GATED ATTENTION-AWARE LAYOUT-PRESERVING PHYSICAL KV CACHE COMPRESSION IN VISION-LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Large vision-language models cache visual representations in decoder key-value (KV) states throughout autoregressive generation, creating a substantial memory burden. However, existing KV compression methods mainly target language-style, one-dimensional sequences and overlook the two-dimensional spatial organization of visual states, so aggressive compression can discard the spatial evidence the decoder still relies on. To address this issue, we present Gated Attention-aware Layout-Preserving KV compression (GALP-KV), a training-free method that physically compresses the decoder KV cache after multimodal prefill while preserving model-aligned visual layout. GALP-KV identifies the visual span in the full decoder KV cache, maps visual entries to the model-aligned grid, protects all non-visual states, and scores entries by target-model attention utility augmented with a weak geometric prior. Its primary causal mechanism is a *budget-gated, layer- and KV-head-specific selection policy*: a shared layout-preserving stable route at moderate budgets, and a per-head proxy route under aggressive compression. Experimental results show that the geometric prior contributes no measurable marginal effect once uniform and tail anchors are present. On Qwen2.5-VL, GALP-KV reduces physical KV memory by 48.76% while preserving near-Full-KV question-answering (QA) behavior, and remains stable at a 56.23% reduction, where the layout-aware mIoU advantage over attention-only selection widens. On LLaVA-1.5-7B, the same policy achieves a 52.60% reduction and changes only 224 predictions relative to Full KV, fewer than uniform, random, and SnapKV retention. Matched-count analysis shows that GALP-KV leaves the smallest largest-empty-radius (spatial blind region) of all compared selectors at the main budget under a paired bootstrap, forming a top tier with NACL across budgets. Grounding remains more sensitive than QA, and the current prototype does not yet provide robust end-to-end latency acceleration.

1 INTRODUCTION

Vision-language models (VLMs) such as LLaVA and Qwen2.5-VL encode visual inputs as tokens, insert them into an autoregressive decoder, and generate responses from visual and textual context Liu et al. (2023); Bai et al. (2025). This architecture supports diverse tasks but inherits a major large language model (LLM) inference bottleneck: the decoder key-value (KV) cache grows with the prefill length and persists throughout autoregressive generation. The bottleneck is especially acute for VLMs, because visual inputs produce many cached states tied to image regions, objects, layouts, and spatial relations; for example, in remote-sensing imagery, a correct prediction may depend on a small vehicle, a runway boundary, a building cluster, or relations among multiple objects. Visual KV compression therefore requires determining which entries can be removed without discarding the spatial evidence required by the decoder.

Efficient VLM methods mainly reduce visual-token overhead before or during multimodal encoding through pruning, merging, packing, or selective retention Chen et al. (2024a); Zhang et al. (2025); Yang et al. (2025b). Although they reduce prefill computation and input length, they do not directly compress physical KV states already created after multimodal prefill; input-side visual-token reduction and post-prefill KV compression are therefore complementary.

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.

Language-only KV compression retains states using heavy-hitter attention, recency, attention sinks, or layer-wise allocation Zhang et al. (2023); Li et al. (2024b); Chen et al. (2024b). Such one-dimensional heuristics may overlook the two-dimensional organization of visual cache entries, and attention-based top-k selection can over-concentrate retained states in a salient region while leaving much of the visual layout uncovered. Recent multimodal methods further consider modality-aware merging, allocation, or inter-modal relevance Liu et al. (2024c); Huang et al. (2025); however, visual KV retention is still commonly formulated as token importance, without explicitly preserving spatial organization under a fixed physical budget.

To this end, we propose GALP-KV, a training-free method for gated attention-aware, layout-preserving physical KV compression after multimodal prefill. It directly compresses decoder KV states, identifies the visual span, aligns entries with the model grid, protects non-visual prompt states, scores entries by target-model attention utility with a weak geometric regularizer, and applies a *budget-gated selection policy*: a shared layout-preserving stable route for captioning and moderate compression, and a layer- and KV-head-specific proxy route under aggressive compression. Component ablations at two budgets show that the gated per-head selection—not the geometric regularizer—is the primary causal driver of downstream referring gains.

We evaluate GALP-KV on Qwen2.5-VL and transfer the same policy to LLaVA-1.5-7B without retuning. VRSBench jointly tests QA, captioning, and referring for semantic generation and spatial grounding Li et al. (2024a). GALP-KV reduces KV length and byte size while preserving QA and caption behavior, whereas grounding remains more sensitive to visual-state removal.

Our main contributions are summarized as follows:

- We formulate physical visual KV compression as post-prefill selection over the actual multimodal decoder KV cache, distinguishing it from input-side visual-token reduction.
- We introduce a training-free policy that aligns visual cache positions with the model-specific grid, scores them with target-model utility plus a weak geometric regularizer, and —most importantly—routes selection between a shared stable layout-preserving set and a per-head proxy set through a budget-gated policy. Ablation evidence attributes the downstream referring gains to this gating mechanism rather than to the geometric prior in isolation.
- We conduct multi-task evaluations on Qwen2.5-VL and LLaVA-1.5-7B, demonstrating substantial physical KV memory reduction with limited generation changes while identifying grounding sensitivity and unoptimized runtime as explicit system limitations.

2 Related Work

KV cache compression has been studied extensively for language-only inference. PagedAttention organizes cache into non-contiguous blocks to reduce fragmentation Kwon et al. (2023); eviction methods retain states by heavy-hitter attention (H2O Zhang et al. (2023)), attention sinks and recency (StreamingLLM Xiao et al. (2024)), observation-window importance (SnapKV Li et al. (2024b)), randomization (NACL Chen et al. (2024b)), or layer-wise budgets (PyramidKV Cai et al. (2024)); KIVI and KVQuant instead lower cache precision through quantization Liu et al. (2024b); Hooper et al. (2024). These principles are effective for one-dimensional text sequences, but they treat cached positions as a linear stream and do not model the two-dimensional spatial structure of visual entries.

Efficient VLM inference has mostly reduced visual computation before or during multimodal prefill, by pruning, merging, packing, or selectively retaining visual tokens Shang et al. (2025); Chen et al. (2024a); Zhang et al. (2025); Yang et al. (2025b); Li et al. (2025); Vasu et al. (2025); Yang et al. (2025a). This input-side direction lowers visual encoding and prefill cost, whereas GALP-KV operates after prefill and physically shortens the decoder `past_key_values`; the two are complementary rather than interchangeable.

More closely related work compresses multimodal KV caches directly. Elastic Cache merges low-importance entries into anchor states Liu et al. (2024c), LOOK-M uses text-prior guidance for long-context compression Wan et al. (2024), VL-Cache allocates capacity by sparsity and modality Tu et al. (2025), and AirCache models inter-modal relevance for selection Huang et al. (2025). These studies show that multimodal caches need policies beyond language-only eviction, yet visual re-

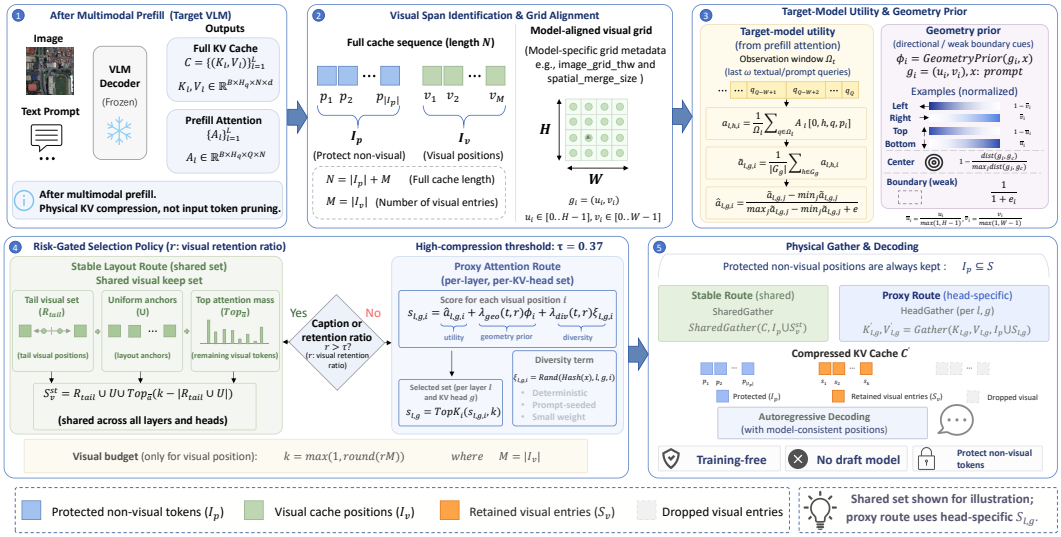


Figure 1: Overview of GALP-KV. After multimodal prefill, GALP-KV aligns visual cache entries with the model grid, performs budget-gated selection, physically gathers retained states, and continues position-consistent decoding.

tention is still framed mainly as token importance. GALP-KV does not claim that multimodal KV compression is new; its focus is the spatial organization of the visual cache: it identifies the true visual span, aligns entries with the model-specific grid, protects non-visual context, and preserves a structured subset of visual KV states under a fixed physical budget.

3 METHOD

GALP-KV is designed for VLMs in which visual representations are inserted into an autoregressive decoder and stored in the decoder KV cache. As illustrated in Fig. 1, GALP-KV operates after multimodal prefill and directly removes selected visual entries from the physical decoder KV cache. The method contains four main stages: visual-span identification and grid alignment, target-model utility estimation, budget-gated visual-state selection, and physical cache gathering followed by position-consistent decoding. All model parameters remain frozen throughout the procedure.

3.1 PROBLEM FORMULATION

Given an image–text prompt x, a VLM performs multimodal prefill and produces a decoder KV cache:

$$C = \{(K_l, V_l)\}_{l=1}^{L}, \qquad K_l, V_l \in R^{B\times H_{\text{kv}}\times N\times d}, \tag{1}$$

where L is the number of decoder layers, H_{kv} is the number of KV heads, N is the prefill cache length, and d is the head dimension.

