Q-ViK: Question-Guided Visual KV Cache Eviction for Large Vision-Language Models

Solhee Hwang
Kookmin University
Seoul, Republic of Korea
solhee0912@kookmin.ac.kr

Jangho Kim*
Kookmin University
Seoul, Republic of Korea
jangho.kim@kookmin.ac.kr

Abstract

Large vision-language models (LVLMs) rely on increasingly long visual contexts, making the KV cache a major inference-time bottleneck. Existing KV cache compression methods typically use pre-decode signals such as prefill attention, visual redundancy, objectness, or layer-wise allocation, but these criteria do not directly indicate which visual tokens in the KV cache will be reused by future answer tokens. We introduce Q-ViK, a question-conditioned visual utility framework for LVLM KV cache eviction. During offline training, we run full-cache decoding and aggregate answer-to-visual attention over cached visual tokens, yielding a privileged *future-utility signal* that reflects the visual KV entries actually used during generation. We use this signal to train a lightweight question-conditioned scorer that predicts visual cache utility from prefill representations alone. At inference time, Q-ViK requires neither full-cache lookahead nor generated traces: it preserves textual KV entries and evicts low-utility visual KV entries with a single post-prefill scoring step. Experiments on seen and held-out multimodal benchmarks show that Q-ViK preserves task-relevant visual evidence in the KV cache more effectively than pre-decode saliency, especially under aggressive KV cache compression.Code is publicly available.

1 Introduction

Large vision-language models (LVLMs) [2, 16, 17, 11, 38, 30] have enabled strong performance on visual question answering, multimodal reasoning, OCR-based comprehension, chart understanding, and document-level image understanding. This capability, however, comes with a substantial inference-time memory cost. High-resolution images and multi-image inputs introduce hundreds or thousands of visual tokens, whose key-value (KV) states must be stored throughout autoregressive decoding. As visual contexts grow, the KV cache becomes a major bottleneck for LVLM deployment [27, 29].

KV cache eviction is a natural solution, but existing LVLM cache and token compression methods mostly rely on pre-decode or prefill-available signals, such as text-prior attention [27, 36], early-layer attention-based visual token pruning [3], visual redundancy or similarity [22, 34], or layer-wise KV allocation [29, 28, 33]. These criteria capture redundancy in visual representations and multimodal KV caches, but they do not directly indicate which visual tokens will receive high attention from future answer tokens. This distinction is important because LVLM generation is question-dependent: a token that appears weak under prompt-time saliency can later become critical evidence for producing the answer. We refer to this discrepancy as the *future-utility mismatch*.

Figure 1 illustrates this mismatch. Given the image and question in Fig. 1(a), prefill saliency in Fig. 1(b) emphasizes broad display-related regions before decoding. In contrast, the *future-utility signal* in Fig. 1(c), obtained from full-cache decoding by aggregating answer-to-visual attention,

*Corresponding author.

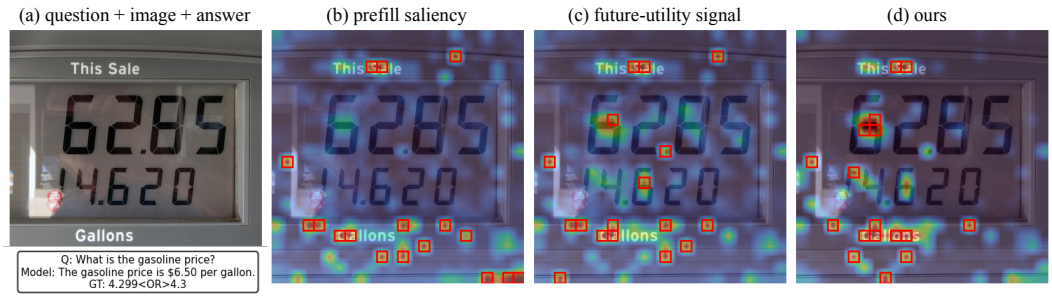

Figure 1: Eviction decisions under prefill saliency, the future-utility signal, and Q-ViK. (a) Image-question-answer example: "What is the gasoline price?" (b) Prefill-saliency eviction preserves broad display-related regions before decoding, but misses fine-grained price evidence. (c) The future-utility signal is obtained by full-cache decoding and aggregating answer-to-visual attention over prefill image tokens. (d) Q-ViK predicts this future-conditioned utility before decoding and retains the answer-relevant visual KV entries.

focuses on fine-grained numerical evidence such as the gasoline price. Therefore, eviction based only on pre-decode saliency may remove visually small but answer-critical tokens. This motivates a different objective for LVLM KV eviction: the cache should preserve visual evidence that future answer tokens are likely to reuse, rather than tokens that are merely salient at prefill time.

To address this problem, we propose *Q-ViK*, a question-guided visual KV eviction framework based on *future answer-conditioned visual utility prediction*. During offline training, full-cache decoding is used to aggregate answer-to-visual attention from generated answer tokens to prefill visual tokens, yielding a layer-wise *future-utility signal*. We treat this signal as an offline teacher signal because it identifies the visual KV entries actually reused during generation, but requires observing the generated answer and is therefore unavailable at inference time. Q-ViK then trains a lightweight question-conditioned visual-utility scorer as a student predictor that estimates this signal from prefill representations alone. At inference time, the teacher signal and generated traces are discarded: Q-ViK invokes only the scorer once after prefill, preserves all textual KV entries, and evicts low-utility visual KV entries before autoregressive decoding. Decoding proceeds with a fixed pruned prompt KV cache, without full-cache lookahead, draft responses, pseudo future queries, repeated scorer calls, or backbone modification.

The text-preserving design follows a modality asymmetry in LVLM prompts. Text tokens encode the instruction and question, while visual tokens dominate the compressible prompt cache. Accordingly, Q-ViK keeps all text-token KVs and allocates the remaining cache budget to visual KVs. This convention makes the reported keep ratio reflect the total retained prompt KV budget, while avoiding the sharp degradation caused by joint text-visual eviction, as validated in Appendix. D.2.

Our contributions are summarized as follows:

- We identify the *future-utility mismatch* in LVLM KV eviction: pre-decode token-importance signals can fail to preserve visual tokens that become important under future answer-time attention.
- We introduce Q-ViK, a future answer-conditioned visual KV eviction framework that uses an offline *future-utility signal* as teacher supervision to train a lightweight question-conditioned visual-utility scorer.
- Experiments on held-out single-image, multi-image, and video benchmarks, as well as student-seen ones, show that Q-ViK achieves stronger top-K agreement with the future-utility signal than prefill saliency, preserves task-relevant visual evidence under aggressive cache compression, and substantially reduces KV memory and decode latency. Q-ViK maintains 53.6 vs. 53.9 full-cache average at a 20% total prompt-KV budget on LLaVA-1.5-7B and 74.1 vs. 74.5 at a 10% image-side budget on LLaVA-OneVision-7B, while reducing prompt-KV memory by up to 89% and decode latency by up to 22%.

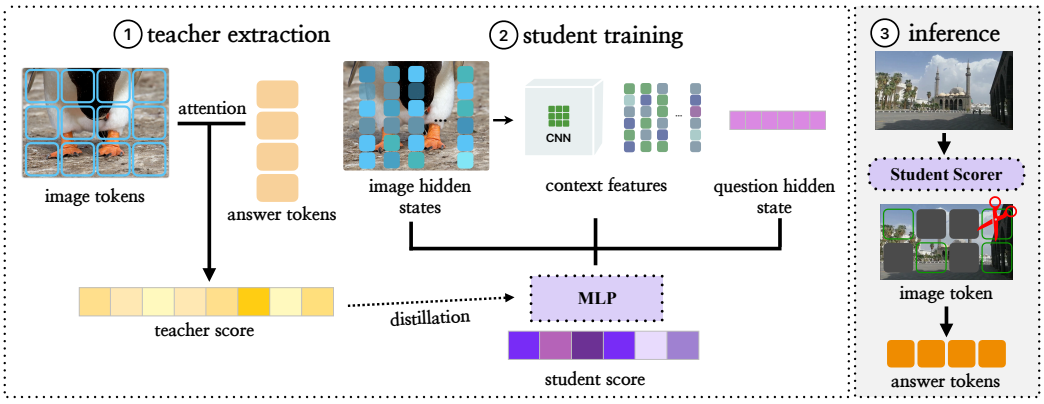

Figure 2: Overall pipeline of Q-ViK. (1) The full-cache LVLM (teacher) aggregates answer-to-image attention into layer-wise *future-utility signals*. (2) A lightweight CNN-MLP scorer (student) learns to predict these signals from image hidden states, their CNN context features, and the question hidden state. (3) At inference, only the student is run once after prefill; all text KVs are kept, and only the top-k image KVs (green) are retained for decoding.

