FUND REGIONS, NOT TOKENS: QUERY-CONDITIONED BUDGET ALLOCATION FOR EXTREME KV-CACHE EVICTION IN VISION-LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision-language models (VLMs) need thousands of visual tokens for multi-page documents and high-resolution charts. The per-request key–value (KV) cache limits how many requests a device can serve concurrently. Existing eviction methods from text LLMs keep the global top-k tokens of a one-dimensional sequence. Visual evidence, however, occupies contiguous two-dimensional regions, and a region stays usable only if enough of its tokens remain. At extreme budgets of 1–3%, a global ranking spreads the cache over many locations and leaves each region too few tokens. To this end, we propose CARVE, which treats KV eviction as query-conditioned budget allocation to regions. CARVE partitions visual tokens into layout-aware cells, commits most of the budget to a few query-ranked regions, and selects tokens within them. When selection-time diagnostics find this allocation unsuitable, CARVE falls back to attention-based selection, and its gated mode CARVE-G to conservative token-level eviction. Experiments cover four models and four benchmarks at equal KV budget. On DocVQA at a 1% budget, CARVE retains 74.4% of the full-cache score on InternVL3-8B, where the strongest text-guided baseline retains 40%. CARVE-G stays within 1.6 points of its text-guided fallback in all sixteen model–budget settings.

1 INTRODUCTION

High-resolution documents, charts, and infographics form large visual contexts in vision-language models (VLMs) (Liu et al., 2024a; Bai et al., 2025b; Zhu et al., 2025). During decoding, each request keeps a key–value (KV) cache of its input in the transformer decoder (Vaswani et al., 2017), and this cache grows linearly with the context length, so per-request context memory is a major constraint in concurrent serving (Pope et al., 2023). *Budget-bounded* cache eviction alleviates this constraint by compressing the context cache to a fixed budget before the first token is decoded, which reduces the memory that each request retains during decoding. The challenge is to preserve the evidence needed to answer the query as the retained budget shrinks.

KV eviction was originally developed for text LLMs, and its core idea reflects this origin. It assigns a scalar saliency to each token, computed by accumulated attention (Zhang et al., 2023), observation-window attention (Li et al., 2024), or positional recency (Xiao et al., 2024), and keeps the global top-k along the sequence dimension. This strategy works well at moderate compression, but its performance drops sharply at the 1–3% budgets that we study here. Extreme budgets also expose a question that moderate compression hides. At 1–3%, how the budget is allocated across the image decides whether the evidence survives. Under matched total budgets, attention-based baselines lose 26–54 ANLS points on DocVQA relative to the full cache, and even the strongest text-guided method retains only 40% of the full-cache DocVQA score at a 1% budget. These results raise a question that token ranking alone cannot answer. Do the retained entries, taken together, preserve enough local context to answer the query?

We argue that, at these extreme budgets, improving token scores alone is insufficient because the selection rule still allocates budget independently across scattered tokens. The evidence needed to answer a visual question, such as a table row, a chart legend, or a totals box, occupies a contiguous two-dimensional region of the image. A 1D top-k over token saliencies is blind to this geometry: it

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.

keeps scattered singletons and evicts the neighbors that make them readable, such as the rest of the row or the axis label beside a data point. At extreme budgets a region needs a minimum number of tokens to stay usable, and a global ranking spreads the tiny cache too thinly to give any region that number. The question is therefore not which tokens to keep but which regions to fund, and whether to commit the budget to them at all.

We implement this idea in CARVE (Confidence-gated Adaptive Region-based Visual-budget Eviction). CARVE partitions the visual tokens into layout-aware 2D cells, scores each cell against the query with a text-grounded relevance signal, commits most of the budget to a few query-ranked regions, and selects tokens within them, so the retained evidence stays concentrated where the question points. A stylized analysis supports this design: the optimal degree of concentration grows with the reliability of the region scores and as the budget shrinks, recovering uniform allocation when the scores are uninformative and full concentration when they are perfect. Because no scoring signal is reliable on every input, selection-time diagnostics decide when to commit: when the text signal is uninformative, selection switches to accumulated attention, and when the score statistics indicate unreliable region evidence, the gated mode CARVE-G falls back to conservative token-level eviction. On full evaluation sets, CARVE-G preserves the extreme-budget gains with a worst single-benchmark regression of 1.6 points relative to the text-guided baseline. To our knowledge, CARVE is the first eviction method that makes 1–3% budgets a practical operating point.