Let

$$I_v = \{p_1, p_2, \ldots, p_M\}, \qquad M = |I_v|, \tag{2}$$

denote the cache positions associated with visual entries. The remaining cache positions are protected:

$$I_p = \{0, 1, \ldots, N-1\} \setminus I_v. \tag{3}$$

These protected positions include textual tokens, multimodal special tokens, and other non-visual context. GALP-KV removes entries only from I_v and always retains all positions in I_p.

Given a visual retention ratio $r \in (0, 1]$, the visual retention budget is:

$$k = \max\left(1, \text{round}(rM)\right). \tag{4}$$

Let $S_v \subseteq I_v$ denote the retained visual set, where $|S_v| = k$. The complete retained position set is

$$S = \mathrm{Sort}\left(I_p \cup S_v\right), \tag{5}$$

where sorting preserves the original cache order. The physical sequence reduction and byte-level KV reduction are measured as

$$\rho_{seq} = 1 - \frac{|S|}{N}, \qquad \rho_{kv} = 1 - \frac{\mathrm{bytes}\left(C'\right)}{\mathrm{bytes}\left(C\right)}, \tag{6}$$

where C' denotes the compressed physical cache.

Conceptually, the objective is to select a visual subset that preserves the generation behavior of the frozen VLM under a fixed cache budget:

$$\begin{aligned} &\max_{S_v} \mathrm{Quality}\left(f_\theta(x; C')\right) \\ &\text{s.t.} \quad |S_v| = k,\ S_v \subseteq I_v,\ I_p \subseteq S. \end{aligned} \tag{7}$$

This expression defines the selection goal and is not optimized through gradient descent. GALP-KV introduces no trainable parameters or auxiliary draft model.

3.2 Model-Aligned Visual Cache Mapping

Visual cache entries do not necessarily form a fixed grid at the tokenizer level. GALP-KV therefore recovers their spatial organization from the metadata and token layout of the target model. For Qwen2.5-VL, the mapping is derived from model-exposed metadata, including `image_grid_thw` and the spatial merge configuration Bai et al. (2025). After identifying the corresponding decoder positions, each visual cache entry $p_i \in I_v$ is assigned a coordinate

$$g_i = (u_i, v_i), \quad u_i \in \{0, \ldots, H-1\},\ v_i \in \{0, \ldots, W-1\}, \tag{8}$$

where (H, W) is the actual visual grid after model-specific spatial merging.

This model-aligned mapping is different from imposing a fixed proxy grid. It preserves the correspondence between each decoder cache entry and the visual region represented by the target model. For models that do not directly expose grid metadata, the mapping can be reconstructed from the visual-projector output layout, provided that the correspondence between projector tokens and decoder positions is available.

3.3 Target-Model Attention Utility

GALP-KV estimates the utility of each visual cache entry from the prefill attention of the deployed VLM. Let

$$A_l \in R^{B \times H_q \times Q \times N} \tag{9}$$

denote the attention tensor at decoder layer l, where H_q is the number of query heads and Q is the number of prefill queries. Following the observation-window principle used in attention-based cache selection Li et al. (2024b), GALP-KV uses the last w textual or prompt queries rather than averaging over visual query positions. Let Ω_l denote this observation window. For visual position p_i, the query-head utility is

$$a_{l,h,i} = \frac{1}{|\Omega_l|} \sum_{q \in \Omega_l} A_l[0, h, q, p_i]. \tag{10}$$

For models using grouped-query attention, several query heads share one KV head Ainslie et al. (2023). Let G_g be the set of query heads associated with KV head g. The utility is reduced to the KV-head level as

$$\widetilde{a}_{l,g,i} = \frac{1}{|G_g|} \sum_{h \in G_g} a_{l,h,i}. \tag{11}$$

To avoid scale differences across layers and heads, the scores are normalized independently:

$$\widehat{a}_{l,g,i} = \frac{\widetilde{a}_{l,g,i} - \min_j \widetilde{a}_{l,g,j}}{\max_j \widetilde{a}_{l,g,j} - \min_j \widetilde{a}_{l,g,j} + \epsilon}. \tag{12}$$

The resulting score is grounded in the target model and does not require an external saliency network.

For the shared stable route introduced below, we additionally compute a global visual utility by averaging over decoder layers and KV heads:

$$\overline{a}_i = \frac{1}{LH_{kv}} \sum_{l=1}^{L} \sum_{g=1}^{H_{kv}} \widehat{a}_{l,g,i}. \tag{13}$$

3.4 Geometry-Aware Visual Priors

Pure attention-based top-k selection may concentrate retained entries in a small salient region; GALP-KV adds three weak geometric cues to counter this collapse. For grid coordinate $g_i = (u_i, v_i)$, normalized positions $\bar{u}_i = u_i / \max(1, H-1)$ and $\bar{v}_i = v_i / \max(1, W-1)$ define directional cues

$$d_i^{\text{left}} = 1-\bar{v}_i,\ d_i^{\text{right}} = \bar{v}_i,\ d_i^{\text{top}} = 1-\bar{u}_i,\ d_i^{\text{bottom}} = \bar{u}_i, \tag{14}$$

a center cue $d_i^{\text{center}} = 1 - \text{dist}(g_i, g_c) / \max_j \text{dist}(g_j, g_c)$ (Euclidean distance in normalized coordinates; denominator 1 when all entries share one grid cell; directions refer to the rendered image frame, not geodetic bearings), and a boundary cue $b_i = 1/(1+e_i)$ where e_i is the distance from g_i to the grid boundary. The final geometric term combines whichever cues the prompt reliably triggers, min–max normalized via $\text{Norm}(\cdot)$:

$$\phi_i = \begin{cases} \text{Norm}(d_i), & \text{directional evidence only,} \\ \text{Norm}(b_i), & \text{boundary cue only,} \\ \text{Norm}(0.75\, d_i + 0.25\, b_i), & \text{both active,} \\ 0, & \text{otherwise.} \end{cases} \tag{15}$$

The contribution is kept weak, acting as a stabilizer rather than a replacement for target-model utility.

3.5 Budget-Gated Visual-State Selection

Different tasks exhibit different failure modes under cache compression. Captioning generally benefits from broad scene coverage, whereas aggressive compression for QA and referring requires more flexible layer- and head-specific selection. GALP-KV therefore uses a budget-gated policy:

$$\pi(x, r) = \begin{cases} \mathit{Stable}, & \text{Caption}(x) = 1 \text{ or } r > \tau, \\ \mathit{Proxy}, & \text{otherwise,} \end{cases} \tag{16}$$

where $\tau = 0.37$ separates moderate from aggressive compression and is fixed before full-validation evaluation; a sensitivity sweep over $\tau \in \{0.30, 0.37, 0.45\}$ (Appendix B.3) shows the routing decision is insensitive across this range. In the current implementation, $\text{Caption}(x)$ is inferred from the fixed prompt template and does not use the target answer or ground-truth annotation.

Stable layout route. The stable route produces one visual keep set shared by all decoder layers and KV heads. It combines a small tail set R_{tail}, an approximately uniform anchor set U, and the remaining high-utility visual positions. The tail set refers to the final entries in the rasterized visual-token sequence and does not represent temporal recency.

Let

$$B = R_{tail} \cup U \tag{17}$$

denote the initial anchor set. To satisfy the exact visual budget, we define

$$\widetilde{B} = \begin{cases} B, & |B| \le k, \\ \text{Top}_{\overline{a}}(B, k), & |B| > k, \end{cases} \tag{18}$$

where $\text{Top}_{\overline{a}}(X, m)$ selects the m positions with the highest global utility from candidate set X. The shared visual keep set is then

$$S_v^{st} = \widetilde{B} \cup \text{Top}_{\overline{a}}\left(I_v \setminus \widetilde{B}, k - \left|\widetilde{B}\right|\right). \tag{19}$$

For prompts with explicit spatial evidence, a guarded replacement operation can introduce a small directional anchor set A_{geo}:

$$S_v^{st} \leftarrow \mathrm{GuardedReplace}\left(S_v^{st}, A_{geo}\right). \tag{20}$$

The operation inserts a geometric anchor only when the corresponding prompt cue is reliable and removes only a low-utility position. The set size therefore remains equal to k.

Proxy attention route. Under high compression, the proxy route uses

$$\begin{aligned} \xi_{l,g,i} &= \mathrm{Rand}\left(\mathrm{Hash}(x), l, g, i\right), \\ s_{l,g,i} &= \widehat{a}_{l,g,i} + \lambda_{\mathrm{geo}}(t,r)\phi_i + \lambda_{\mathrm{div}}(t,r)\xi_{l,g,i}, \\ S_{l,g} &= \mathrm{TopK}_i\left(s_{l,g,i}, k\right). \end{aligned} \tag{21}$$

The prompt-seeded diversity term has a small weight and resolves brittle ties without uncontrolled stochasticity.

3.6 Physical Cache Gathering and Position-Consistent Decoding

For the stable route, the complete ordered keep set is

$$S^{st} = \mathrm{Sort}\left(I_p \cup S_v^{st}\right). \tag{22}$$

The physical cache tensors are gathered as

$$K_l' = K_l[:,:,S^{st},:], \qquad V_l' = V_l[:,:,S^{st},:]. \tag{23}$$

For the proxy route, each layer and KV head uses

$$S_{l,g}^{all} = \mathrm{Sort}\left(I_p \cup S_{l,g}\right), \tag{24}$$

followed by

$$K_{l,g}' = K_l[:,g,S_{l,g}^{all},:], \qquad V_{l,g}' = V_l[:,g,S_{l,g}^{all},:]. \tag{25}$$

The same indices are applied to keys and values. Dropped entries are physically removed rather than logically masked.

After gathering, the compact cache is converted back into a decoder-compatible cache object. The physical cache index and the logical model position are handled separately: retained entries preserve their original order, while decode-time position identifiers continue from the original multimodal prefill positions. This prevents cache slicing from incorrectly resetting rotary or multimodal positional phases Su et al. (2024); Bai et al. (2025). All non-visual positions remain protected throughout decoding.