2 Related Work

Large Vision-Language Models Large vision-language models (LVLMs) extend language models to visual inputs by converting image features into token sequences that can be processed by the language model. A common architecture consists of a visual encoder, a vision-language projector or adapter, and an autoregressive language model [13, 6, 16, 17]. The LLaVA family is a representative line of LVLMs that aligns visual features with instruction-following language models through visual instruction tuning [16, 17]. LLaVA-1.5 strengthens this design with a CLIP-ViT-L-336px visual encoder and an MLP projector, establishing a widely used open LVLM baseline [17]. LLaVA-OneVision further extends the LLaVA-style framework to single-image, multi-image, and video scenarios, using more flexible visual representations for diverse visual inputs [11].

These LVLMs expose visual information to the language model through visual tokens. This tokenized interface makes them compatible with LLM-style autoregressive decoding, but also creates an opportunity for visual-token and visual-KV compression.

KV Cache Eviction and Visual Token Pruning KV cache compression has been widely studied for efficient autoregressive inference. In LLMs, methods such as H_2O [37], StreamingLLM [32], and SnapKV [15] establish that retaining heavy-hitter, recent, or observation-window tokens preserves generation quality with a small cache. This idea has since been extended to vision-language models, where image tokens dominate the KV cache: LOOK-M reduces multimodal KV caches using text-prior attention together with cache merging [27], and PrefixKV searches adaptive layer-wise prefix configurations to allocate cache budget across layers [29]. These methods compress the cache using prompt-time attention or layer-wise budget allocation. Q-ViK differs by learning to predict visual KV utility from future answer-to-image attention.

A more recent line of work targets only image tokens, recognizing that the redundancy in LVLMs is concentrated in the visual modality. FastV [3] prunes visual tokens in early LLM layers using attention scores. SparseVLM [36] ranks visual tokens with text-vision cross attention. VisionZip [34] compresses redundant tokens at the vision-encoder interface. VFlowOpt [35] prunes image tokens using attention and patch-level entropy under a visual information flow objective. Our method belongs to this image-only line: we preserve all text-token KV entries and restrict eviction to image tokens.

3 Method

Existing KV cache eviction and token-reduction methods typically rely on attention or other pre-decode signals as token-importance proxies [37, 27, 3, 36]. However, as discussed in Sec. 1, such proxies suffer from the *future-utility mismatch*: they do not directly reveal which image tokens will

be reused by future answer tokens. We address this mismatch by using full-cache answer-to-image attention as an offline *future-utility signal* and training a lightweight per-layer question-conditioned visual-utility scorer to predict this signal from prefill representations alone. Figure 2 illustrates the overall pipeline. Section 3.2 defines the future-utility signal, Section 3.3 describes the question-conditioned visual-utility scorer and its training objective, and Section 3.4 details the inference procedure.

3.1 Preliminaries

Notation. We consider a transformer-based LVLM [26] with L layers, hidden dimension D, and N_h attention heads of head dimension $d_h = D/N_h$. For a multimodal prompt, let $X^{(l)} \in \mathbb{R}^{N_{\text{prompt}} \times D}$ denote the layer-l prefill hidden states. Let $\mathcal{I}_{\text{img}}$ and $\mathcal{I}_q$ be the image-token and question-token positions, with $N_I = |\mathcal{I}_{\text{img}}|$ and $N_Q = |\mathcal{I}_q|$. We group all non-image prompt tokens, including system, instruction, question, and separator tokens, into $\mathcal{I}_{\text{text}}$, where $N_{\text{text}} = |\mathcal{I}_{\text{text}}|$ and $N_{\text{prompt}} = N_{\text{text}} + N_I$. For a local image-token index i, $\pi_i \in \mathcal{I}_{\text{img}}$ gives its prompt position. Q-ViK predicts layer-wise image-token scores $\hat{s}^{(l)} \in \mathbb{R}^{N_I}$ used to select which visual KV entries to retain.

3.2 Future-Utility Signal

We first define a privileged teacher that measures how much each image token is used during answer generation. The teacher is available only during offline training: a frozen LVLM performs full-cache decoding, and we record answer-to-image attention from generated answer tokens to prefill image tokens.

After the full-cache LVLM generates an answer of length T, each generated answer token attends to the prefill image tokens through the decoder attention layers. For layer l and attention head $m \in \{1, \dots, N_h\}$, we denote the attention weight from the t-th generated answer token to the i-th image token as

$$a_{t,i}^{(l,m)} = \left[\text{softmax}_{p \in \mathcal{K}_t} \left(\frac{q_t^{(l,m)} k_p^{(l,m)\top}}{\sqrt{d_h}} \right) \right]_{\pi_i}, \qquad i = 1, \dots, N_I, \tag{1}$$

where $\mathcal{K}_t$ denotes the causal key set available to the t-th generated token. The softmax is taken over all available keys, and only the entry at the image-token position π_i is extracted.

We aggregate these answer-to-image attention weights over generated tokens and attention heads to obtain a layer-wise teacher score:

$$s_i^{(l),\text{teacher}} = \frac{1}{TN_h} \sum_{t=1}^{T} \sum_{m=1}^{N_h} a_{t,i}^{(l,m)}, \qquad i = 1, \dots, N_I. \tag{2}$$

The resulting vector $s^{(l),\text{teacher}} \in \mathbb{R}^{N_I}$ indicates the image-token utility revealed by full-cache answer generation.

Since the total attention assigned to image tokens varies across samples and layers, we convert the teacher score into a probability distribution:

$$y_i^{(l),\text{teacher}} = \frac{s_i^{(l),\text{teacher}}}{\sum_{j=1}^{N_I} s_j^{(l),\text{teacher}} + \epsilon}, \qquad i = 1, \dots, N_I, \tag{3}$$

where $\epsilon > 0$ is a small constant for numerical stability. This normalization preserves the relative utility ranking among image tokens while providing a stable distillation target.

Crucially, the teacher distribution $y^{(l),\text{teacher}}$ requires generating the answer with the full image KV cache and is therefore unavailable at inference time. We use it only as privileged offline supervision for the prefill-time scorer described next.

3.3 Question-Conditioned Visual-Utility Scorer

We train a lightweight scorer $f_\theta^{(l)}$ that predicts the teacher distribution $y^{(l),\text{teacher}}$ using only prefill-stage representations. The student exploits the local structure of image tokens by aggregating

neighboring visual information with a visual context encoder, and then predicts question-conditioned image-token utility.

Given the layer-l image-token representations $X_{\text{img}}^{(l)}$ and question-token representations $X_q^{(l)}$, we compute a pooled question feature, token-level visual features, and context-aware visual features:

$$\bar{q}^{(l)} = W_q \,\text{Mean}(X_q^{(l)}), \qquad \bar{x}_i^{v,(l)} = W_v x_i^{v,(l)}, \qquad \bar{C}^{(l)} = g_\phi(X_{\text{img}}^{(l)}).$$

Here, g_ϕ is a visual context encoder instantiated according to the image-token geometry of the backbone, e.g., a 2D convolutional encoder for fixed-grid image tokens or a layout-safe 1D convolutional encoder for AnyRes-style token sequences [11]. Let $\bar{c}_i^{(l)} = \bar{C}^{(l)}[i,:]$ denote the context-aware feature for image token i.

To make the score question-conditioned, we fuse token-level visual features, context-aware visual features, and the pooled question feature using concatenation and element-wise interaction terms:

$$\hat{s}_i^{(l)} = \text{MLP}_\psi\left([\bar{x}_i^{v,(l)}; \bar{c}_i^{(l)}; \bar{q}^{(l)}; \bar{x}_i^{v,(l)} \odot \bar{q}^{(l)}; \bar{c}_i^{(l)} \odot \bar{q}^{(l)}]\right), \qquad i = 1, \ldots, N_I.$$

The element-wise products encode question-dependent relevance at both token and local-context levels. This yields student scores $\hat{s}^{(l)} \in \mathbb{R}^{N_I}$ over image tokens.

We normalize $\hat{s}^{(l)}$ with a softmax over image tokens to obtain $\hat{y}^{(l)}$, and train the student with a distribution regression loss and a top-vs-bottom ranking loss:

$$\mathcal{L}^{(l)} = \text{MSE}\left(\hat{y}^{(l)},\, y^{(l),\text{teacher}}\right) + \lambda_{\text{rank}}\, \mathcal{L}_{\text{rank}}^{(l)}. \tag{4}$$

The MSE term aligns the predicted utility distribution with the teacher distribution. For the ranking term, we select the top-α and bottom-β fractions of image tokens under the teacher distribution:

$$\mathcal{I}_{\text{top}}^{(l)} = \text{TopK}_{\lceil \alpha N_I \rceil}\left(y^{(l),\text{teacher}}\right), \qquad \mathcal{I}_{\text{bot}}^{(l)} = \text{BottomK}_{\lceil \beta N_I \rceil}\left(y^{(l),\text{teacher}}\right). \tag{5}$$

We then apply a pairwise margin loss on the normalized student distribution:

$$\mathcal{L}_{\text{rank}}^{(l)} = \frac{1}{|\mathcal{I}_{\text{top}}^{(l)}||\mathcal{I}_{\text{bot}}^{(l)}|} \sum_{i \in \mathcal{I}_{\text{top}}^{(l)}} \sum_{j \in \mathcal{I}_{\text{bot}}^{(l)}} \max\left(0,\, m - \left(\hat{y}_i^{(l)} - \hat{y}_j^{(l)}\right)\right). \tag{6}$$

This term penalizes cases where teacher-important image tokens are not assigned sufficiently larger predicted utilities than teacher-unimportant tokens.