In summary, our main contributions include:

- We identify the allocation of the budget across regions as the key decision in extreme KV-cache eviction of visual contexts and cast it as a query-conditioned allocation to committed regions. We propose CARVE, which partitions the visual tokens into layout-aware 2D cells, commits the budget to a small set of query-ranked regions, and switches to attention-based top-k when the text signal is uninformative. A stylized analysis shows that the optimal concentration grows with the scorer reliability and with the budget severity, which motivates the commit-or-fall-back design.
- We propose CARVE-G, a signal-driven confidence gate that falls back to conservative token-level eviction. On full evaluation sets, its worst single-benchmark regression is 1.6 points across all sixteen model–budget settings, and its mean gain reaches up to $+19.1$ at the 1% budget. It also achieves the best average at the 10% budget on InternVL3.
- We show that CARVE makes 1–3% budgets practical. At the 1% budget, it retains 74.4% of the full-cache DocVQA score where the strongest text-guided baseline retains 40%, a gain of $+31.3$ on InternVL3-8B, across four models and four benchmarks under the same total KV budget. A matched-budget analysis of 60k samples identifies where the committed allocation helps, and decode latency is similar across methods.

2 Related work

KV-cache compression for text LLMs. Existing eviction methods score each token and keep the top-k, where the score is computed by accumulated attention (Zhang et al., 2023; Liu et al., 2023), observation-window attention (Li et al., 2024), attention sinks plus recency (Xiao et al., 2024), or per-layer and per-head budget schedules (Cai et al., 2024; Ge et al., 2024; Wang et al., 2025a). All of them treat the context as a one-dimensional sequence, so the position of a token in the 2D image space plays no role in what is kept. Orthogonal systems work reduces the per-token cache size (e.g., multi-query and grouped-query attention; Shazeer, 2019; Ainslie et al., 2023) or the attention IO cost (Dao et al., 2022), and budget-bounded eviction is complementary to both.

KV-cache compression for VLMs. Another line of research adapts cache compression to multimodal inputs and has made substantial progress on this problem. It allocates the budget across *layers* (Wan et al., 2025), *heads* (Wang et al., 2025b; Zeng et al., 2026), and *modalities* (Li et al., 2025; Tu et al., 2025), refines token-level scoring (Liu et al., 2026c; Chen et al., 2026a; Yang et al., 2026; Chen et al., 2026b), or merges cached entries (Wan et al., 2024; Liu et al., 2024b). Spatial signals have also begun to enter the scoring layer, including hidden-state saliency for screenshots (Huang et al., 2025), attention–coordinate mutual information (Han et al., 2026), state-space absorption that preserves visual continuity (Liu et al., 2026b), and compression of spatially coherent attention blocks (Huang et al., 2026). TGV-KV (Liu et al., 2026a) scores visual tokens by their grounding in the query text and is the strongest text-guided baseline in our comparison.