3.7 Complexity and Implementation Scope

GALP-KV computes the retention policy once after prefill. Given an observation window of length w and M visual entries, the time and KV-memory complexities are

$$\begin{aligned} T_{\mathrm{agg}} &= O(LH_q wM), \qquad T_{\mathrm{sel}} = O(LH_{\mathrm{kv}} M \log k), \\ T_{\mathrm{gather}} &= O(LH_{\mathrm{kv}}(|I_p|+k)d), \\ M_{\mathrm{KV}} &: \; O(LH_{\mathrm{kv}}Nd) \rightarrow O(LH_{\mathrm{kv}}(|I_p|+k)d). \end{aligned} \tag{26}$$

The current prototype exposes prefill attention tensors and gathers the cache through a custom decoding path, so we claim physical KV reduction rather than optimized end-to-end latency. Because the resident KV is smaller, GALP-KV decodes strictly faster per token than Full KV; the selection overhead amortizes within a low-hundreds-token crossover, earlier than every compressed baseline measured (Appendix E.2).

Table 1: Full-validation QA and referring results on Qwen2.5-VL under two visual retention budgets. Reg.QA items transitioning from Full-KV-correct to compressed-incorrect (relaxed metric, 37,409 questions). Best and second-best among compressed methods shown in bold and underlined, respectively. Compared baselines: NACL Chen et al. (2024b), SnapKV Li et al. (2024b), H2O Zhang et al. (2023), ElasticCache Liu et al. (2024c).

		QA		Referring			
Method	KV Red. (%) ↑	Exact ↑	Relaxed ↑	Acc. ↑	mIoU ↑	ΔmIoU ↑	Reg. ↓
Panel A: Visual retention ratio $r = 0.435$							
Full KV	0.00	0.4712	0.5293	0.4172	0.3907	+0.0000	0
NACL	48.76	0.4711	0.5281	**0.4170**	**0.3890**	**-0.0016**	**370**
SnapKV	48.76	0.4708	0.5281	0.4135	0.3846	-0.0061	574
H2O	48.76	0.4711	0.5285	0.3493	0.3309	-0.0597	1,744
ElasticCache	48.76	0.4669	0.5281	0.2633	0.2615	-0.1292	3,091
GALP-KV	48.76	**0.4712**	**0.5286**	0.4118	0.3848	-0.0058	503
Panel B: Visual retention ratio $r = 0.35$							
Full KV	0.00	0.4712	0.5293	0.4172	0.3907	+0.0000	0
NACL	56.23	**0.4713**	0.5283	**0.4175**	**0.3882**	**-0.0024**	**418**
SnapKV	56.23	0.4708	0.5281	0.4086	0.3807	-0.0100	744
H2O	56.23	0.4710	**0.5284**	0.3225	0.3102	-0.0804	2,196
ElasticCache	56.23	0.4661	0.5280	0.2215	0.2360	-0.1547	3,638
GALP-KV	56.23	**0.4713**	0.5283	0.4154	0.3873	-0.0033	449

Table 2: Full-validation captioning results on Qwen2.5-VL under the same two visual retention budgets as Table 1. Captioning metrics are absolute scores; best and second-best among compressed methods are shown in bold and underlined, respectively.

Method	KV Red. (%) ↑	CIDEr ↑	ROUGE-L ↑	BLEU-4 ↑
Panel A: Visual retention ratio $r = 0.435$				
Full KV	0.00	0.0476	0.2398	0.0386
NACL	48.76	0.0472	0.2413	0.0388
SnapKV	48.76	0.0463	0.2385	0.0365
H2O	48.76	0.0467	0.2394	0.0375
ElasticCache	48.76	**0.0984**	**0.2415**	**0.0419**
GALP-KV	48.76	0.0442	0.2374	0.0361
Panel B: Visual retention ratio $r = 0.35$				
Full KV	0.00	0.0476	0.2398	0.0386
NACL	56.23	0.0478	**0.2408**	0.0385
SnapKV	56.23	0.0458	0.2374	0.0355
H2O	56.23	0.0462	0.2390	0.0371
ElasticCache	56.23	**0.0973**	0.2397	**0.0406**
GALP-KV	56.23	0.0442	0.2358	0.0352

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

We use Qwen2.5-VL-3B-Instruct as the primary backbone Bai et al. (2025) and evaluate transfer on LLaVA-1.5-7B Liu et al. (2023; 2024a). All parameters remain frozen. We evaluate on the VRSBench validation split Li et al. (2024a). Experiments run on an NVIDIA RTX 3090 using batch size one, greedy decoding, and seed 42. Visual grids are recovered from model metadata for Qwen2.5-VL and from projector outputs for LLaVA-1.5-7B. All methods preserve identical non-visual cache positions.

We compare GALP-KV with Full KV, uniform and random retention, SnapKV Li et al. (2024b), H2O Zhang et al. (2023), NACL Chen et al. (2024b), and Elastic Cache Liu et al. (2024c). Official KVCache-Factory reproductions additionally include PyramidKV Cai et al. (2024), StreamingLLM

Table 3: Matched-count layout analysis recomputed on *true* retained visual positions (141 visual entries per sample, $r = 0.435$, 600 VRSBench QA image-sessions). Higher quadrant entropy ↑ is more uniform; smaller largest-empty-radius ↓ is a smaller spatial blind region. GALP-KV attains the smallest blind region, while NACL is marginally more uniform in quadrant entropy. Best per column in **bold**. Baselines follow the protocols of SnapKV Li et al. (2024b), H2O Zhang et al. (2023), NACL Chen et al. (2024b), and ElasticCache Liu et al. (2024c).

Selector	Kept visual	Quadrant entropy ↑	Largest empty radius ↓
SnapKV	141	0.969	0.142
H2O	141	0.877	0.190
NACL	141	**0.990**	0.092
ElasticCache	141	0.897	0.179
GALP-KV	141	0.987	**0.090**

Table 4: Cross-backbone transfer on LLaVA-1.5-7B at 52.60% physical KV reduction. The POPE natural-image transfer is reported in Appendix F.3.

Method	QA Relaxed ↑	Δ Relaxed ↑	Changed predictions ↓	KV Red. (%) ↑
Full KV	0.5400	+0.0000	0	0.00
Uniform	0.5397	-0.0002	753	52.60
Random	0.5393	-0.0007	1,012	52.60
SnapKV Li et al. (2024b)	0.5401	+0.0001	410	52.60
GALP-KV	0.5399	-0.0001	224	52.60

Xiao et al. (2024), L2Norm Devoto et al. (2024), and CaM Zhang et al. (2024). We report task performance, cache reduction, prediction changes, layout statistics, latency, and throughput. Physical KV reduction follows Eq. 6.

4.2 Main Quantitative Results

Table 1 reports full-validation QA and referring results (captioning in Table 2) under two matched visual-KV budgets. GALP-KV reduces KV memory by 48.76% and 56.23% at r=0.435 and r=0.35 while preserving Full-KV QA (exact 0.4712/0.4713, relaxed within 0.0007/0.0010). Referring is more sensitive: NACL Chen et al. (2024b) tops mIoU at both budgets with GALP-KV close behind, and the layout-aware GALP-KV overtakes attention-only SnapKV Li et al. (2024b) by a gap that widens from $+0.0002$ at r=0.435 to $+0.0066$ at r=0.35; both stay well ahead of H2O Zhang et al. (2023) and ElasticCache Liu et al. (2024c). Paired bootstrap at r=0.435 shows a QA-relaxed advantage over NACL ($+0.00056$, $[+0.00024, +0.00088]$) and a referring-mIoU disadvantage (-0.00416, $[-0.00600, -0.00236]$): near-Full-KV QA and competitive grounding without uniformly surpassing NACL.

Table 2 shows a task-dependent captioning trade-off: GALP-KV stays close to Full KV but scores lower on CIDEr, ROUGE-L, and BLEU-4, while NACL most closely preserves Full-KV caption behavior. ElasticCache attains the highest CIDEr and BLEU-4 but at the cost of severe referring degradation and a mean output length that falls from 74.5 to 45.3 words, i.e., a change in generation length and style rather than improved understanding. Overall, GALP-KV offers the more balanced trade-off across QA, grounding, and captioning under the same budget.

4.3 Quality–Memory Trade-off

Figure 2 evaluates the quality–memory trade-off over five physical KV reduction levels on fixed diagnostic subsets (exact values in Appendix B.4): QA relaxed accuracy remains nearly unchanged across all budgets, while referring mIoU varies substantially and caption CIDEr deviations grow at the most aggressive budget, indicating that no single selector dominates all tasks and compression levels.

4.4 Mechanism and Layout Analysis

Comparison with Alternative Selectors. Table 3 compares the complete method with alternative selectors under the same retained visual-token count, with every statistic recomputed on the *true* retained positions (coverage-logging correction and exact confidence intervals in Appendix B.5). At r=0.435, GALP-KV attains the smallest largest-empty-radius of all selectors—a blind-region advantage a paired bootstrap confirms is significant against every baseline—together with the second-highest quadrant entropy, marginally below the slightly more uniform NACL, with which it forms a common top tier across budgets. Layout statistics are therefore read as structural diagnostics of cache geometry rather than substitutes for grounding.

Causal Component Contributions. The cumulative component assembly at two budgets (r=0.435 stable route; r=0.35 proxy route), with matched visual entries per sample, is analyzed in Table 8 (Appendix B.5). The uniform and tail anchors progressively improve layout coverage, whereas the isolated +Geometry step contributes no measurable marginal effect at either budget (Ref. mIoU moves by -0.0006 at r=0.35), ruling out geometry as an additive causal contributor. In contrast, the full GALP-KV with budget-gated, layer- and KV-head-specific selection improves Ref. mIoU by $+0.0094$ and $+0.0241$ at the two budgets. Panel B shows a decisive dissociation: the full method loses matched-count layout coverage relative to +Geometry yet gains on downstream grounding, so the benefit comes from the gated per-head selection policy rather than from the layout statistics themselves.