3.4 Inference-time Eviction

Algorithm 1 summarizes the overall procedure, including offline teacher construction, student training, and inference-time visual KV eviction. At inference time we cannot use the full-cache teacher of Section 3.2, since $y^{(l),\text{teacher}}$ is only available after answer generation. Instead, Q-ViK computes the student score $\hat{s}^{(l)}$ once at the end of the prefill stage and uses it to prune the prompt KV cache before decoding. Given an input image and question, the LVLM first performs a full-prompt prefill forward pass. At each layer l, we obtain the image-token hidden states $X_{\text{img}}^{(l)}$ and the question-token hidden states $X_q^{(l)}$. The trained student $f_\theta^{(l)}$ then predicts a raw utility score $\hat{s}^{(l)} \in \mathbb{R}^{N_I}$ from these prefill representations.

Algorithm 1: Q-ViK Training and Inference

Require: dataset $\mathcal{D}$, frozen LVLM F, student f_θ
Require: eviction layers $\mathcal{S}$, retain budget k_I
Ensure: trained scorer and pruned prompt KV cache

// Offline training
for each $(v, q, a) \in \mathcal{D}$ **do**
  Run full-cache decoding with F
  Build teacher target $y^{(l),\text{teacher}}$
  Predict $\hat{y}^{(l)}$ from prefill features
  Update f_θ using Eq. (4)
end for
// Inference-time eviction
Run prefill on a test prompt
for each $l \in \mathcal{S}$ **do**
  Predict image-token scores $\hat{s}^{(l)}$
  $\mathcal{R}^{(l)} \leftarrow \text{TopK}(\hat{s}^{(l)}, k_I)$
  Keep all text KVs and selected image KVs
end for
Decode with the pruned KV cache

Keep-ratio convention. Q-ViK always preserves text-token KVs and applies eviction only to image-token KVs. Unless otherwise specified, we report the retain ratio over the total prompt KV

cache. Under this default protocol, all text KVs are retained first, and the remaining budget is assigned to image KVs. For image-only comparison protocols, we may instead report an image-side retain ratio to match the native budget convention of image-only baselines. This changes only the evaluation budget convention, not the Q-ViK scoring mechanism: the student always scores image tokens, all text KVs are preserved, and eviction candidates are restricted to image KVs.

4 Experiments

Experimental Setup Full dataset descriptions and evaluation protocol details are provided in Appendix A. The student is trained on TextVQA [24], ScienceQA [19], and GQA [10] with teacher scores from full-cache decoding. We evaluate on student-seen benchmarks (ScienceQA-IMG (SQA^I), TextVQA, and GQA) and on held-out benchmarks covering evidence-oriented VQA and captioning (DocVQA [21], ChartQA [20], COCO Captions [5], NoCaps [1], and TextCaps [23]) as well as general perception, reasoning, and hallucination (MME [8], MMBench-EN [18], MMStar [4], POPE [14], and VizWiz [9]). We use LLaVA-1.5-7B [17] and LLaVA-OneVision-7B [11] as backbones. On LLaVA-1.5-7B, we compare against the KV eviction methods LOOK-M [27] and PrefixKV [29] and the token pruning methods FastV [3] and SparseVLM [36] under the total prompt-KV protocol, where Retain is defined over all prompt KVs and Q-ViK preserves all text KVs while assigning the remaining budget to image KVs. On LLaVA-OneVision-7B, we compare against VisionZip [34] and VFlowOpt [35] under the image-only protocol, where $\text{Retain}_{\text{img}}$ denotes the image-side keep ratio and all text tokens/KVs are kept unchanged. *Future Oracle* is a privileged non-deployable reference that selects visual KVs using future answer-time attention from full-cache decoding; the generated full-cache answer is discarded, and generation is re-run with the pruned prompt KV cache. For multi-image and video generalization, we use generation subsets of MileBench [25] (CLEVR-Change, IEdit, and Spot-the-Diff), SEED-Bench-video [12], and Video-MME [7], and evaluate them on LLaVA-OneVision-7B under the image-only protocol..

4.1 Task Performance under Total Prompt-KV Compression

We first evaluate on LLaVA-1.5-7B under the total prompt-KV protocol at 50% and 20% retain ratios. As shown in Tab. 1, Q-ViK achieves the best average among deployable methods at both ratios. At 50% retain, Q-ViK matches Full Cache (53.9) and outperforms SparseVLM (53.3), FastV (52.8), LOOK-M (52.1), and PrefixKV (49.7). The gap widens under the tighter budget: at 20% retain, Q-ViK stays at 53.6, whereas LOOK-M drops to 50.8 and SparseVLM, FastV, and PrefixKV collapse to 38.3, 35.4, and 34.3. Token pruning methods degrade sharply on evidence-heavy benchmarks (e.g., FastV: 19.5 on TextVQA and 11.4 on DocVQA), and PrefixKV fails on TextVQA (10.8), POPE (18.0), and VizWiz (5.9), suggesting that pre-decode saliency or layer-wise allocation can discard answer-critical evidence under aggressive compression. Q-ViK also remains close to Full Cache on held-out benchmarks and tracks the privileged Future Oracle (53.9 and 53.7 at 50% and 20%), recovering most of its selection benefit from prefill alone. Removing all visual KVs after prefill instead lowers the average from 45.2 to 38.4 on the evidence-oriented benchmarks (Appendix D.4), confirming that the retained visual evidence remains necessary during decoding.

4.2 Image-Only Visual Compression on LLaVA-OneVision

We next evaluate on LLaVA-OneVision and compare against image-only compression methods (VisionZip, VFlowOpt). Since these baselines define their budget over image tokens only, we adopt an image-side keep ratio $\text{Retain}_{\text{img}}$ for fair comparison; all text tokens/KVs remain unchanged and compression is applied only to image tokens or image KVs at each method's native intervention stage. We report 50%, 25%, and 10% image retain ratios.

As shown in Tab. 2, Q-ViK achieves the best average across retain ratios, with the largest gains under aggressive compression. At 50% image retain, Q-ViK reaches 74.6 average, matching Full Cache (74.5), compared with 73.9 for VFlowOpt and 73.3 for VisionZip. At 10% retain, Q-ViK still achieves 74.1, whereas VFlowOpt and VisionZip drop to 62.6 and 60.1. The gap is especially large on evidence-focused tasks: VFlowOpt loses 16.8 points on TextVQA and 35.0 points on DocVQA, while Q-ViK maintains 75.1 and 86.8, respectively. These results suggest that future-utility prediction is more effective at selecting answer-relevant visual tokens under tight KV budgets.

Table 1: Task performance on LLaVA-1.5-7B under the total prompt-KV compression protocol. Retain denotes the total prompt-KV keep ratio. This table compares methods evaluated under a total prompt-KV budget; Q-ViK preserves all text KVs and assigns the remaining budget to image KVs. Future Oracle is a privileged non-deployable baseline that selects image KVs using full-cache answer-conditioned utility. COCO, NoCaps, and TextCaps are evaluated with ROUGE-L, while the remaining QA benchmarks use their task-specific metrics. Avg. is the arithmetic mean of scores over all benchmarks; for averaging only, MME is rescaled to a 0–100 range by dividing by 20. **Red** and **blue** denote the best and second-best results among compressed methods at each retain ratio.

