonathan Brandt, Xiaohui Shen, and Stan Sclaroff. Top-down neural attention by excitation backprop. *International Journal of Computer Vision*, 126 (10):1084–1102, 2018.

Xin Zhang, Yanzhao Zhang, Wen Xie, Mingxin Li, Ziqi Dai, Dingkun Long, Pengjun Xie, Meishan Zhang, Wenjie Li, and Min Zhang. Gme: Improving universal multimodal retrieval by multimodal llms. *arXiv preprint arXiv:2412.16855*, 2024.

Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2921–2929, 2016.

A Centroid Construction Variants

Table 6: Centroid construction ablation on Flickr30K and MSCOCO. Soft assignment, used by SaMer, achieves the best performance across all metrics, showing that differentiable soft assignment centroids better preserve retrieval evidence than mean, normalized, medoid, or object-weighted alternatives.

Centroid variant	Flickr30K			MSCOCO		
	R@1	R@5	nDCG@10	R@1	R@5	nDCG@10
Mean	80.3	95.4	89.7	49.3	73.9	65.3
Normalized	82.3	96.1	90.7	51.5	75.5	67.0
Medoid	80.6	95.6	89.9	50.1	74.6	65.9
Object weighted	82.2	96.1	90.7	51.6	75.4	67.0
Soft assignment	**82.4**	**96.3**	**90.9**	**51.6**	**75.5**	**67.1**

Table 6 compares different ways of constructing merged representative tokens. The mean variant averages assigned tokens without additional normalization, while the normalized variant applies normalization after centroid construction. The medoid variant selects the token closest to the cluster center, preserving an existing token rather than forming an averaged representative. The object-weighted variant gives higher weight to tokens associated with object regions during centroid construction.

Soft assignment, used by SaMer, performs best across all metrics on both Flickr30K and MSCOCO. Compared with hard mean or medoid-style representatives, soft assignment centroids better preserve retrieval-relevant evidence by allowing tokens to contribute to clusters according to assignment confidence. The gain over the object-weighted variant further suggests that object information is most effective when used to shape the merge assignment itself, rather than only reweighting tokens after clusters have already been formed.

B Implementation Details

We implement SaMer on top of existing multi-vector vision-language retrievers by modifying only the image-side post-projector representation. Unless otherwise stated, all main experiments use $K = 64$ merged image tokens, a spatial weight of $\gamma = 0.1$, a retrieval temperature of $\tau = 0.07$, and a soft-assignment temperature of $\tau_s = 0.07$. During compression-aware adaptation, the vision encoder and language backbone are frozen, and only the shared projection layer is updated. We train for 1,746 optimization steps using AdamW with a per-GPU batch size of 32, gradient accumulation over 2 steps, and an effective batch size of 256 across 4 NVIDIA A100 80GB GPUs. We use a learning rate of 2×10^{-4}, a weight decay of 1×10^{-5}, and a cosine learning-rate schedule with a warmup ratio of 0.1.

Object annotations are used only during training to construct the object-aware merge prior. Each image token is assigned a label based on its spatial coordinate: tokens outside all bounding boxes are treated as background, tokens inside a single bounding box receive the corresponding object-instance label, and tokens contained in multiple bounding boxes are assigned to the smallest box. These labels are used solely to compute the instance inconsistency penalty and are not treated as an auxiliary supervision signal. At inference time, SaMer discards all object labels and bounding boxes and performs annotation-free feature-spatial soft assignment.

C Grounding Metrics

We define grounding metrics to test whether compressed tokens retain phrase-level visual evidence. Given a query phrase, we compute a normalized relevance map $\hat{S}(x) \in [0, 1]$ over image/grid locations x, and compare it with the ground-truth phrase box mask $G(x) \in \{0, 1\}$. Since SaMer aims to preserve object- and region-level evidence under compression, the metrics evaluate relevance concentration, weak localization, and spatial coverage.

BoxMass. BoxMass is our primary grounding metric. It measures whether phrase relevance is concentrated inside the annotated object region, following attention correctness and activation-based grounding evaluations (Liu et al., 2017; Rajabi & Kosecka, 2024):

$$\text{BoxMass} = \frac{\sum_x \hat{S}(x)G(x)}{\sum_x \hat{S}(x) + \epsilon}. \tag{8}$$

A higher BoxMass means that more phrase-specific evidence falls inside the correct object region, directly testing whether merging preserves the queried visual evidence.