4.5 Broader Evidence: Fidelity, Transfer, and Physical Reduction

Behavioral Fidelity and Cross-Query Reusability. The aggregate accuracy changes little for every selector ($|\Delta$ relaxed $| \leq 0.0013$ at $r = 0.435$); per-sample flips against Full KV and cross-query cache reuse are therefore examined. Over all 37,409 VRSBench QA pairs, GALP-KV is observed to produce the fewest net regressions and damaged images among deterministic selectors (6.75 MiB KV saved per question), and freezing its cache for later questions shifts relaxed QA accuracy by at most 0.006 with an answer-flip rate $\leq 0.6\%$ (the full analysis is provided in Appendix F).

Cross-Backbone Transfer. The cross-backbone transfer without method-specific retuning is evaluated in Table 4. At 52.60% physical KV reduction on LLaVA-1.5-7B, GALP-KV changes only 224 predictions relative to Full KV (SnapKV 410, uniform 753, random 1,012). Although SnapKV is marginally higher in relaxed accuracy, GALP-KV shows lower behavioral drift rather than an accuracy advantage. A POPE natural-image transfer at $r = 0.35$ (Appendix F.3, Table 15) supports cross-domain non-degradation: relative to Full KV, GALP-KV changes accuracy by at most 0.0017 across the three splits at parse rate 1.0, matching SnapKV and NACL.

Physical Resource and Runtime. On the 50-sample long-generation subset, physical gathering reduces the cache from 380 to 197 positions and from 13.36 to 6.93 MiB (48.16% KV reduction); all matched-budget selectors retain the same length, so this validates physical cache removal rather than an algorithm-specific memory advantage. The prototype does not lower the CUDA allocation peak or provide optimized wall-clock acceleration.

5 Conclusion

In this paper, we analyzed the visual key-value cache memory bottleneck in large vision-language models and found that existing compression methods inherit one-dimensional, language-style heuristics that discard the spatial evidence the decoder still relies on under aggressive compression. In this light, we presented GALP-KV, a training-free method that physically compresses the decoder KV cache after multimodal prefill while preserving model-aligned visual layout, built on a budget-gated, layer- and KV-head-specific selection policy and a model-aligned grid mapping with a weak geometric regularizer. Experiments on Qwen2.5-VL with transfer to LLaVA-1.5-7B and POPE confirm that the gated routing, not the geometric prior, drives downstream grounding gains. The overall results demonstrate that our method achieves superior performance while maintaining computational efficiency, though grounding remains task-sensitive and the prototype lacks end-to-end acceleration.

AI USE STATEMENT

We used a generative AI assistant to help draft and check LaTeX text and to review experimental scripts. All scientific claims, results, and code were verified by the authors, who take full responsibility for the final content.

REPRODUCIBILITY STATEMENT

All experiments use frozen, publicly available backbones (Qwen2.5-VL-3B-Instruct and LLaVA-1.5-7B) and the public VRSBench, RSICD, and COCO/POPE datasets. Model versions, decoding settings (batch size one, greedy decoding, seed 42), prompts, budget definitions, and evaluator implementations are described in the main text and Appendix A. Anonymized code and per-sample prediction logs will be released to support full reproduction.

ETHICS STATEMENT

This work uses only public benchmark datasets and pretrained models and involves no human subjects or personally identifiable data. Physical KV-cache compression reduces inference memory; we are not aware of specific ethical risks beyond those common to vision-language models.

REFERENCES

Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebrón, and Sumit Sanghai. GQA: training generalized multi-query transformer models from multi-head checkpoints. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pp. 4895–4901. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.EMNLP-MAIN.298. URL https://doi.org/10.18653/v1/2023.emnlp-main.298.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Ming-Hsuan Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report. *CoRR*, abs/2502.13923, 2025. doi: 10.48550/ARXIV.2502.13923. URL https://doi.org/10.48550/arXiv.2502.13923.

Zefan Cai, Yichi Zhang, Bofei Gao, Yuliang Liu, Tianyu Liu, Keming Lu, Wayne Xiong, Yue Dong, Baobao Chang, Junjie Hu, and Wen Xiao. Pyramidkv: Dynamic KV cache compression based on pyramidal information funneling. *CoRR*, abs/2406.02069, 2024. doi: 10.48550/ARXIV.2406.02069. URL https://doi.org/10.48550/arXiv.2406.02069.

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 Ales Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol (eds.), *Computer Vision - ECCV 2024 - 18th European Conference, Milan, Italy, September 29-October 4, 2024, Proceedings, Part LXXXI*, volume 15139 of *Lecture Notes in Computer Science*, pp. 19–35. Springer, 2024a. doi: 10.1007/978-3-031-73004-7_2. URL https://doi.org/10.1007/978-3-031-73004-7_2.

Yilong Chen, Guoxia Wang, Junyuan Shang, Shiyao Cui, Zhenyu Zhang, Tingwen Liu, Shuohuan Wang, Yu Sun, Dianhai Yu, and Hua Wu. NACL: A general and effective KV cache eviction framework for LLM at inference time. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pp. 7913–7926. Association for Computational Linguistics, 2024b. doi: 10.18653/V1/2024.ACL-LONG.428. URL https://doi.org/10.18653/v1/2024.acl-long.428.

Alessio Devoto, Yu Zhao, Simone Scardapane, and Pasquale Minervini. A simple and effective l_2 norm-based strategy for KV cache compression. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pp. 18476–18499. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.EMNLP-MAIN.1027. URL https://doi.org/10.18653/v1/2024.emnlp-main.1027.

Coleman Hooper, Sehoon Kim, Hiva Mohammadzadeh, Michael W. Mahoney, Yakun Sophia Shao, Kurt Keutzer, and Amir Gholami. Kvquant: Towards 10 million context length LLM inference with KV cache quantization. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), *Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/028fcbcf85435d39a40c4d61b42c99a4-Abstract-Conference.html.

Kai Huang, Hao Zou, Bochen Wang, Ye Xi, Zhen Xie, and Hao Wang. Aircache: Activating inter-modal relevancy KV cache compression for efficient large vision-language model inference. In *IEEE/CVF International Conference on Computer Vision, ICCV 2025, Honolulu, HI, USA, October 19-25, 2025*, pp. 23958–23967. IEEE, 2025. doi: 10.1109/ICCV51701.2025.02221. URL https://doi.org/10.1109/ICCV51701.2025.02221.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In Jason Flinn, Margo I. Seltzer, Peter Druschel, Antoine Kaufmann, and Jonathan Mace (eds.), *Proceedings of the 29th Symposium on Operating Systems Principles, SOSP 2023, Koblenz, Germany, October 23-26, 2023*, pp. 611–626. ACM, 2023. doi: 10.1145/3600006.3613165. URL https://doi.org/10.1145/3600006.3613165.

Wentong Li, Yuqian Yuan, Jian Liu, Dongqi Tang, Song Wang, Jie Qin, Jianke Zhu, and Lei Zhang. Tokenpacker: Efficient visual projector for multimodal LLM. *Int. J. Comput. Vis.*, 133(10): 6794–6812, 2025. doi: 10.1007/S11263-025-02491-7. URL https://doi.org/10.1007/s11263-025-02491-7.

Xiang Li, Jian Ding, and Mohamed Elhoseiny. Vrsbench: A versatile vision-language benchmark dataset for remote sensing image understanding. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), *Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024a. URL http://papers.nips.cc/paper_files/paper/2024/hash/05b7f821234f66b78f99e7803fffa78a-Abstract-Datasets_and_Benchmarks_Track.html.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. In *The 2023 Conference on Empirical Methods in Natural Language Processing*, 2023. URL https://openreview.net/forum?id=xozJw0kZXF.

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. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), *Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024b. URL http://papers.nips.cc/paper_files/paper/2024/hash/28ab418242603e0f7323e54185d19bde-Abstract-Conference.html.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 -*

16, 2023, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/6dcf277ea32ce3288914faf369fe6de0-Abstract-Conference.html.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pp. 26286–26296. IEEE, 2024a. doi: 10.1109/CVPR52733.2024.02484. URL https://doi.org/10.1109/CVPR52733.2024.02484.

Zirui Liu, Jiayi Yuan, Hongye Jin, Shaochen (Henry) Zhong, Zhaozhuo Xu, Vladimir Braverman, Beidi Chen, and Xia Hu. KIVI: A tuning-free asymmetric 2bit quantization for KV cache. In Ruslan Salakhutdinov, Zico Kolter, Katherine A. Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, volume 235 of *Proceedings of Machine Learning Research*, pp. 32332–32344. PMLR / OpenReview.net, 2024b. URL https://proceedings.mlr.press/v235/liu24bz.html.

Zuyan Liu, Benlin Liu, Jiahui Wang, Yuhao Dong, Guangyi Chen, Yongming Rao, Ranjay Krishna, and Jiwen Lu. Efficient inference of vision instruction-following models with elastic cache. In Ales Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol (eds.), *Computer Vision - ECCV 2024 - 18th European Conference, Milan, Italy, September 29-October 4, 2024, Proceedings, Part XVII*, volume 15075 of *Lecture Notes in Computer Science*, pp. 54–69. Springer, 2024c. doi: 10.1007/978-3-031-72643-9_4. URL https://doi.org/10.1007/978-3-031-72643-9_4.

Yuzhang Shang, Mu Cai, Bingxin Xu, Yong Jae Lee, and Yan Yan. Llava-prumerge: Adaptive token reduction for efficient large multimodal models. In *IEEE/CVF International Conference on Computer Vision, ICCV 2025, Honolulu, HI, USA, October 19-25, 2025*, pp. 22857–22867. IEEE, 2025. doi: 10.1109/ICCV51701.2025.02122. URL https://doi.org/10.1109/ICCV51701.2025.02122.