Method	Student-seen			Student-unseen									Avg.
	SQA^{I}	TextVQA	GQA	DocVQA	ChartQA	COCO	NoCaps	TextCaps	MME	MMB^{EN}	MMStar	POPE	VizWiz	
Retain = 100%														
Full Cache	69.3	45.9	62.0	28.1	17.9	55.5	59.4	47.4	1514.7	64.5	33.5	86.7	54.2	53.9
Retain = 50%														
Future Oracle	69.2	45.9	62.0	28.2	18.0	55.4	59.3	47.3	1518.8	64.3	33.6	86.8	54.3	53.9
FastV	69.0	44.7	59.1	23.8	**17.7**	**55.6**	**59.3**	**47.6**	1482.7	63.7	33.3	83.7	**54.8**	52.8
SparseVLM	68.5	**45.5**	60.7	**27.1**	17.4	55.5	58.8	47.4	1459.9	63.7	**34.8**	**86.9**	53.7	**53.3**
LOOK-M	**69.4**	37.1	**61.9**	23.9	15.7	54.5	56.9	43.5	**1521.8**	**64.0**	33.1	86.7	**54.8**	52.1
PrefixKV	67.1	40.5	58.3	26.1	12.9	55.2	58.6	47.2	**1508.0**	60.0	**34.7**	86.4	23.4	49.7
Ours	**69.3**	**46.2**	**61.9**	**28.1**	**18.2**	**55.9**	**59.4**	**47.6**	1506.9	**64.3**	33.2	**86.8**	54.7	**53.9**
Retain = 20%														
Future Oracle	69.4	45.9	62.0	28.0	18.0	55.5	59.2	47.0	1502.5	64.7	33.0	86.9	54.0	53.7
FastV	64.3	19.5	42.0	11.4	12.4	40.9	44.6	36.5	762.2	20.6	22.6	57.4	50.5	35.4
SparseVLM	63.0	22.8	46.2	12.3	12.7	43.5	45.2	37.0	936.7	22.9	23.2	72.7	49.2	38.3
LOOK-M	**69.2**	**33.1**	**61.6**	**19.3**	**15.2**	51.9	53.6	39.9	**1513.0**	**64.0**	**34.5**	**86.3**	**55.8**	**50.8**
PrefixKV	52.4	10.8	33.8	14.5	6.2	**54.2**	**57.2**	**45.1**	1428.2	43.1	**33.2**	18.0	5.9	34.3
Ours	**69.2**	**45.8**	**61.9**	**26.5**	**18.0**	**56.0**	**58.7**	**47.1**	**1504.1**	**64.4**	33.1	**86.5**	**54.6**	**53.6**

NoCaps is an exception: VisionZip and VFlowOpt sometimes exceed Q-ViK and the full-cache baseline, likely because broad visual coverage benefits open-ended captioning more than answer-conditioned evidence selection.

4.3 Future-Utility Alignment under Eviction

We further analyze whether Q-ViK reduces the future-utility mismatch discussed in Sec. 1. Since KV eviction is a top-K retention problem, we measure Jaccard@K between each method's retained image-token set and the teacher's top-K set, where the teacher ranks image tokens by the attention they receive from the full-cache answer. As shown in Figs. 3(a) and 3(c), Q-ViK achieves higher Jaccard@K than prefill saliency at every keep ratio, indicating that it retains more of the visual evidence the full-cache model actually uses when answering.

Figures 3(b) and 3(d) relate the mismatch, defined as $1 - \mathrm{Jaccard@}K$, to full-response preservation, i.e., the fraction of answers under eviction that match the full-cache answer (Full Cache = 1.0). As the keep ratio decreases, the mismatch grows for both methods, but prefill saliency shows a larger increase in mismatch and deviates further from the full-cache answer. At a 20% keep ratio, for instance, prefill saliency preserves only 70.3% and 91.3% of full-cache answers on LLaVA-1.5-7B and LLaVA-OneVision-7B, respectively, whereas Q-ViK preserves 77.3% and 94.0%. Together with Tab. 1, this indicates that Q-ViK learns to retain the visual evidence used during answer generation rather than following prompt-time saliency, which explains why it closely tracks both Full Cache and the Future Oracle.

Qualitative analysis. Fig. 4 visualizes the image tokens retained after eviction at the same budget, compared with those selected by the teacher (Future Oracle). For the question "What piece of furniture is it?" (GT: bed), Q-ViK concentrates more on the bed region than prefill saliency and closely resembles the teacher's selection. Additional examples are provided in Appendix D.1.

Table 2: LLaVA-OneVision results under the image-only visual compression protocol. $\text{Retain}_{\text{img}}$ denotes the fraction of image tokens/KVs retained; all text tokens/KVs are preserved. TextCaps and NoCaps are evaluated with ROUGE-L, while TextVQA, DocVQA, and ChartQA use their task-specific QA metrics. MME reports the perception score, and MMB_{EN}, MMStar, POPE, VizWiz, and SQA^{I} use their standard evaluation metrics. Avg. is computed over all 11 benchmarks, with MME rescaled to a 100-point scale (score/20). **Red** and **blue** denote the best and second-best results among compressed methods at each retention ratio.

Method	Student-seen		Student-unseen									Avg.
	SQA^{I}	TextVQA	DocVQA	ChartQA	NoCaps	TextCaps	MME	MMB_{EN}	MMStar	POPE	VizWiz	
Retain$_{\text{img}}$ = 100%												
Full Cache	95.9	75.9	90.1	80.0	55.3	52.4	1539.0	80.1	61.8	90.3	60.4	74.5
Retain$_{\text{img}}$ = 50%												
VisionZip	94.1	73.3	**88.1**	73.2	**61.7**	48.2	**1586.9**	80.4	58.1	**89.2**	**61.2**	73.3
VFlowOpt	**95.2**	**74.5**	88.0	**77.3**	**57.3**	**51.2**	**1570.3**	**81.0**	**60.9**	**89.3**	60.0	**73.9**
Ours	**95.9**	**76.2**	**90.1**	**80.2**	55.1	**52.6**	1570.2	**80.8**	**62.0**	89.1	**60.3**	**74.6**
Retain$_{\text{img}}$ = 25%												
VisionZip	90.1	67.1	79.3	62.5	**61.1**	45.1	**1575.3**	79.0	54.1	**89.1**	**60.4**	69.7
VFlowOpt	**92.2**	**72.5**	**82.3**	**68.9**	**59.2**	**45.5**	1558.5	**79.6**	**57.3**	88.7	**60.9**	**71.4**
Ours	**95.8**	**75.8**	**89.2**	**80.2**	55.1	**52.4**	**1569.0**	**80.8**	**61.9**	**89.1**	60.3	**74.5**
Retain$_{\text{img}}$ = 10%												
VisionZip	86.5	51.5	45.5	34.3	**58.2**	**42.7**	1433.9	**75.1**	50.0	84.9	**60.8**	60.1
VFlowOpt	**87.6**	**59.1**	**55.1**	**52.6**	51.9	41.1	**1465.4**	74.4	**51.3**	**85.0**	57.3	**62.6**
Ours	**95.8**	**75.1**	**86.8**	**79.5**	**54.8**	**52.3**	**1565.1**	**80.9**	**61.9**	**88.9**	**60.4**	**74.1**

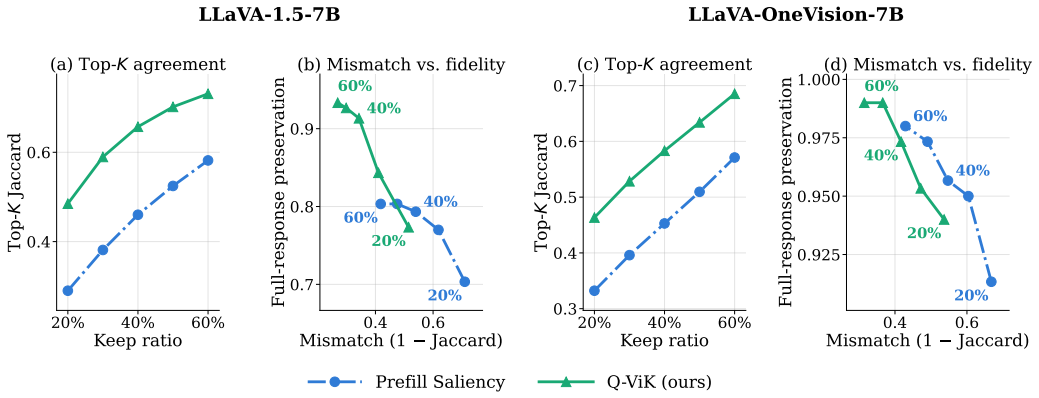

Figure 3: **Alignment with full-cache future utility.** Panels (a, b) show LLaVA-1.5-7B, and panels (c, d) show LLaVA-OneVision-7B. Panels (a, c) report Jaccard@K between retained image tokens and the teacher's top-K set. Panels (b, d) relate mismatch to full-response preservation (Full Cache = 1.0). For both backbones, keep ratios are defined over the total prompt KV cache, with all text KVs preserved. As the keep ratio decreases, prefill saliency shows a larger mismatch and lower preservation than Q-ViK.