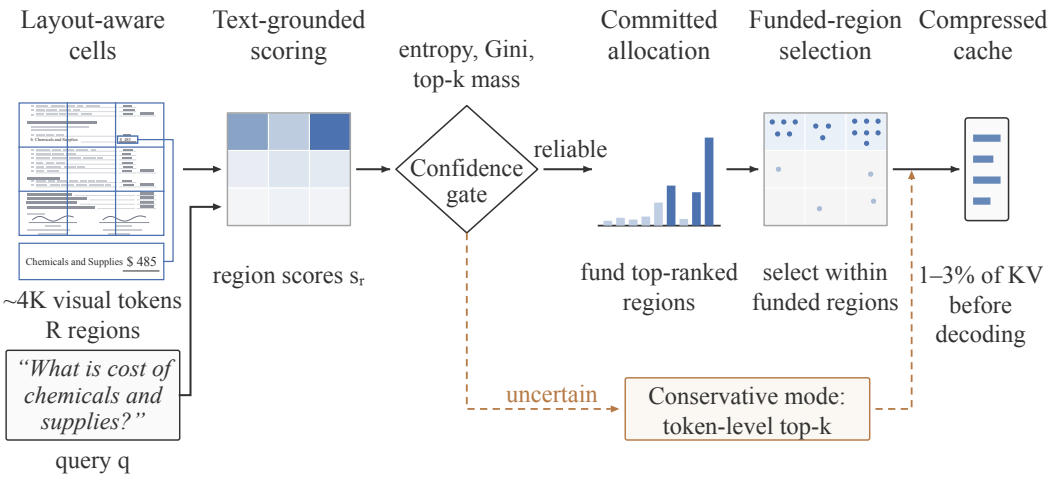

Figure 1: **The CARVE pipeline.** The text-grounded region path allocates the budget to the top-ranked spatial cells and selects tokens within them. CARVE-G adds confidence gating: when the region evidence is reliable, the region mode is activated; otherwise, the selection falls back to token-level top-k under the same budget. Compression is performed per layer before decoding. Preliminary switches are omitted. The query-relevant cells are outlined and magnified in the inset; downstream scores and token selections are schematic.

Input-side visual token pruning. A parallel line of research reduces the number of visual tokens that enter or propagate through the language model. FastV drops low-attention tokens after early layers (Chen et al., 2024), SparseVLM sparsifies tokens with text guidance (Zhang et al., 2025), PruMerge merges redundant tokens (Shang et al., 2025), and HiRED drops tokens for each high-resolution tile under a given budget (Arif et al., 2025). These methods exploit spatial structure such as tiles, grids, and regions to reduce the visual tokens processed by the language model.

Our distinction. CARVE builds on text-grounded token relevance but changes how the cache budget is allocated: it commits capacity to a small set of query-ranked 2D regions before selecting tokens within them. Spatial structure thus governs budget commitment, extending its use in token scoring and continuity preservation. CARVE-G further conditions this commitment on the reliability of region evidence, falling back to token-level eviction when that evidence is unreliable. While input-side pruning reduces the tokens processed during prefill, CARVE targets the context cache retained for decoding.

3 METHOD

3.1 PROBLEM SETUP

A VLM takes a prompt of N tokens as input, of which N_v are visual (up to $\approx$4,000 in our setting, dominating the prompt length) and the rest are text. A KV budget $\rho \in (0, 1]$ allows at most $B = \lceil \rho N \rceil$ cached entries per layer. Attention statistics are streamed per layer during prefill, with attention maps reduced immediately rather than accumulated. The context cache is compressed once before the first token is decoded. The decoding then uses this compressed cache. New token entries are appended in the same way for all methods, and the generation is limited to 32 tokens. All methods, including ours, use the same total KV cache budget. At $\rho = 0.01$ with $N \approx 4{,}000$, each layer retains $B = \lceil \rho N \rceil \approx 40$ entries.

Our memory target is the context cache retained per request during decoding. Techniques that bound transient prefill memory, such as chunked prefill and prefill peak-memory reduction (Kim & Bae, 2026), and memory-capped streaming encoding (Kim et al., 2025), are complementary to ours.

Token-level eviction solves $\max_{S:|S|=B} \sum_{i \in S} \alpha_i$ for a token saliency α, which amounts to a 1D top-k that is blind to the image geometry. We instead decompose the decision into two steps. We

first allocate the visual budget over 2D regions, and then select tokens within the funded regions. The allocation is the key decision at extreme budgets, and Section 6 isolates its contribution empirically.

3.2 LAYOUT-AWARE 2D CELLS

Modern VLMs expose an explicit 2D layout for their visual tokens. Examples include the anyres tile grids in LLaVA-NeXT (Liu et al., 2024a), the dynamically tiled sub-images in InternVL3 (Zhu et al., 2025), and the native patch grids in Qwen2.5-VL and Qwen3-VL (Bai et al., 2025b;a). CARVE partitions the visual tokens of each image into R contiguous rectangular cells $\{\mathcal{C}_r\}_{r=1}^{R}$ that are aligned with this native layout. A cell is therefore a spatially coherent block of the input, such as a slab of a page or a tile of a chart, instead of an interval of the flattened sequence. We give the per-model construction details in Appendix B. The method is otherwise identical across architectures.