Jianlin Su, Murtadha H. M. Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024. doi: 10.1016/J.NEUCOM.2023.127063. URL https://doi.org/10.1016/j.neucom.2023.127063.

Dezhan Tu, Danylo Vashchilenko, Yuzhe Lu, and Panpan Xu. Vl-cache: Sparsity and modality-aware KV cache compression for vision-language model inference acceleration. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=HMrcv7Q4Ub.

Pavan Kumar Anasosalu Vasu, Fartash Faghri, Chun-Liang Li, Cem Koc, Nate True, Albert Antony, Gokula Santhanam, James Gabriel, Peter Grasch, Oncel Tuzel, and Hadi Pouransari. Fastvlm: Efficient vision encoding for vision language models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11-15, 2025*, pp. 19769–19780. Computer Vision Foundation / IEEE, 2025. doi: 10.1109/CVPR52734.2025.01841. URL https://openaccess.thecvf.com/content/CVPR2025/html/Vasu_FastVLM_Efficient_Vision_Encoding_for_Vision_Language_Models_CVPR_2025_paper.html.

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 Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2024, Miami, Florida, USA, November 12-16, 2024*, volume EMNLP 2024 of *Findings of ACL*, pp. 4065–4078. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.FINDINGS-EMNLP.235. URL https://doi.org/10.18653/v1/2024.findings-emnlp.235.

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, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=NG7sS51zVF.

Cheng Yang, Yang Sui, Jinqi Xiao, Lingyi Huang, Yu Gong, Chendi Li, Jinghua Yan, Yu Bai, Ponnuswamy Sadayappan, Xia Hu, and Bo Yuan. Topv: Compatible token pruning with inference time optimization for fast and low-memory multimodal vision language model. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11-15, 2025*, pp. 19803–19813. Computer Vision Foundation / IEEE, 2025a. doi: 10.1109/CVPR52734.2025.01844. URL https://openaccess.thecvf.com/content/CVPR2025/html/Yang_TopV_Compatible_Token_Pruning_with_Inference_Time_Optimization_for_Fast_CVPR_2025_paper.html.

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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11-15, 2025*, pp. 19792–19802. Computer Vision Foundation / IEEE, 2025b. doi: 10.1109/CVPR52734.2025.01843. URL https://openaccess.thecvf.com/content/CVPR2025/html/Yang_VisionZip_Longer_is_Better_but_Not_Necessary_in_Vision_Language_CVPR_2025_paper.html.

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 Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, volume 267 of *Proceedings of Machine Learning Research*. PMLR / OpenReview.net, 2025. URL https://proceedings.mlr.press/v267/zhang25s.html.

Yuxin Zhang, Yuxuan Du, Gen Luo, Yunshan Zhong, Zhenyu Zhang, Shiwei Liu, and Rongrong Ji. Cam: Cache merging for memory-efficient llms inference. In Ruslan Salakhutdinov, Zico Kolter, Katherine A. Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, volume 235 of *Proceedings of Machine Learning Research*, pp. 58840–58850. PMLR / OpenReview.net, 2024. URL https://proceedings.mlr.press/v235/zhang24n.html.

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark W. Barrett, Zhangyang Wang, and Beidi Chen. H2O: heavy-hitter oracle for efficient generative inference of large language models. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/6ceefa7b15572587b78ecfcebb2827f8-Abstract-Conference.html.

A Implementation and Reproducibility

A.1 Models, Decoding, and Prompts

We follow the models, datasets, frozen-backbone setting, and matched non-visual cache protection described in the main paper. Full-validation runs use batch size one, greedy decoding, and random seed 42. The Qwen2.5-VL experiments use Python 3.10, PyTorch 2.5.1 with CUDA 12.1, Transformers 4.55.4, and `qwen-vl-utils` 0.0.14 on one 24-GB NVIDIA RTX 3090. The default configuration uses `max_pixels`=602112 and automatic model precision. The maximum numbers of newly generated tokens are 64 for QA, 96 for captioning, 32 for referring, and 16 for POPE. The official LLaVA KVCache-Factory reproduction uses Transformers 4.44.2 to preserve the released attention-monkeypatch semantics.

Prompts are fixed across methods. QA uses "Answer the remote-sensing image question concisely. Use a short phrase when possible." Captioning uses "Describe the remote-sensing image in one concise paragraph. Mention the main objects and their spatial relations. Do not use bullet points."

Table 5: Frozen GALP-KV parameters used for full-validation evaluation.

Component	Frozen setting
Random seed	42
Observation window	32 queries
Attention pooling	AvgPool, kernel 7
Stable tail set	16 visual entries
Uniform anchors	Ratio 0.25, minimum 8; caption stable route
Risk threshold	$\tau = 0.37$
Spatial proxy-query ratio	0.016
Spatial geometry weight	0.070 directional; 0.040 coverage
Spatial diversity weight	0.024
Non-spatial diversity weight	0.026
Guarded replacements	At most one

Referring requests one box in the benchmark coordinate format, and POPE requests only a yes/no answer. The compression policy receives neither target answers nor ground-truth task labels.

A.2 Visual Span and Position Handling

For Qwen2.5-VL, visual-span identification, grid alignment, protected positions, and position-consistent gathering follow the definitions in the main paper. For LLaVA-1.5-7B, the runner first locates `image_token_index`. If the tokenizer exposes a single image placeholder, it is expanded to the contiguous visual span created inside the model:

$$M_{\text{vis}} = N_{\text{cache}} - (N_{\text{input}} - N_{\text{placeholder}}) . \tag{27}$$

Tokens after the placeholder are shifted by the same expansion amount, and the expanded visual sequence preserves projector raster order. Because the LLaVA experiment evaluates behavioral transfer rather than retained-layout geometry, no LLaVA layout metric is reported.

A.3 GALP-KV Policy Parameters

Table 5 lists the frozen full-validation configuration. Caption detection is a deterministic template check for the phrases "describe the remote-sensing image" and "describe this remote-sensing image." The direction parser maps {top, upper, north}, {bottom, lower, south}, {left, west}, and {right, east} to rendered-image coordinates and also recognizes corner and center expressions. In the stable spatial route, guarded correction replaces at most one low-utility visual state while protecting the highest-attention portion of the selected set; caption prompts do not use this replacement. If the caption anchor union exceeds the visual budget, it is truncated by attention utility; otherwise, the remaining budget is filled with the highest-utility visual states.

A.4 Layout Diagnostics

For a selected visual set S on an $H \times W$ grid, the four image quadrants define counts $n_1, \ldots, n_4$ and probabilities

$$p_q = \frac{n_q}{\sum_{j=1}^{4} n_j}. \tag{28}$$

The normalized layout entropy is

$$H_{\text{layout}} = -\frac{1}{\log 4} \sum_{q=1}^{4} p_q \log p_q, \tag{29}$$

Table 6: Parameter sensitivity on fixed development subsets. Threshold rows use $r = 0.435$; geometry (Geo.) and diversity (Div.) rows use $r = 0.35$.

Setting	QA Rel. ↑	mIoU ↑	Ref. Acc. ↑	ΔCIDEr ↑
$\tau = 0.30$	0.488	0.4058	0.4600	+0.0171
$\tau = 0.37$	0.488	0.4058	0.4600	+0.0171
$\tau = 0.45$	0.488	0.4011	0.4467	+0.0171
Geo. $0.5\times$	0.490	0.3953	0.4467	+0.0121
Geo. $0.75\times$	0.488	0.3960	0.4467	+0.0121
Geo. $1.0\times$	0.490	0.3950	0.4467	+0.0121
Div. $0.0\times$	0.490	0.3938	0.4367	+0.0121
Div. $1.0\times$	0.490	0.3950	0.4467	+0.0121
Div. $2.0\times$	0.492	0.3899	0.4400	+0.0121

where zero-count quadrants contribute zero. Values near one indicate a balanced quadrant distribution. The largest empty radius is

$$R_{\text{empty}} = \max_{g \in \mathcal{G} \setminus S} \min_{s \in S} \frac{\|g - s\|_2}{\sqrt{H^2 + W^2}}, \tag{30}$$

where $\mathcal{G}$ is the complete visual grid. Lower values indicate smaller spatial blind regions. For matched-count analysis, each larger selected set is deterministically downsampled in original visual order to the minimum retained count among the compared methods before both statistics are recomputed.

B Additional Quantitative Diagnostics

B.1 Caption Generation Audit

All caption outputs use the same prompt, PTB tokenization, and evaluation backend. No method in the common eviction interface produces empty outputs or prompt mismatches at $r = 0.435$. Full KV and GALP-KV generate 74.5 and 73.4 words on average, reach the 96-token cap on 3,907 and 3,436 samples, and have repeated 3-gram ratios of 0.0081 and 0.0080, respectively. SnapKV, H2O, and NACL generate 73.6, 74.1, and 73.9 words on average, with repeated 3-gram ratios of 0.0081, 0.0088, and 0.0086. Under the corrected Elastic Cache merge protocol, the mean output length is 45.3 words, compared with 74.5 for Full KV. This supports the main-paper interpretation that the higher Elastic Cache caption scores partly reflect altered generation length and style.

B.2 Stable, Proxy, and Gated Policies

At $r = 0.35$, the stable route obtains QA relaxed accuracy 0.5413, referring mIoU 0.3834, caption CIDEr 0.0434, layout entropy 0.9771, and empty radius 0.1090. The proxy route improves QA, referring, and captioning to 0.5420, 0.3934, and 0.0486, but its layout entropy falls to 0.5940 and empty radius rises to 0.3679. The gated policy matches proxy QA and referring (0.5420 and 0.3934), uses the stable caption route (CIDEr 0.0434), and yields layout entropy 0.6730 with empty radius 0.3224. The gated design is therefore a task-dependent compromise rather than a policy that dominates both fixed routes.