4.4 Latency and Memory Efficiency

Tab. 3 reports decode latency and prompt KV-cache memory on DocVQA samples, together with the average performance from Tab. 1 and Tab. 2. On LLaVA-1.5-7B, Q-ViK reduces prompt-KV memory by 80% at a 20% retain ratio while keeping decode latency on par with Full Cache (28.0 vs. 28.6 ms/tok) and the average performance nearly unchanged (53.6 vs. 53.9). Under the same KV budget, the baselines achieve averages from 34.3 (PrefixKV) to 50.8 (LOOK-M). On LLaVA-OneVision-7B, Q-ViK reduces KV memory by 75% and decode latency by 20% at a 25% retain ratio while matching the average performance of Full Cache (74.5), and by 89% and 22%, respectively, at 10%. Its efficiency is comparable to that of VFlowOpt at the same budget, while its average performance is

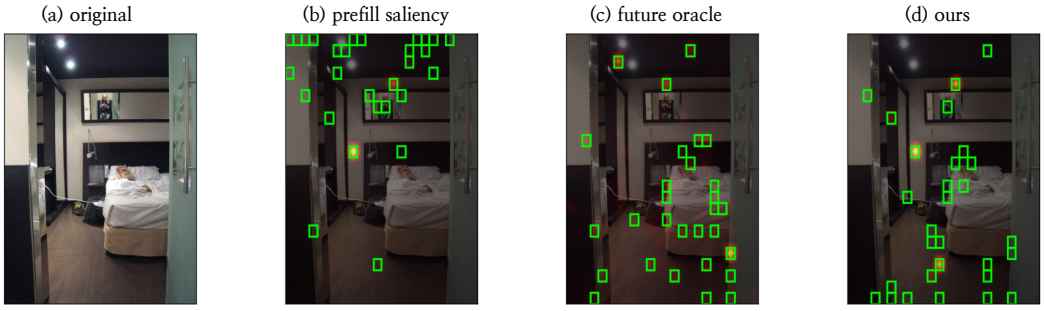

Figure 4: **Post-eviction retained tokens.** GQA example: "What piece of furniture is it?" (GT: bed). Green boxes denote retained image tokens after eviction at the same budget. Q-ViK concentrates more on the bed region than prefill saliency, closely tracking the teacher (Future Oracle).

Table 3: Latency and memory efficiency. Decode latency and prompt KV-cache memory are measured on 100 DocVQA samples (single RTX A6000, batch size 1, greedy decoding); Avg. is the average performance in Tab. 1 and Tab. 2, and retain ratios follow each table's protocol. Baselines run in the environments required by their released code, so their latency also reflects implementation differences.

Method	Latency ↓ (ms/tok)	KV Cache ↓ (MB)	Avg. ↑	Latency ↓ (ms/tok)	KV Cache ↓ (MB)	Avg. ↑
LLaVA-1.5-7B	*Retain* = 50%			*Retain* = 20%		
Full Cache	28.6	320.7	53.9	28.6	320.7	53.9
FastV	39.2 (+37%)	160.3 (−50%)	52.8	40.2 (+41%)	64.2 (−80%)	35.4
SparseVLM	45.1 (+58%)	160.3 (−50%)	53.3	45.2 (+58%)	64.4 (−80%)	38.3
LOOK-M	83.1 (+191%)	159.9 (−50%)	52.1	84.8 (+197%)	63.6 (−80%)	50.8
PrefixKV	43.0 (+50%)	160.6 (−50%)	49.7	38.8 (+36%)	64.3 (−80%)	34.3
Ours	**29.6** (+4%)	160.3 (−50%)	**53.9**	**28.0** (−2%)	64.2 (−80%)	**53.6**
LLaVA-OneVision-7B	*Retain*$_{\text{img}}$ = 25%			*Retain*$_{\text{img}}$ = 10%		
Full Cache	36.9	400.8	74.5	36.9	400.8	74.5
VisionZip	33.9 (−8%)	288.4 (−28%)	69.7	30.1 (−19%)	129.4 (−68%)	60.1
VFlowOpt	30.0 (−19%)	102.4 (−74%)	71.4	**27.9** (−24%)	42.5 (−89%)	62.6
Ours	**29.5** (−20%)	101.9 (−75%)	**74.5**	28.7 (−22%)	42.2 (−89%)	**74.1**

substantially higher (74.1 vs. 62.6 at 10%). The larger latency gain on LLaVA-OneVision-7B reflects its much longer visual context (thousands of image tokens versus 576 in LLaVA-1.5-7B).

4.5 Multi-image and Video Generalization

We further evaluate whether the single-image-trained student transfers to multi-image and video inputs on LLaVA-OneVision-7B at 10% image retain under the image-only protocol.

Multi-image. We use three ROUGE-L generation subsets from MileBench [25] (CLEVR-Change, IEdit, and Spot-the-Diff). As shown in Tab. 4, Q-ViK leads on all three subsets and matches or slightly exceeds Full Cache (21.6, 30.6, and 27.3 vs. 21.3, 29.9, and 26.9), whereas VFlowOpt and VisionZip degrade substantially on CLEVR-Change (16.5 and 18.8) and IEdit (25.8 and 24.2). Full category-level MileBench results are reported in Appendix C.4.

Video. We use SEED-Bench-video [12] and Video-MME [7], with Video-MME split by duration. As shown in Tab. 5, Q-ViK again leads on all splits and stays within 0.3 points of Full Cache, whereas VisionZip and VFlowOpt drop by 6.0 and 9.0 points on the Video-MME Short split, respectively.

Table 4: Multi-image generation (ROUGE-L) on MileBench with LLaVA-OneVision under image-side visual compression. $\text{Retain}_{\text{img}}$ denotes the image-side keep ratio; all text tokens/KVs are preserved.

$\text{Retain}_{\text{img}}$	Method	CLEVR	IEdit	Spot-Diff
100%	Full Cache	21.3	29.9	26.9
10%	VisionZip	**18.8**	24.2	**25.9**
	VFlowOpt	16.5	**25.8**	25.4
	Ours	**21.6**	**30.6**	**27.3**

Table 5: Video generalization on SEED-Bench-video and Video-MME with LLaVA-OneVision under the same image-side compression protocol. Video-MME is split by duration (Short / Medium / Long).

$\text{Retain}_{\text{img}}$	Method	SEED	Video-MME		
		Video	Short	Med.	Long
100%	Full Cache	57.0	70.8	54.7	49.7
10%	VisionZip	**56.5**	**64.8**	**53.4**	46.3
	VFlowOpt	55.4	61.8	53.0	**48.3**
	Ours	**56.9**	**70.9**	**54.6**	**49.4**

Together, these results provide zero-shot evidence that the future-utility scorer generalizes beyond single-image training to multi-image and video inputs.

5 Conclusion

In this paper, we present Q-ViK, a future-utility-guided visual KV eviction framework for efficient generation in large vision-language models. Unlike existing methods that rely on pre-decode saliency or visual redundancy, Q-ViK directly targets the central question in LVLM cache compression: which visual KV entries will actually be reused by future answer tokens? It learns this signal by distilling full-cache answer-to-image attention into a lightweight question-conditioned student, which predicts future answer-relevant visual utility using only prefill-stage representations. At inference time, Q-ViK requires only a single post-prefill scoring step and performs visual KV eviction without lookahead decoding, draft generation, iterative scoring, or any modification to the backbone LVLM. Extensive experiments on single-image, held-out, multi-image benchmarks and video benchmarks show that Q-ViK remains close to privileged future-utility selection even under aggressive compression, while substantially reducing KV memory and decoding latency. These findings demonstrate that effective LVLM KV eviction should be guided by future answer-conditioned evidence rather than prompt-time saliency alone, and establish Q-ViK as a practical, deployable approach for high-fidelity compressed LVLM inference.

Limitations. Q-ViK preserves all text-token KVs, which bounds total cache reduction for text-heavy inputs; sequentially applying text-KV eviction further reduces memory but degrades text-heavy long-context tasks (Appendix D.3). In addition, the scorer must be trained for each backbone, although this one-time adaptation is lightweight, taking only 3.2 and 9.9 GPU-hours on a single RTX 3090 for LLaVA-1.5-7B and LLaVA-OneVision-7B, respectively (Appendix B.2). Q-ViK also underperforms on MileBench needle-in-a-haystack retrieval (Appendix C.4).

Acknowledgements

This work was supported by Hyundai Motor Company and Kia, and partially support of "Cooperative Research Program for Agriculture Science and Technology Development (Project No. RS-2024-00332198)" Rural Development Administration, Republic of Korea, and supported by the Institute of Information Communications Technology Planning Evaluation(IITP) grant funded by the Korea government(MSIT) (o.RS-2025-02219317, AI Star Fellowship(Kookmin University)), and the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (RS-2025-23524783).

References

[1] Harsh Agrawal, Karan Desai, Yufei Wang, Xinlei Chen, Rishabh Jain, Mark Johnson, Dhruv Batra, Devi Parikh, Stefan Lee, and Peter Anderson. Nocaps: Novel object captioning at

scale. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8948–8957, 2019.

[2] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35: 23716–23736, 2022.

[3] Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang. An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models. In *European Conference on Computer Vision*, pages 19–35. Springer, 2024.

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

[5] Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*, 2015.

[6] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale N Fung, and Steven Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning. *Advances in neural information processing systems*, 36:49250–49267, 2023.

[7] Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, et al. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 24108–24118. IEEE, 2025.

[8] Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, et al. Mme: A comprehensive evaluation benchmark for multimodal large language models. *Advances in Neural Information Processing Systems*, 38, 2026.

[9] 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 *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3608–3617. IEEE, 2018.

[10] 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*, pages 6700–6709, 2019.

[11] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, et al. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024.

[12] Bohao Li, Yuying Ge, Yixiao Ge, Guangzhi Wang, Rui Wang, Ruimao Zhang, and Ying Shan. Seed-bench: Benchmarking multimodal large language models. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13299–13308, 2024. doi: 10.1109/CVPR52733.2024.01263.