RegionHit. RegionHit measures whether the model localizes the target phrase at all, following hit-based grounding and pointing-style protocols (Rohrbach et al., 2016; Zhang et al., 2018). Let $T = \{0.1, 0.2, \ldots, 0.9\}$ and $M_t(x) = \mathbf{1}[\hat{S}(x) \geq t]$ denote the thresholded high-relevance region. We define

$$\text{RegionHit} = \frac{1}{|T|} \sum_{t \in T} \mathbf{1}\left[\text{IoU}(M_t, G) \geq 0.05\right]. \tag{9}$$

We use a low IoU threshold because the goal is weak phrase-region localization rather than segmentation-quality prediction.

CoverageIoU. CoverageIoU measures how well the high-relevance region covers the target box, following activation-map localization analyses (Zhou et al., 2016; Selvaraju et al., 2020):

$$\text{CoverageIoU} = \frac{1}{|T|} \sum_{t \in T} \text{IoU}(M_t, G). \tag{10}$$

Unlike RegionHit, CoverageIoU measures the degree of spatial agreement rather than only whether the relevance map reaches the target. All three metrics are higher-is-better and together evaluate whether SaMer preserves phrase-level evidence after aggressive visual token compression.

D Merging vs. Pruning Comparison

Table 7: Merging vs. pruning on Flickr30K and MSCOCO under training-free and adapted settings. SaMer consistently outperforms pruning baselines.

Variant	Flickr30K			MSCOCO		
	R@1	R@5	nDCG@10	R@1	R@5	nDCG@10
ColPali	77.0	92.9	86.9	47.4	71.7	63.3
w/o FT						
Random pruning	68.6	88.5	80.9	40.5	65.3	56.7
Spatial pruning	68.8	88.6	81.3	41.1	65.8	57.3
SaMer	**73.5**	**91.1**	**84.5**	**45.1**	**69.7**	**61.1**
w/ FT						
Random pruning	79.9	94.9	89.0	48.1	72.7	64.0
Spatial pruning	79.5	94.9	89.0	49.0	73.2	64.7
SaMer	**82.4**	**96.3**	**90.9**	**51.6**	**75.5**	**67.1**

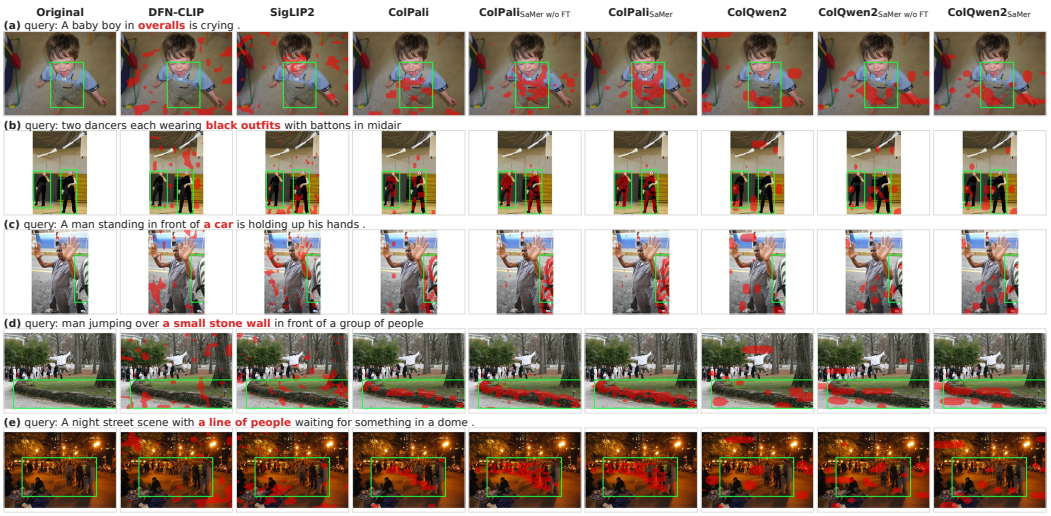
Table 7 shows that merging is more effective than pruning for multi-vector retrieval. In the training-free setting, SaMer incurs a smaller drop than random and spatial pruning, improving Flickr30K R@1 from 68.6 and 68.8 to 73.5, and MSCOCO R@1 from 40.5 and 41.1 to 45.1. This gap reflects a key difference between the two strategies: pruning removes visual evidence before the query is known, whereas merging keeps it in aggregated representatives. After projection-only adaptation, SaMer remains strongest across all metrics, reaching 82.4 R@1 on Flickr30K and 51.6 on MSCOCO. These results show that representative centroids preserve retrieval-relevant visual evidence better than discarding tokens, and that the advantage persists after adapting the projection space.