B.3 Threshold and Proxy-Weight Sensitivity

Table 6 varies one parameter at a time. Thresholds 0.30 and 0.37 produce identical results at $r = 0.435$ because both use the stable branch. Raising the threshold to 0.45 activates the proxy branch and reduces referring mIoU by 0.0047. Geometry scaling between $0.5\times$ and $1.0\times$ changes mIoU by at most 0.0010, whereas doubling diversity reduces mIoU from 0.3950 to 0.3899. Caption values remain unchanged because caption prompts use the stable branch.

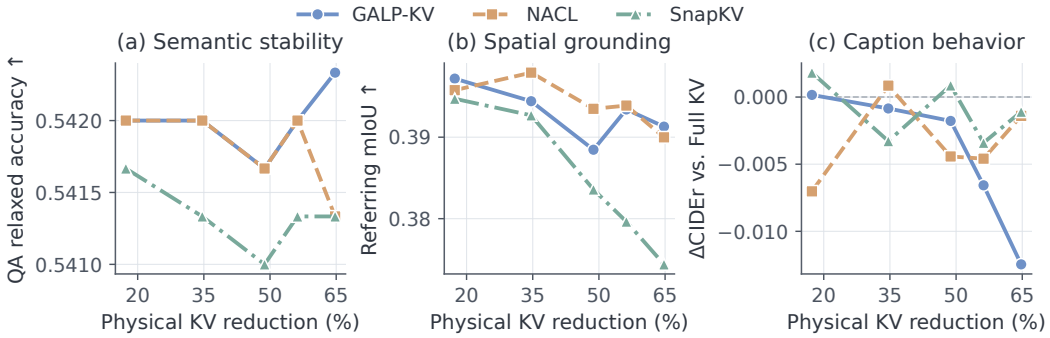


Figure 2: Five-budget quality–memory trade-off on Qwen2.5-VL diagnostic subsets.

Table 7: Exact Qwen2.5-VL budget–quality results on fixed diagnostic subsets. Caption values are corpus-level CIDEr changes from Full KV.

Visual r	Method	KV Red. ↑	QA Rel. ↑	mIoU ↑	Reg. ↓	ΔCIDEr ↑
0.800	SnapKV	17.32	0.5417	0.3947	19	+0.0018
0.800	NACL	17.32	0.5420	0.3958	21	-0.0070
0.800	GALP-KV	17.32	0.5420	0.3972	14	+0.0002
0.600	SnapKV	34.64	0.5413	0.3927	42	-0.0033
0.600	NACL	34.64	0.5420	0.3979	25	+0.0008
0.600	GALP-KV	34.64	0.5420	0.3944	34	-0.0009
0.435	SnapKV	48.76	0.5410	0.3836	75	+0.0009
0.435	NACL	48.76	0.5417	0.3935	35	-0.0044
0.435	GALP-KV	48.76	0.5417	0.3885	67	-0.0018
0.350	SnapKV	56.22	0.5413	0.3796	92	-0.0034
0.350	NACL	56.22	0.5420	0.3939	36	-0.0046
0.350	GALP-KV	56.22	0.5420	0.3934	42	-0.0066
0.250	SnapKV	64.75	0.5413	0.3744	123	-0.0011
0.250	NACL	64.75	0.5413	0.3900	62	-0.0014
0.250	GALP-KV	64.75	0.5423	0.3913	55	-0.0125

B.4 Exact Values for the Budget Sweep

Figure 2 and Table 7 report the exact values underlying the budget–quality trade-off. The fixed diagnostic subsets contain 3,000 QA, 2,000 referring, and 1,000 caption samples. QA remains nearly constant throughout the sweep, while grounding and captioning become more sensitive as the physical budget decreases. At the most aggressive setting, GALP-KV has the highest mIoU among the three selectors but also its largest negative CIDEr deviation, reinforcing the absence of a uniformly dominant selector.

B.5 Component Ablations and Extended Matched-Count Analyses

Causal component chain. Table 8 traces the construction of GALP-KV from attention-only selection to the complete policy under an identical retained count at *two* budgets. Uniform and tail anchors progressively improve layout (Panel A: entropy 0.9508 → 0.9743, radius 0.1456 → 0.1187; Panel B: 0.9276 → 0.9638 and 0.1760 → 0.1276). The isolated geometry step contributes no measurable marginal effect at either budget: layout statistics are identical to four decimals at r=0.435 and differ by < 0.001 mIoU at r=0.35. Full GALP-KV then jumps on Ref. mIoU by +0.0094 (A) and +0.0241 (B) while *losing* matched-count layout coverage in Panel B (entropy 0.9638 → 0.9437, radius 0.1276 → 0.1554). Because the Full pipeline improves downstream grounding even as its layout indicators regress, the causal attribution is unambiguous: **the gain comes from budget-gated per-head selection, not from the geometric prior or from the layout metrics themselves.** We

Table 8: Causal component chain at two budgets on Qwen2.5-VL (n=6,000 matched-count samples per cell). Adding uniform and tail anchors progressively improves layout; the geometric regularizer leaves every reported statistic unchanged at r=0.435 and moves Ref. mIoU by -0.0006 at r=0.35, while Full GALP-KV (adding per-head gated selection) contributes $+0.0094$ and $+0.0241$ respectively. Downstream gains are therefore driven by the budget-gated per-head selection, not by the geometric prior in isolation.

		Task			Layout	
Variant	Visual kept	QA Relaxed ↑	Ref. mIoU ↑	CIDEr ↑	Entropy ↑	Empty radius ↓
Panel A: Stable route, $r = 0.435$ (141 visual entries matched)						
Attention	141	0.5413	0.3838	0.0517	0.9508	0.1456
+Uniform	141	0.5417	0.3810	0.0509	0.9641	0.1235
+Tail	141	0.5410	0.3794	0.0482	0.9743	0.1187
+Geometry	141	0.5410	0.3791	0.0482	0.9743	0.1187
Full GALP-KV	141	0.5417	0.3885	0.0482	0.9832	0.0954
Panel B: Proxy route active, $r = 0.35$ (113 visual entries matched)						
Attention	113	0.5400	0.3740	0.0484	0.9276	0.1760
+Uniform	113	0.5403	0.3699	0.0455	0.9453	0.1310
+Tail	113	0.5397	0.3699	0.0434	0.9638	0.1276
+Geometry	113	0.5397	0.3693	0.0434	0.9638	0.1276
Full GALP-KV	113	**0.5420**	**0.3934**	0.0434	0.9437	0.1554

therefore retain the geometric term only as a structural regularizer and report layout statistics as diagnostics of retained-cache geometry rather than as targets of optimization.

Main-budget matched-count comparison. These statistics are recomputed on the *true* retained visual positions; an earlier logging placeholder that reported compressed-cache indices instead of original sequence positions had collapsed the SnapKV and NACL measurements onto the same contiguous slice and understated their coverage. On the corrected positions at the main budget r=0.435, GALP-KV attains the smallest largest-empty-radius of all selectors, and a paired session bootstrap over the full 600 images with 10,000 resamples shows this blind-region advantage is statistically significant against every baseline, including the strongest NACL ($\Delta = -0.0024$, 95% CI $[-0.0037, -0.0011]$). Its quadrant entropy is the second highest, marginally below NACL, which is slightly more uniform ($\Delta = -0.0035$, 95% CI $[-0.0044, -0.0026]$).

Extended matched-count budgets. This blind-region ordering is budget-specific: under a second matched budget (r=0.55, 178 retained visual tokens per sample, same 600 sessions) GALP-KV still leads the question-agnostic selectors by a wide, significant margin (empty-radius Δ against SnapKV, H2O, and ElasticCache all in $[-0.084, -0.040]$, 95% CIs excluding 0) and stays within 0.002 quadrant entropy of NACL, but there NACL attains a marginally smaller empty radius than GALP-KV ($\Delta = +0.005$, 95% CI $[+0.004, +0.006]$). GALP-KV and NACL therefore form a common top tier on both structural metrics at either budget, well ahead of the non-adaptive selectors, while the razor-thin ordering between these two adaptive methods flips with the compression level; we accordingly report GALP-KV's layout advantage as a tier-level rather than always-first claim. We extend the matched-count sweep to a more aggressive budget (r=0.35, 113 retained visual tokens, same 600 sessions) that exercises GALP-KV's *head-specific* route rather than the shared-layout route used above. Because that route retains a different visual subset for each KV head, there is no single shared keep set; we report the true original positions of one representative head (layer 0, KV-head 0) whose retained set is nonetheless count-matched to the baselines, and explicitly do not treat this single head as the per-head union. On these corrected positions the same tier ordering persists: GALP-KV's empty radius is significantly smaller than SnapKV, H2O, and ElasticCache (Δ in $[-0.095, -0.042]$, 95% CIs excluding 0) and its quadrant entropy exceeds all three, while NACL again edges GALP-KV ($\Delta_{\text{radius}} = +0.014$, CI $[+0.012, +0.017]$). The head-specific route is also the most question-adaptive: its cross-query Jaccard falls to 0.355 and its frozen-cache transfer to 0.523, against $\approx$ 0.99–1.00 transfer for the non-adaptive saliency selectors (SnapKV, H2O, ElasticCache). This is the geometric signature the method section predicts for per-head, per-question retention—it sharpens the adaptivity–reusability tension rather than contradicting the layout result, and it sup-

Table 9: Paired referring differences on layout-sensitive subsets at $r = 0.435$. Differences are GALP-KV minus NACL; absolute values for the most informative subsets are stated in the text.

Subset	Samples	Diff.	95% CI
All	16,153	-0.0042	$[-0.0060, -0.0023]$
Small object	6,646	+0.0001	$[-0.0018, 0.0020]$
Edge object	9,582	+0.0004	$[-0.0017, 0.0024]$
Corner object	2,296	+0.0019	$[-0.0019, 0.0060]$
Center object	1,920	-0.0143	$[-0.0208, -0.0079]$
Directional	14,674	-0.0031	$[-0.0049, -0.0013]$
Relation	2,182	-0.0037	$[-0.0081, 0.0009]$
Small or edge	11,535	0.0000	$[-0.0019, 0.0018]$

plies the direct evidence for the aggressive-compression branch that the earlier placeholder-logged r=0.35 run could not.