3.3 TEXT-GROUNDED REGION SCORES

Each visual token i carries a scalar relevance α_i, which measures how strongly the model grounds it in the query text. CARVE treats this scorer as a pluggable component and operates only on α_i. In our experiments, we instantiate the scorer with the strongest text-guided relevance signal available (Liu et al., 2026a; Section 4). CARVE aggregates the token relevances within each cell into a region score s_r (Appendix B) and forms the region distribution

$$\pi_r(\tau) \;=\; \frac{\exp(s_r/\tau)}{\sum_{r'} \exp(s_{r'}/\tau)}, \qquad r = 1, \ldots, R, \tag{1}$$

where the temperature τ controls the spatial concentration. As $\tau \to 0$, only the top region is funded. As $\tau \to \infty$, the budget is spread uniformly. Our analysis below studies this family, while the implemented allocator fixes $\tau = 1$ on the normalized scores and commits the budget according to the region ranking (Section 3.4).

3.4 COMMITTED BUDGET ALLOCATION

The central design decision is whether to concentrate the budget. At extreme budgets, the trade-off clearly favors concentration, because a region needs a sufficient number of retained tokens to remain usable, and spreading a tiny cache over many regions risks starving all of them. At a 1% budget, even *blind* concentration outperforms blind spreading (Section 6.1). The region path of CARVE therefore commits the budget. It funds a small set of cells at the top of the region ranking and reserves most of the budget for the tokens inside them (Section 3.5).

Since the commitment is only as reliable as the ranking it follows, CARVE computes shape statistics of the region distribution as empirical diagnostics of the score reliability. From π, it forms

$$p \;=\; \lambda\, u + (1 - \lambda)\, \pi, \qquad \lambda = \tilde{\mathrm{H}}(\pi), \tag{2}$$

where u is uniform over regions and $\tilde{\mathrm{H}}(\pi) \in [0, 1]$ is the normalized entropy of π. Mixing with the uniform distribution preserves the ranking, so p and π yield the same funded set. These statistics are evaluated per layer at selection time. The entropy drives the signal switch of Section 3.5, and the entropy, Gini, and top-k-mass statistics of p drive the confidence gate of Section 3.6, whose region mode also weights the in-region ranking by p_r. The same rule is used for every task and model.

Why commit, and when. Proposition 1 (Appendix A) analyzes a stylized retrieval model with K regions, one of which contains the evidence needed to answer the query. The scorer ranks that region first with probability p, and the retrieval success is a concave, saturating function of the budget that the region receives. In the two-region case, the optimal concentration increases monotonically with the score reliability. For arbitrary K, the analysis establishes the two endpoints of the trade-off. The allocation is uniform when the ranking is independent of the answer region, and fully concentrated when the scores are perfect. In a closed-form instance, the optimal share of the top region is $\min\{1,\ \frac{1}{2} + \frac{c}{2B} \ln \frac{p}{1-p}\}$, which increases as the budget B shrinks, up to the cap. CARVE uses this trade-off to guide the mode selection. It prefers concentration when the budget is tight, and switches signals or falls back to a conservative selection when the diagnostics indicate unreliable scores. The measured gains of the commitment grow with the compression severity (Section 6).

3.5 FUNDED-REGION SELECTION

The region ranking determines the *funded set*, which consists of the top $\mathrm{round}(0.2B)$ cells for a per-layer budget B (Appendix B). We reserve most of the budget for this set and fill it by selecting the tokens with the highest base relevance α_i within the union of the funded cells. The remaining slots are filled in global relevance order over all unselected tokens, including both text and visual tokens. Text tokens therefore compete for the same total budget as visual tokens. In the gated region mode, p_r additionally weights the in-region ranking (Appendix B). Since the selection is restricted to a small set of funded regions, the retained tokens remain spatially clustered instead of scattering across the image, which keeps each funded region legible.

Signal-adaptive selection. No relevance signal is informative on every input. For chart-style queries, the text guidance often provides weak grounding, whereas the accumulated prefill attention concentrates on the plotted evidence. CARVE therefore compares the normalized entropy $\tilde{\mathrm{H}}$ of the text-grounded region distribution with that of the same cells scored by the accumulated attention. If the gap indicates that the attention signal is substantially more informative, the layer switches entirely to attention-based token-level top-k at the same budget. A single fixed threshold pair is shared across all tasks and models (Appendix B). The switch introduces no new allocation or modality parameters. Section 6.2 analyzes this switch in detail, including the reallocation toward visual tokens that it produces.