E QUALITATIVE GROUNDING EXAMPLES

Figure 4 provides qualitative grounding examples. Figure 4a compares single-vector baselines, full multi-vector retrievers, and SaMer variants, while Figure 4b compares compression methods applied to multi-vector retrievers. The compression comparison shows that baseline methods often recover part of the target region but can spread relevance to nearby objects or background regions, such as surrounding clothing, road areas, or non-target people. SaMer more consistently concentrates relevance inside the annotated phrase region after compression, supporting the quantitative grounding results.

F SAMER ALGORITHM

Algorithm 1 summarizes the compression-aware adaptation procedure of SaMer. The key difference between training and inference is the object-aware prior. During training, box labels define an instance inconsistency penalty inside the soft assignment. During inference, the object prior is discarded and the image-side representation is constructed using bbox-free feature-spatial soft assignment.



(a) Comparison with single-vector baselines, full multi-vector retrievers, and SaMer variants.



(b) Comparison with compression methods applied to multi-vector retrievers.

Figure 4: Qualitative grounding examples. Red denotes the phrase-relevance map and green denotes the ground-truth box for the highlighted phrase. (a) compares SaMer with single-vector baselines and full multi-vector retrievers. (b) compares SaMer with compression methods applied to multi-vector retrievers. Across both settings, SaMer better concentrates phrase relevance inside the annotated region, while compression baselines often spread relevance to nearby objects or background regions.

Algorithm 1 Compression-aware Adaptation and Inference with SaMer

Require: Training batch $\mathcal{B}$, token budget K, spatial weight γ, retrieval temperature τ, soft-assignment temperature τ_s

Require: Frozen vision encoder f_v, frozen language backbone f_t, trainable shared projection layer g_θ

Training: compression-aware projection-only adaptation

for each image-text pair $(I, q) \in \mathcal{B}$ **do**

Extract hidden states $H_I = f_v(I)$ and $H_q = f_t(q)$

Project image and query tokens into the retrieval space:

$$V = \{v_i\}_{i=1}^{N} = g_\theta(H_I), \qquad Q = \{q_j\}_{j=1}^{M} = g_\theta(H_q)$$

Initialize K representatives with feature centroids μ_k and spatial centroids s_k

Assign each image token a training-time box label b_i using its spatial coordinate p_i

Compute the feature-spatial distance:

$$d(i,k) = \left(1 - v_i^\top \mu_k\right) + \gamma \|p_i - s_k\|_2^2$$

Compute stop-gradient hard representative assignment:

$$c_i = \arg\min_{k \in [K]} d(i,k)$$

Compute hard-assignment label distribution $P_k(b)$ and instance penalty:

$$P_{\text{inst}}(i,k) = 1 - P_k(b_i)$$

Compute object-aware soft assignment:

$$a_{i,k}^{\text{obj}} = \text{softmax}_k\left(-\frac{d(i,k) + P_{\text{inst}}(i,k)}{\tau_s}\right)$$

Construct merged image tokens:

$$r_k = \text{normalize}_2\left(\frac{\sum_i a_{i,k}^{\text{obj}} v_i}{\sum_i a_{i,k}^{\text{obj}}}\right)$$

Compute compressed late-interaction score:

$$S_K(q, I) = \frac{1}{M} \sum_{j=1}^{M} \max_{k \in [K]} q_j^\top r_k$$

end for

Update only g_θ using the multi-positive retrieval loss

Inference: annotation-free retrieval

Extract and project visual tokens $V = \{v_i\}_{i=1}^{N}$

Compress V into K merged tokens $R(I) = \{r_k\}_{k=1}^{K}$ using bbox-free feature-spatial soft assignment

Cache $R(I)$ in the retrieval index

Extract and project query tokens $Q(q) = \{q_j\}_{j=1}^{M}$

Score each query-image pair using compressed late interaction:

$$S_K(q, I) = \frac{1}{M} \sum_{j=1}^{M} \max_{k \in [K]} q_j^\top r_k$$

return Retrieval score $S_K(q, I)$