Geometry-prior causal ablation. To probe whether the geometry prior is causally helpful, we run a matched-budget referring ablation (VRSBench referring dev300, $r = 0.435$) that enables the components in isolation. Every isolated step lowers paired referring mIoU relative to Full KV, and a per-sample paired bootstrap (over the 300 referring samples, 10,000 resamples) sharpens which regressions are real: adding uniform coverage anchors ($\Delta = -0.018$, 95% CI $[-0.034, -0.004]$) and adding the geometry term ($\Delta = -0.020$, CI $[-0.038, -0.003]$) regress significantly, whereas attention-only top-k ($\Delta = -0.012$, CI $[-0.027, +0.003]$) and the shuffled fixed-proxy-grid control ($\Delta = -0.011$, CI $[-0.027, +0.003]$) fall within noise. The complete budget-gated GALP-KV policy is statistically indistinguishable from Full KV ($\Delta = +0.0016$, CI $[-0.013, +0.016]$). The gate is therefore what *prevents* the measurable damage that the isolated geometric additions inflict, rather than delivering a significant gain at this sample size. The same ablation on VRSBench captioning (dev300, r=0.435) is near-null in the corpus-aggregated CIDEr/BLEU (every variant within 0.004 of Full KV; complete GALP-KV +0.0017 CIDEr), but per-image paired METEOR/ROUGE-L expose a small yet significant regression for the isolated geometry step (e.g. ΔROUGE-L -0.010, CI $[-0.014, -0.005]$) that the full gate drives back to the smallest, partly insignificant change (GALP-KV ΔMETEOR -0.0026, CI $[-0.006, +0.001]$). Caption quality thus neither gains from an isolated geometric term nor is it strongly disturbed by the complete policy, consistent with the geometry prior chiefly guarding spatially sensitive grounding rather than improving free-form generation.

C Paired Statistics and Spatial Failure Subsets

C.1 Additional Paired Comparison with NACL

We use 10,000 paired bootstrap replicates at $r = 0.435$. The main paper reports QA relaxed accuracy, referring mIoU, and caption CIDEr. The remaining metrics show the same limited-superiority pattern: QA exact has mean difference $+0.000160$ with 95% CI $[-0.000214, 0.000535]$; referring accuracy is -0.005136 with CI $[-0.008664, -0.001733]$; caption BLEU-4 is -0.001913 with CI $[-0.002605, -0.001233]$; and caption ROUGE-L is -0.003822 with CI $[-0.004595, -0.003036]$. Positive differences favor GALP-KV. Thus, exact QA is statistically indistinguishable, while the additional referring and caption metrics favor NACL.

C.2 Layout-Sensitive Referring Subsets

Table 9 reports deterministic subsets derived from target boxes and prompt keywords. Small, edge, corner, relation, and combined small-or-edge subsets have intervals that include zero. The clearest GALP-KV deficit occurs for center objects, where mIoU is 0.4377 versus 0.4520 for NACL; directional prompts also favor NACL, with 0.3727 versus 0.3757. On the complete parsed set, the corresponding values are 0.3848 and 0.3890. Broader retained-cache coverage therefore does not imply better grounding for every spatial condition.

Table 10: Official KVCache-Factory reproduction on the complete LLaVA-1.5-7B VRSBench QA validation split. KV reduction is measured on the inspection subset; runner-dependent latency is not compared.

Method	Relax. ↑	Δ Rel. ↑	KV Red.	Changed ↓
Full KV	0.5400	+0.0000	0.00	0
SnapKV	0.4853	-0.0546	53.76	14,304
H2O	0.4975	-0.0425	53.76	15,535
PyramidKV	0.5069	-0.0330	53.51	9,571
StreamingLLM	0.5303	-0.0096	53.76	6,147
L2Norm	0.5311	-0.0088	50.40	7,477
CaM	0.0107	-0.5292	53.76	37,222

D Baseline Protocols and Fairness

D.1 Matched Qwen Protocol

The matched Qwen protocol fixes the backbone, prompts, decoding, visual retention count, and protected non-visual positions. SnapKV follows its released observation-window scoring rule, applies it only to the identified visual span, and averages layer scores to obtain a shared physical keep set. H2O and NACL are likewise restricted to visual candidates. The suffix "adapted" is omitted from table names, but these rows use visual-span interfaces rather than unmodified language-only runners.

Elastic Cache is a merging method rather than pure eviction. The corrected `visual_protected_v2` adapter protects all non-visual states, selects exactly $k = \mathrm{round}(rM)$ visual anchors per layer from prefill attention, and assigns each dropped visual state to the nearest selected anchor in original sequence order. Each anchor key and value is replaced by the arithmetic mean of the anchor and assigned visual states. The final tensor matches the physical visual length and byte budget of the eviction methods, but an Elastic Cache anchor contains merged information whereas GALP-KV, SnapKV, H2O, and NACL retain unmodified states. The comparison is therefore a matched-budget evaluation, not a pure selector ranking.

D.2 Official LLaVA KVCache-Factory Protocol

Table 10 reports the official KVCache-Factory reproduction under Transformers 4.44.2. This external implementation audit is separate from the matched-runner ranking in the main paper. StreamingLLM and L2Norm preserve relaxed accuracy most closely, whereas SnapKV, H2O, and PyramidKV show larger degradation and CaM exhibits severe performance degradation under this runner. Physical allocation remains method dependent: most rows retain a representative sequence length of 286, PyramidKV retains 551 despite 53.51% byte reduction, and L2Norm retains 618.48 despite a 50.40% byte reduction. This confirms that representative sequence length is not proportional to physical bytes under nonuniform layer- or head-wise allocation.

D.3 VLM-Specific Baseline Audit

LOOK-M provides an official modified LLaVA/LLaMA stack, but its text-prior policy compresses the full multimodal sequence and performs per-head merging under an older Transformers environment. Extracting only part of that algorithm into the Qwen visual-only runner would not be an official reproduction. At the time of the audit, no verified author implementation was available for VL-Cache or AirCache. We therefore rank Elastic Cache through the corrected matched-budget adapter, report LOOK-M only as an audited protocol, and avoid self-written VL-Cache or AirCache scores. This reduces baseline coverage but prevents protocol-incompatible rankings.

Table 11: Prototype runtime (Qwen2.5-VL, GALP-KV v41, 24-image VRSBench caption subset, r=0.435; all compressed selectors cut resident visual KV by 48.2%). Cap is `max_new_tokens`, Out. is the mean generated length, latency is total generate wall-clock in ms, Tok/s is throughput. At caps 64 and 128 all methods except ElasticCache reach the forced length, so latency is directly comparable; beyond 128 tokens methods stop at different lengths and total latency is no longer comparable. Peak CUDA allocation is flat across all compressed selectors (7,460.6 MiB) and not reduced relative to Full KV.

Method	Cap	Out.	Latency	Tok/s
Full KV	64	64.0	979.4	65.5
SnapKV	64	64.0	1,126.3	56.8
H2O	64	64.0	1,129.1	56.7
NACL	64	64.0	1,121.9	57.1
ElasticCache	64	64.0	1,373.5	46.6
GALP-KV	64	64.0	1,079.6	59.3
Full KV	128	128.0	1,997.6	64.1
SnapKV	128	128.0	1,990.2	64.3
H2O	128	128.0	1,992.2	64.3
NACL	128	128.0	2,000.2	64.0
ElasticCache	128	117.8	2,152.1	54.7
GALP-KV	128	128.0	1,979.0	64.7

E System and Runtime Diagnostics

E.1 Prototype Runtime

At a matched physical budget (r=0.435) on Qwen2.5-VL (24-image VRSBench caption subset, GALP-KV = v41), every compressed selector reduces resident visual KV by 48.2% at every generation cap (Table 11). Because the prototype exposes prefill attention tensors and materializes the full cache before compressing, this byte reduction does not lower the *peak* CUDA allocation: measured peaks stay flat at 7,460.6 MiB for all compressed selectors versus 7,235.9–7,460.6 MiB for Full KV, so we claim physical-KV reduction rather than serving-capacity savings. On wall-clock, the fixed prefill-and-gather overhead is un-amortized at short outputs—at a forced 64-token generation GALP-KV is 10.5% slower than Full KV (paired per-sample bootstrap $\Delta = +10.5\%$, 95% CI $[+8.3\%, +11.9\%]$)—yet GALP-KV carries the lowest overhead among the compressed selectors (1,080 ms vs. 1,126/1,129/1,122/1,374 ms for SnapKV/H2O/NACL/ElasticCache). By a forced 128 tokens—where every selector emits the identical length, so total time is directly comparable—the overhead is fully amortized and GALP-KV turns marginally *faster* than Full KV (1,979 vs. 1,998 ms; paired bootstrap $\Delta = -0.93\%$, 95% CI $[-1.10\%, -0.76\%]$, single-pass timing), while retaining 48.2% fewer KV bytes; direct equal-length tests place the empirical crossover between 64 and 128 tokens, and a pooled two-parameter cost model (§E.2) gives a more conservative amortization length $L^* \approx 180$, the earliest among all compressed selectors. Beyond these forced caps the methods stop at different lengths, so total latency is no longer comparable across methods; we therefore characterize the prototype as reaching parity with, and marginally improving on, Full KV at matched length rather than delivering a robust end-to-end speedup.

E.2 Runtime Cost Model and Amortization Crossover

To turn the observed crossover into a predictive statement, we fit a two-parameter runtime model to every selector, $\text{latency}(L) \approx \Delta F + t\,L$, where L is the number of generated tokens, t is the per-token decode time, and ΔF is the intercept gap over Full KV—the one-time selection/compression overhead (prefill is common to all selectors for a fixed image). Ordinary least squares over all 96 $(L, \text{latency})$ points per selector (24 images $\times$ 4 caps) yields $R^2 \geq 0.997$ throughout (Table 12, Fig. 3).