[13] 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*, pages 19730–19742. PMLR, 2023.

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

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

[16] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 34892–34916. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/6dcf277ea32ce3288914faf369fe6de0-Paper-Conference.pdf.

[17] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 26296–26306, June 2024.

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

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

[20] Ahmed Masry, Xuan Long Do, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. Chartqa: A benchmark for question answering about charts with visual and logical reasoning. In *Findings of the association for computational linguistics: ACL 2022*, pages 2263–2279, 2022.

[21] 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 (WACV)*, pages 2200–2209, January 2021.

[22] 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 (ICCV)*, pages 22857–22867, October 2025.

[23] Oleksii Sidorov, Ronghang Hu, Marcus Rohrbach, and Amanpreet Singh. Textcaps: a dataset for image captioning with reading comprehension. In *European conference on computer vision*, pages 742–758. Springer, 2020.

[24] 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 (CVPR)*, June 2019.

[25] Dingjie Song, Shunian Chen, Guiming Hardy Chen, Fei Yu, Xiang Wan, and Benyou Wang. Milebench: Benchmarking mllms in long context. *arXiv preprint arXiv:2404.18532*, 2024.

[26] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

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

[28] Zhongwei Wan, Hui Shen, Xin Wang, Che Liu, Zheda Mai, and Mi Zhang. MEDA: Dynamic KV cache allocation for efficient multimodal long-context inference. In Luis Chiruzzo, Alan Ritter, and Lu Wang, editors, *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 2485–2497, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-189-6. doi: 10.18653/v1/2025.naacl-long.125. URL https://aclanthology.org/2025.naacl-long.125/.

[29] Ao Wang, Hui Chen, Jianchao Tan, Kefeng Zhang, Xunliang Cai, Zijia Lin, Jungong Han, and Guiguang Ding. PrefixKV: Adaptive prefix KV cache is what vision instruction-following models need for efficient generation. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026. URL https://openreview.net/forum?id=tDG6bY48ch.

[30] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024.

[31] Weiyun Wang, Shuibo Zhang, Yiming Ren, Yuchen Duan, Tiantong Li, Shuo Liu, Mengkang Hu, Zhe Chen, Kaipeng Zhang, Lewei Lu, et al. Needle in a multimodal haystack. *Advances in Neural Information Processing Systems*, 37:20540–20565, 2024.

[32] Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=NG7sS51zVF.

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

[34] 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 (CVPR)*, pages 19792–19802, June 2025.

[35] Sihan Yang, Runsen Xu, Chenhang Cui, Tai Wang, Dahua Lin, and Jiangmiao Pang. Vflowopt: A token pruning framework for lmms with visual information flow-guided optimization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 23924–23934, 2025.

[36] 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 *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=80faIPZ67S.

[37] Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Re, Clark Barrett, Zhangyang Wang, and Beidi Chen. H2o: Heavy-hitter oracle for efficient generative inference of large language models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=RkRrPp7GKO.

[38] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*, 2023.

Table 6: Datasets and evaluation metrics used in Q-ViK experiments. "Train / seen" datasets are used for scorer training and reported as student-seen benchmarks; "Held-out" datasets are used only for evaluation. For ScienceQA, we evaluate on its image subset (SQA^I). TextNIAH denotes the TextNeedleInAHaystack subset of MileBench, and MM-NIAH covers retrieval, counting, and reasoning over long interleaved image–text documents. SEED-Bench-video denotes the video portion of SEED-Bench, and Video-MME is reported by video duration (Short / Medium / Long).

Dataset	Role	Input type	Task type	Metric
TextVQA [24]	Train / seen	Single image	OCR-aware VQA	VQA score
ScienceQA [19]	Train / seen	Single image	Science QA	Accuracy
GQA [10]	Train / seen	Single image	Compositional VQA	Accuracy
DocVQA [21]	Held-out	Single image	Document VQA	ANLS
ChartQA [20]	Held-out	Single image	Chart QA	Relaxed accuracy
COCO Captions [5]	Held-out	Single image	Captioning	ROUGE-L
NoCaps [1]	Held-out	Single image	Novel-object captioning	ROUGE-L
TextCaps [23]	Held-out	Single image	OCR-aware captioning	ROUGE-L
MME [8]	Held-out	Single image	Perception (yes/no)	Perception score
MMBench-EN [18]	Held-out	Single image	Multiple-choice QA	Accuracy
MMStar [4]	Held-out	Single image	Vision-centric QA	Accuracy
POPE [14]	Held-out	Single image	Object hallucination	F1
VizWiz [9]	Held-out	Single image	Real-world VQA	VQA score
MileBench subsets [25]	Held-out	Multi-image	Multi-image generation	ROUGE-L
MileBench TextNIAH	Held-out	Multi-image	Long-context retrieval	Accuracy
MM-NIAH [31]	Held-out	Interleaved	Long-context reasoning	Accuracy
SEED-Bench-video [12]	Held-out	Video	Video multiple-choice QA	Accuracy
Video-MME [7]	Held-out	Video	Video multiple-choice QA	Accuracy

A Datasets and Evaluation Protocols

A.1 Datasets and Evaluation Metrics

Table 6 summarizes the datasets and metrics used in our experiments. The student scorer is trained on TextVQA, ScienceQA, and GQA. Accordingly, TextVQA, GQA, and ScienceQA-IMG (SQA^I) are reported as student-seen benchmarks. All remaining benchmarks are held out from scorer training: DocVQA, ChartQA, COCO Captions, NoCaps, and TextCaps for evidence-oriented VQA and captioning; MME, MMBench-EN (MMB_{EN}), MMStar, POPE, and VizWiz for general multimodal perception and reasoning; MileBench subsets and MM-NIAH for multi-image and long interleaved multimodal contexts; and SEED-Bench-video and Video-MME for video understanding.

A.2 Evaluation Protocols

We use three evaluation protocols according to the compression target.

Total prompt-KV compression. For LLaVA-1.5-7B, we report results under the total prompt-KV compression protocol. In this setting, the retain ratio is defined over all prompt KVs, including both text and image KVs. Q-ViK preserves all text KVs and assigns the remaining budget to image KVs.

Image-only visual compression. For LLaVA-OneVision-7B, we report image-side retain ratios when comparing with image-only visual compression methods. In this setting, all text tokens or text KVs are kept unchanged, and compression is applied only to image tokens or image KVs according to each method's native intervention stage. This protocol allows image-only methods such as VisionZip and VFlowOpt to be evaluated under their native compression target while comparing Q-ViK under the same visual budget. The MileBench and video experiments also follow this protocol.

Sequential visual–text eviction. To examine compatibility with text-KV eviction (Appendix D.3), we apply a two-stage pipeline to LLaVA-OneVision immediately after prefill. Let N_{img} and N_{txt}

Table 7: One-time backbone adaptation cost on a single NVIDIA RTX 3090. No backbone parameters are updated.

Backbone	Teacher gen. (GPU-h)	Student training (GPU-h)	Total (GPU-h)	Peak VRAM (GiB)
LLaVA-1.5-7B	0.210	2.942	3.152	16.28
LLaVA-OneVision-7B	0.612	9.300	9.912	22.14

denote the numbers of image and text KVs in the prompt, and let r_{img} and r_{txt} denote the image and text keep ratios. In Stage 1, Q-ViK scores the image KVs and retains the top-$\lfloor r_{\mathrm{img}} N_{\mathrm{img}} \rfloor$ entries. In Stage 2, StreamingLLM [32] is applied to the text KVs and retains $\lfloor r_{\mathrm{txt}} N_{\mathrm{txt}} \rfloor$ entries consisting of attention-sink and recent tokens. Because the two stages operate on disjoint token sets, no structural modification to either method is required.

B Implementation Details

B.1 Teacher Score Extraction

Teacher scores are collected offline from full-cache decoding. For each training sample, we run the frozen LVLM with the full prompt KV cache and record answer-to-image attention from generated answer tokens to prefill image tokens. The per-layer teacher score $s_i^{(l),\mathrm{teacher}}$ and its sum-normalized distribution $y_i^{(l),\mathrm{teacher}}$ are computed as defined in Eq. (2) and Eq. (3), respectively. This teacher is privileged because it requires full-cache answer generation and is therefore unavailable at inference time.

B.2 One-time Adaptation Cost

Q-ViK requires a one-time adaptation per backbone, consisting of offline teacher generation and student scorer training; the backbone itself is kept frozen. Table 7 reports the cost on a single NVIDIA RTX 3090 (24 GB). The total offline cost is 3.2 and 9.9 GPU-hours for LLaVA-1.5-7B and LLaVA-OneVision-7B, respectively, and peak memory remains within a single 24 GB GPU. The teacher is discarded after training, so Q-ViK supports frozen LVLMs through lightweight adaptation rather than universal zero-shot transfer across checkpoints.

B.3 Hardware and Measurement Setup

The adaptation cost (Table 7) and the scorer size study (Table 10) are measured on a single NVIDIA RTX 3090 (24 GB), the sequential visual–text eviction experiments (Table 12) on a single NVIDIA RTX 4090 (24 GB), and the latency and memory comparison (Tab. 3) on a single NVIDIA RTX A6000. Latency is reported as decoding latency, i.e., the average wall-clock time per generated token.

C Additional Results

C.1 Correctness-Filtered Training

We examine whether training the student only on samples that the full-cache LVLM answers correctly improves eviction. As shown in Tab. 8, correctness filtering lowers performance on all three tasks, by 2.5 points on average. Final-answer correctness is a coarse proxy for the utility of an attention trace: an incorrect answer may still attend to relevant regions or partially valid evidence, whereas hard filtering removes difficult samples and diverse attention patterns. Note that Q-ViK aims to preserve the frozen full-cache LVLM's future visual-KV access rather than to correct its hallucinations, so erroneous traces are not excluded from training.

Table 8: Correctness-filtered training on LLaVA-1.5-7B at 20% total prompt-KV retention. Q-ViK results are from Tab. 1.

Method	TextVQA	DocVQA	ChartQA	Avg.
Correctness-filtered	43.3	23.0	16.4	27.6
Q-ViK	45.8	26.5	18.0	30.1

Table 9: Scorer architecture ablation on LLaVA-1.5-7B under the total prompt-KV protocol. CNN+MLP denotes the full Q-ViK scorer. Results come from a separate controlled run and may differ slightly from Tab. 1.

Keep ratio	Scorer	ChartQA	COCO caps
0.5	MLP-only	18.0	**55.9**
	CNN-only	18.1	55.8
	CNN+MLP	**18.2**	**55.9**
0.2	MLP-only	17.7	55.5
	CNN-only	17.9	55.2
	CNN+MLP	**18.0**	**56.0**

C.2 Ablation on Scorer Architecture

We ablate the scorer architecture to measure the contribution of local visual context and question-conditioned fusion. The MLP-only variant scores each image token using token-level features without local spatial context. The CNN-only variant uses local visual context but removes the full fusion used by Q-ViK. The full Q-ViK scorer combines token-level visual features, local context features, and question-conditioned interaction terms. Table 9 reports the results. CNN-only improves over MLP-only on ChartQA, suggesting that local visual context helps identify evidence-relevant image tokens. The full CNN+MLP scorer achieves the best or tied-best performance across both keep ratios, indicating that token-level visual features and local context features are complementary.

C.3 Scorer Size

We further vary the scorer size while keeping the architecture fixed. Table 10 reports the parameter count, scoring latency, additional VRAM, and accuracy at 20% total prompt-KV retention on LLaVA-1.5-7B, where Base is the default scorer used in all main experiments. Across the three sizes, performance varies by at most 0.13, 0.20, and 0.28 points on DocVQA, TextVQA, and ChartQA, respectively. The Small scorer uses 52.3% fewer parameters and 62.5% less additional VRAM than Base with comparable accuracy, whereas Large nearly doubles the parameter count of Base without improvement. Performance thus saturates at the Small/Base scale, allowing the scorer size to be chosen according to the deployment budget.

C.4 Additional MileBench Category Results

Table 11 reports additional MileBench category-level results on LLaVA-OneVision-7B at 10% image retain. Each score is computed as the mean over datasets with valid evaluation outputs within the corresponding category. The S-5 category is omitted because the `nuscenes` subset requires dataset-specific preprocessing that is incompatible with our current evaluation pipeline. Q-ViK outperforms VisionZip and VFlowOpt on most MileBench categories, including temporal reasoning, object/scene understanding, semantic QA, text-rich image QA, and visual relation generation. The main exception is NH, where Q-ViK underperforms the baselines, suggesting that needle-in-a-haystack retrieval under extreme distractors may require a different retention strategy. Note that the NH score here is ROUGE-L averaged over TextNeedleInAHaystack and ImageNeedleInAHaystack, and is therefore not directly comparable to the TextNeedleInAHaystack accuracy reported in Table 12.

Table 10: Scorer size–performance trade-off at 20% total prompt-KV retention on a single NVIDIA RTX 3090. Scores come from a separate controlled run and may differ slightly from Table 1.

Size	Params (M)	Latency (ms)	Added VRAM (MiB)	DocVQA	TextVQA	ChartQA
Small	**74.43**	**23.15**	**168.7**	26.81	**45.76**	17.80
Base	156.18	24.72	449.8	**26.86**	45.59	**17.84**
Large	309.17	26.33	622.3	26.73	45.56	17.56

Table 11: Additional MileBench category-level results on LLaVA-OneVision-7B at 10% image retain. Each score is the mean over datasets with valid outputs in the corresponding category; S-3 corresponds to the average of the three subsets in Tab. 4. S-5 is omitted due to dataset-specific preprocessing requirements.

Task	Included datasets	VisionZip	VFlowOpt	Ours
T-1	ActionLocalization, ActionPrediction, ActionSequence	39.2	37.2	**58.7**
T-2	ObjectExistence, ObjectInteraction, MovingAttribute, ObjectShuffle	46.1	45.1	**57.1**
T-3	EgocentricNavigation, MovingDirection	**32.0**	28.5	31.8
T-4	CounterfactualInference, StateChange, CharacterOrder, SceneTransition	51.9	52.3	**62.6**
S-1	WebQA, TQA, MultiModalQA, WikiVQA	75.4	73.0	**78.4**
S-2	SlideVQA, OCR-VQA, DocVQA	68.8	71.3	**84.3**
S-3	CLEVR-Change, IEdit, Spot-the-Diff	23.0	22.6	**26.5**
S-4	MMCoQA, ALFRED	47.7	48.0	**51.5**
S-5	nuscenes	–	–	–
NH	TextNeedleInAHaystack, ImageNeedleInAHaystack	30.5	**30.8**	24.8
IR	GPR1200	10.0	10.5	**38.3**

D Additional Analyses

D.1 Additional Future-Utility Mismatch Analysis

Setup for Fig. 3. The alignment analysis in Sec. 4.3 uses 100 samples each from ChartQA, DocVQA, and TextVQA (300 per backbone). The keep ratio is defined over total prompt tokens for both backbones, and all methods retain the same number K of image tokens in each layer. Jaccard@K is computed per layer between each method's retained image tokens and the teacher's top-K set, and is averaged over layers and samples. Full-response preservation is the exact match, after normalization, between the answer under eviction and the full-cache answer. Prefill saliency ranks image tokens by H2O-style accumulated prefill attention [37], averaged over heads.

Qualitative examples. Figure 5 provides additional examples from TextVQA, DocVQA, and GQA, which require reading the zip code of Google Inc., localizing page-number evidence in a document, and identifying whether the street surface is cobblestone, respectively. Across these cases, the teacher focuses on localized answer-critical regions that are not fully captured by prefill saliency, and Q-ViK more closely follows the teacher by retaining image tokens near the evidence used for answer generation. These examples support our claim that future answer-conditioned utility provides a more reliable eviction signal than prompt-time saliency alone.

D.2 Why Visual-only Eviction?

We verify the text-preserving design by comparing image-only eviction with indiscriminate prompt-KV eviction. As shown in Fig. 6, image-only eviction maintains stable ROUGE-L and PPL across keep ratios on DocVQA, whereas evicting text and image KVs together causes sharp degradation. We observe similar trends on other benchmarks: joint text-visual eviction consistently degrades more rapidly than image-only eviction, justifying Q-ViK's restriction of eviction to image KVs.