Two structural facts follow. First, every layout-aware selector has a *lower* per-token decode time than Full KV ($t \approx 14.3$ vs. 15.07 ms/token): a smaller resident visual KV decodes strictly faster, so compression carries *no* decode-phase penalty. Second, this benefit must repay the fixed overhead

Table 12: Fitted two-parameter runtime cost model latency$(L) \approx \Delta F + t\,L$ (OLS over 24 images $\times$ 4 caps, $R^2 \geq 0.997$). ΔF is the intercept gap over Full KV (one-time selection/compression overhead); t is per-token decode time; L^* is the generated-token length at which the selector's total time meets Full KV. Smaller is better for both ΔF and L^*; GALP-KV amortizes earliest.

Method	ΔF (ms)	t (ms/tok)	L^* vs Full
GALP-KV	**119**	14.40	**178**
SnapKV	160	14.30	206
NACL	168	14.30	219
H2O	173	14.35	240
ElasticCache	358	15.11	> 512
Full KV	0	15.07	–

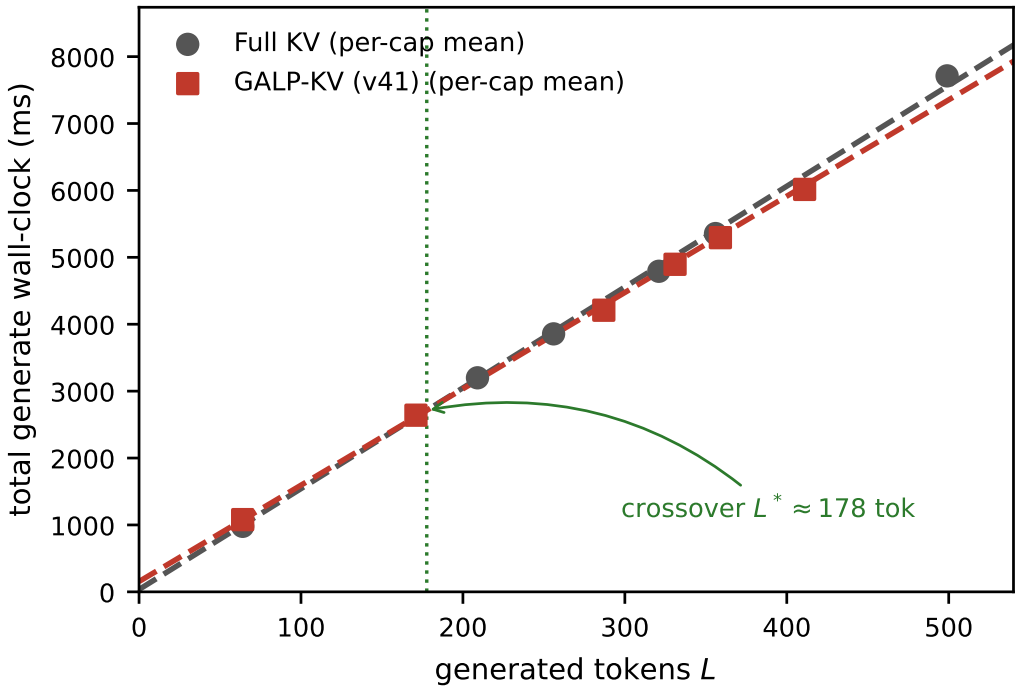


Figure 3: Total generate wall-clock vs generated tokens for Full KV and GALP-KV (24-image VRSBench caption subset, r=0.435). Markers are per-cap means, dashed lines are the OLS cost-model fits, and the dotted vertical line marks the fitted crossover $L^* \approx 178$ tokens, beyond which the memory saving carries no runtime cost.

ΔF; the amortization length is $L^* = \Delta F/(t_{\text{Full}} - t)$. GALP-KV has the smallest overhead among the compressed selectors ($\Delta F \approx 119$ ms) and therefore the earliest crossover, $L^* \approx 178$ tokens, ahead of SnapKV (≈ 206), NACL (≈ 219), and H2O (≈ 240). ElasticCache both pays the largest overhead ($\Delta F \approx 358$ ms) and decodes no faster than Full KV ($t \approx 15.1$), so it never crosses within the 512-token range. The 48.2% KV reduction is thus obtained at no per-token runtime cost and repays its one-time selection overhead within a short low-hundreds-token window, sooner than any competitor. This is deliberately *not* an end-to-end speedup claim: below L^* the overhead dominates, the fit carries a $\sim$50 ms residual that shifts L^* (direct equal-length paired tests already reach parity by 128 tokens), and peak allocation is unchanged.

Table 13: Behavioral fidelity vs. Full KV on all 37,409 VRSBench QA pairs at $r = 0.435$. Damaged = Full-correct→wrong; Recovered = Full-wrong→correct; Net = Recovered−Damaged.

Method	Damaged ↓	Recovered ↑	Net ↑	Δ Relaxed
GALP-KV	**371**	348	**-23**	-0.0006
H2O	372	343	-29	-0.0008
NACL	375	331	-44	-0.0012
SnapKV	387	344	-43	-0.0011
ElasticCache	440	479	+39	+0.0010

E.3 Stage-Level CUDA Memory

The post-compression KV component decreases from 13.36 to 6.93 MiB. For Full KV, overall CUDA peaks are 7,235.9, 7,460.6, and 7,460.6 MiB at maximum generation caps 128, 512, and 1,024; the corresponding decode peaks are 7,194.6, 7,206.7, and 7,206.7 MiB. For GALP-KV, prefill and overall peaks are 7,460.6 MiB at all three caps, while compression and decode peaks are 7,354.6 and 7,363.1 MiB. Model weights, exposed attention tensors, and temporary buffers dominate the allocation peak, so the prototype verifies physical KV reduction rather than total peak-memory reduction.

F Behavioral Fidelity and Cross-Query Reusability Details

This appendix provides the full analyses and tables behind the main-text summary in Sec. 4.5.

F.1 Behavioral Fidelity under Compression

The aggregate accuracy changes little for every selector ($|\Delta$ relaxed ≤ 0.0013 at $r = 0.435$), which can hide how much generation each method actually disturbs. We therefore decompose per-sample changes against Full KV on all 37,409 VRSBench QA pairs into *damaged* cases (Full KV correct $\rightarrow$ compressed wrong) and *recovered* cases (Full KV wrong $\rightarrow$ compressed correct). Because most flips cancel in aggregate, the damaged/recovered split rather than the net accuracy is the informative fidelity signal. As Table 13 shows, GALP-KV produces the fewest net regressions and the fewest damaged images among the deterministic selectors, moving Full-KV behavior the least while saving 6.75 MiB of KV per question.

F.2 Cross-Query Cache Reusability

A physical cache is often reused across several questions about the same image, so a question-agnostic keep set is more reusable than a question-specific one. We measure this at prefill on 600 multi-question VRSBench image-sessions (average 4 questions per image): for two questions A, B of the same image, $\mathrm{transfer}(B \mid A)$ is the fraction of B's retained visual evidence that is still present in the cache selected for A (Table 14). SnapKV, H2O, and ElasticCache are nearly query-invariant at prefill (transfer ≈ 0.99–1.00, union/single ≈ 1.0): their shared-prefix cache is effectively lossless across questions but barely adapts to the question. GALP-KV and NACL genuinely re-select per question: a cache frozen on one question drops 11.4% (GALP-KV) and 17.3% (NACL) of another question's visual evidence, and NACL is the most path-dependent selector (lowest order-invariant core $\cap/\cup = 0.56$). This exposes an adaptivity versus reusability trade-off that single-question accuracy cannot see. GALP-KV's shared stable route is the more reusable end of this spectrum. We then test end to end whether these prefill gaps change final answers: over 100 image-sessions we freeze the cache selected for the first question and answer every later question from it, replaying the anchor's visual keep set at the target's absolute positions while always retaining the target's non-visual tokens (320 target-question pairs). Relative to each selector's own per-question cache, frozen reuse changes relaxed QA accuracy by at most 0.006 (GALP-KV, H2O, and ElasticCache within 0.000; SnapKV -0.003; NACL -0.006), with an answer-flip rate $\leq 0.6\%$ at a healthy own-question accuracy of ≈ 0.58. The few predictions that do change are paraphrases that relaxed scoring treats as identical (e.g. "cloud" vs. "clouds"). The prefill-level evidence loss of the adaptive selectors therefore does not translate into measurable short-answer degradation: the protected non-

Table 14: Cross-query cache reusability at prefill (600 VRSBench QA image-sessions, $r = 0.435$). transfer$(B \mid A)$: fraction of B's visual evidence kept by a cache frozen on A. $\cap/\cup$: order-invariant shared core; $\cup$/single: union over a single query.

Selector	transfer$(B \mid A)\uparrow$	$\cap/\cup\uparrow$	$\cup$/single
SnapKV	0.992	0.970	1.02
H2O	1.000	1.000	1.00
ElasticCache	1.000	0.999	1.00
GALP-KV	0.886	0.672	1.23
NACL	0.827	0.557	1.33

visual states plus the retained visual span remain answer-sufficient, which bounds the practical cost of query adaptation on QA even though grounding remains more sensitive.

F.3 POPE Natural-Image Transfer

Table 15 evaluates natural-image transfer on POPE at $r = 0.35$. POPE measures object hallucination using binary object-existence questions with random, popular, and adversarial negative-object sampling Li et al. (2023). Relative to Full KV, GALP-KV changes accuracy by $+0.0017$, $+0.0007$, and -0.0003 across the three splits while maintaining a parse rate of 1.0. SnapKV, NACL, and GALP-KV produce identical predictions, supporting cross-domain non-degradation rather than selector-specific superiority.

Table 15: POPE natural-image transfer at $r = 0.35$.

Method	Random Acc. ↑	Popular Acc. ↑	Adversarial Acc. ↑	Parse Rate ↑
Full KV	0.9050	0.8897	0.8373	1.0000
SnapKV	0.9067	0.8903	0.8370	1.0000
NACL	0.9067	0.8903	0.8370	1.0000
GALP-KV	0.9067	0.8903	0.8370	1.0000