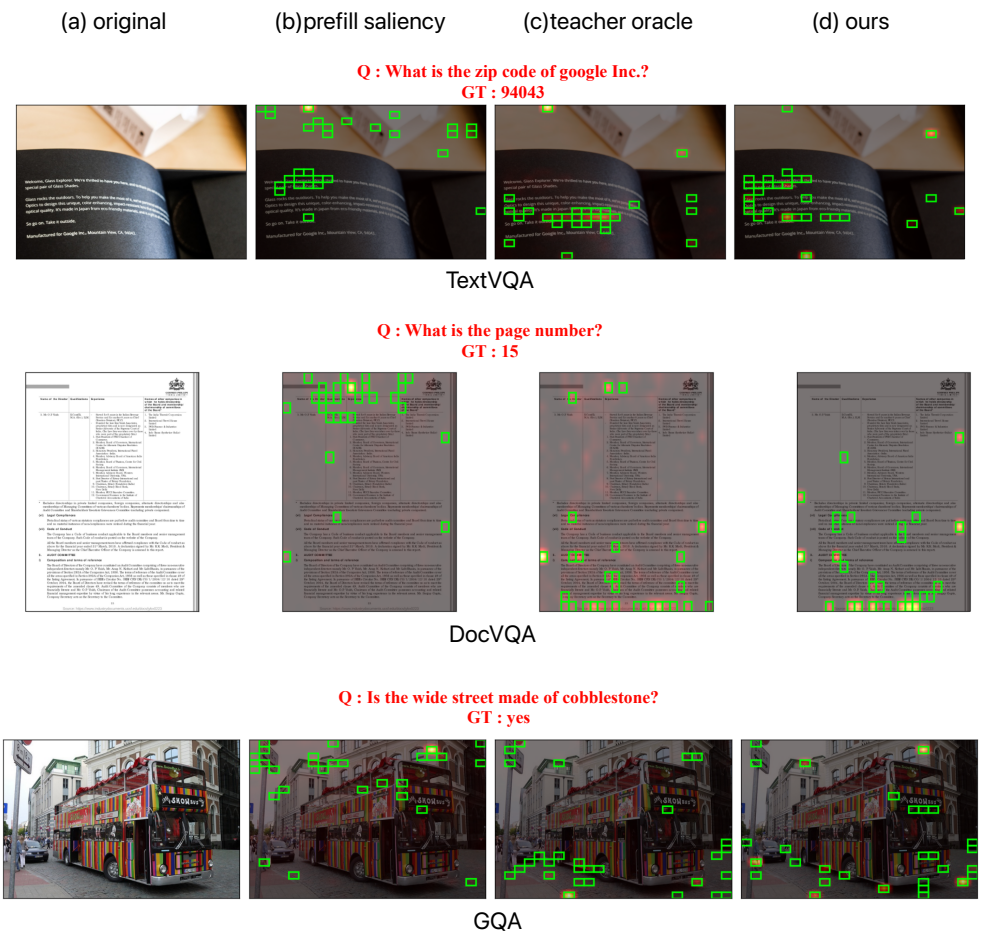

Figure 5: **Additional qualitative examples of future-utility mismatch.** Each row shows an example from TextVQA, DocVQA, and GQA, respectively. Columns correspond to (a) the original image, (b) prefill-saliency-based retained tokens, (c) the full-cache teacher, and (d) Q-ViK. Green boxes denote retained or high-utility image tokens under the same visual budget. Prefill saliency often highlights broad prompt-relevant regions, whereas the teacher concentrates on localized evidence actually reused during answer generation. Q-ViK better follows the teacher, retaining answer-relevant visual tokens before decoding.

Restricting eviction to image KVs does not preclude further text-side compression. Text-KV eviction methods such as StreamingLLM [32] and SnapKV [15] target text KVs, so they can be applied after Q-ViK without structural modification (Appendix A.2); we evaluate this combination in Appendix D.3.

D.3 Sequential Visual–Text Eviction

Tab. 12 reports the accuracy–efficiency trade-off when StreamingLLM is applied to text KVs after Q-ViK. With all text KVs preserved, Q-ViK at 10% image retain matches or exceeds Full Cache on both benchmarks while reducing decoding latency by 26% and 22% and prompt-KV memory by 63% and 44% on MM-NIAH and TextNeedleInAHaystack, respectively. Adding StreamingLLM further reduces prompt-KV memory, but accuracy drops sharply as the text keep ratio decreases (e.g., from 99.4 to 74.4 and 47.5 on TextNeedleInAHaystack at 80% and 50% text keep), while the additional latency savings are small. Text preservation is therefore the accuracy-oriented default, and sequential text-KV eviction is an optional, task-dependent trade-off.

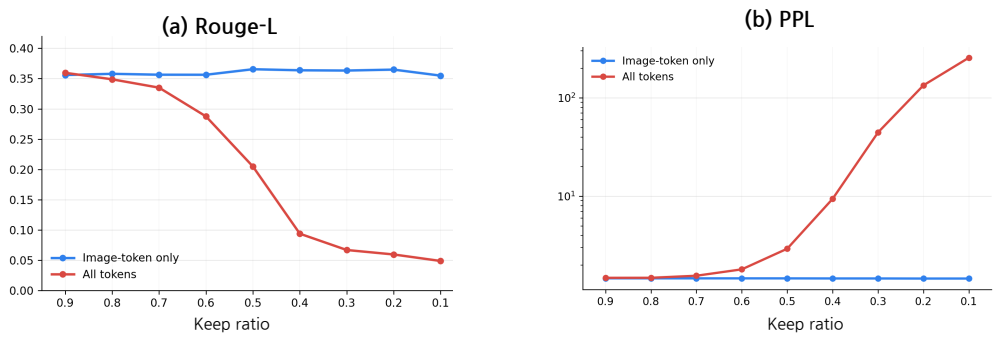

Figure 6: **Image-only vs. all-token eviction on DocVQA.** ROUGE-L and PPL as a function of keep ratio under image-only and indiscriminate prompt-KV eviction. Image-only eviction preserves the textual instruction/question context and degrades more gracefully than evicting image and text KVs together.

Table 12: Accuracy–efficiency trade-off of sequentially applying Q-ViK and StreamingLLM on LLaVA-OneVision-7B, measured on a single NVIDIA RTX 4090. Image and text keep ratios are defined separately over image and text KVs.

Method	Image keep	Text keep	Score ↑ (%)	Latency ↓ (ms/tok)	Peak mem. ↓ (GiB)	Prompt KV ↓ (GiB)
MM-NIAH						
Full Cache	100%	100%	69.2	31.7	16.572	0.611
Q-ViK	10%	100%	70.1	23.5	15.712	0.226
Q-ViK + StreamingLLM	10%	80%	64.2	23.0	15.734	0.189
Q-ViK + StreamingLLM	10%	50%	56.5	21.9	15.666	0.134
MileBench TextNeedleInAHaystack						
Full Cache	100%	100%	99.4	34.1	15.963	0.801
Q-ViK	10%	100%	99.4	26.7	15.756	0.449
Q-ViK + StreamingLLM	10%	80%	74.4	25.4	15.721	0.367
Q-ViK + StreamingLLM	10%	50%	47.5	23.2	15.567	0.244

D.4 Removing All Visual KVs after Prefill

To verify that visual KVs remain necessary during decoding, we remove all visual KVs after prefill on LLaVA-1.5-7B. As shown in Tab. 13, this lowers the average from 45.2 to 38.4, with large drops on text-rich benchmarks such as TextVQA and DocVQA. In contrast, Q-ViK retains 20% of the prompt KVs and reaches 44.9, close to Full Cache, indicating that decoding still relies on visual KVs and that Q-ViK preserves the ones it needs.

Table 13: Removing all visual KVs after prefill on LLaVA-1.5-7B. Full Cache and Q-ViK results are from Tab. 1.

Method	TextVQA	GQA	DocVQA	ChartQA	COCO	NoCaps	TextCaps	Avg.
Full Cache	45.9	62.0	28.1	17.9	55.5	59.4	47.4	45.2
Drop-all visual	31.8	61.7	18.3	15.1	50.9	52.4	38.9	38.4
Q-ViK (20%)	45.8	61.9	26.5	18.0	56.0	58.7	47.1	44.9

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The claims in the abstract and introduction accurately reflect the proposed Q-ViK method and the experimental scope, including student-seen, held-out, multi-image, video, and efficiency evaluations.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper includes a limitations section discussing text-preservation constraints and the offline cost of teacher-signal collection.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: The paper does not present formal theoretical results or proofs.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The paper describes the model backbones, datasets, teacher-signal construction, student training objective, retain-ratio protocols, baselines, and evaluation metrics needed to reproduce the main results.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The Q-ViK implementation and experimental configurations are publicly available through the code link in the abstract. The experiments use publicly available datasets and model checkpoints.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [Yes]

Justification: The paper describes the training datasets, evaluation benchmarks, model backbones, teacher-score extraction, scorer loss, retain-ratio conventions, and baseline comparison protocols. Detailed training hyperparameters and implementation settings, including loss weights, optimizer settings, and eviction layers, are provided in the publicly available code.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: We do not report error bars or statistical significance tests due to the computational cost of repeated LVLM evaluations across multiple datasets and baselines.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Appendix B.3 reports the GPU used for each experiment (RTX 3090 for the adaptation cost and scorer-size study, RTX 4090 for sequential eviction, and RTX A6000 for the latency and memory comparison), and Appendix B.2 reports the one-time adaptation cost (3.2 and 9.9 GPU-hours).

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research uses publicly available models and datasets and conforms to the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

Justification: As Q-ViK improves LVLM inference efficiency, it may lower the deployment barrier for LVLMs, which inherits the broader societal risks of the underlying models.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: The paper does not release a new high-risk foundation model, image generator, scraped dataset, or user-facing system requiring additional safeguards.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: The paper cites the existing models, datasets, and baselines used in the experiments.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.

- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The Q-ViK implementation is publicly available through the code link in the abstract. The paper documents the method, and the released code provides implementation details and experimental configurations.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: This work does not involve crowdsourcing experiments or research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

Justification: This work does not involve human-subject studies and therefore does not require IRB approval.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: The core method does not use external LLMs as an important, original, or non-standard component. Any LLM use for writing, editing, or formatting does not affect the methodology, scientific rigor, or originality of the research.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.