3.6 CONFIDENCE GATE

At a 1% budget, a misplaced allocation is unrecoverable. Therefore, CARVE monitors the statistics of the region-score distribution, including the entropy, the Gini coefficient, and the top-k mass, and applies a fixed decision rule. When these statistics indicate unreliable region evidence, the selection falls back to the conservative token-level allocation of the base scorer, which matches the strongest text-guided baseline at the same budget. The thresholds are fixed once and shared across all tasks and models. We denote this gated deployment mode CARVE-G. The switches of Section 3.5 act first and identically in both modes. The gate then evaluates each layer that remains on the region path. Layers that fail the gate revert to the conservative selection, while those that pass blend the region score into the in-region ranking (Appendix B). On full evaluation sets, CARVE-G preserves the extreme-budget gains, with a worst single-benchmark regression of 1.6 points relative to the fallback baseline, and it also improves the performance at larger budgets (Section 6.3).

4 EXPERIMENTAL SETUP

Models. We evaluate four VLMs that span three layout mechanisms. LLaVA-NeXT-Mistral-7B (Liu et al., 2024a; Jiang et al., 2023) uses anyres tiles, InternVL3-8B (Zhu et al., 2025) uses dynamic tiling, and Qwen2.5-VL-7B (Bai et al., 2025b) and Qwen3-VL-8B (Bai et al., 2025a) use native patch grids. Inputs are processed at resolutions that yield up to approximately 4,000 visual tokens.

Benchmarks and budgets. We use four benchmarks whose answers are grounded in localized visual evidence, covering DocVQA (Mathew et al., 2021) and InfographicVQA (Mathew et al., 2022) (ANLS), TextVQA (Singh et al., 2019) (accuracy), and ChartQA (Masry et al., 2022) (relaxed accuracy). We use the full standard evaluation splits throughout, i.e., the validation sets of DocVQA, InfographicVQA, and TextVQA and the 2,500-sample ChartQA test split. We evaluate KV budgets of 10%, 5%, 3%, and 1% (Table 1). Our analysis focuses on the extreme budgets, where the allocation decision matters most.

Baselines. We compare CARVE with TGV-KV (Liu et al., 2026a), which is the strongest text-guided baseline and our base scorer, and with SnapKV (Li et al., 2024), MixKV (Liu et al., 2026c) (per-head fused-importance eviction, re-implemented following the reference description), H2O (Zhang et al., 2023), PrefixKV (Wang et al., 2025a), and StreamingLLM (Xiao et al., 2024). All methods run through the same generation loop, are compressed at the same point, and use the same total KV cache budget. We also include ElasticCache (Liu et al., 2024b), which we evaluate at every reported budget on all four models, and our adaptation is verified against the reference implementation (Appendix B).

Table 1: **Main results at four KV budgets** (scores $\times 100$; DocVQA/InfoVQA ANLS, TextVQA accuracy, ChartQA relaxed accuracy; full evaluation sets, Section 3.1). Per budget–dataset column within each model group: best **bold**, second underlined, full cache in gray and excluded; ties follow unrounded values and share emphasis. Ret.: four-benchmark mean as a percentage of the matched full-cache mean (full cache = 100). ElasticCache is adapted as described in Appendix B; its sharp drop is intrinsic to the method. †Identical selections at 1%: SnapKV always keeps its 64-token window, which exceeds the budget (Appendix B).

		1% KV budget					3% KV budget					5% KV budget					10% KV budget				
	Method	Doc	Text	Chart	Info	Ret.	Doc	Text	Chart	Info	Ret.	Doc	Text	Chart	Info	Ret.	Doc	Text	Chart	Info	Ret.
InternVL3-8B	Full cache	89.8	80.0	82.3	69.6	100	89.8	80.0	82.3	69.6	100	89.8	80.0	82.3	69.6	100	89.8	80.0	82.3	69.6	100
	SnapKV	36.4†	44.0†	62.2†	46.5†	58.8	41.1	51.9	63.7	48.6	63.8	66.7	66.2	68.4	55.2	79.7	81.9	76.6	78.3	63.1	93.2
	H2O	40.7	51.4	**69.2**	48.5	65.2	51.4	61.4	71.6	52.2	73.5	58.1	65.4	75.9	55.5	79.3	70.2	72.1	80.5	59.9	87.9
	PrefixKV	40.0	52.9	49.1	44.7	58.1	52.0	61.9	59.2	44.8	67.8	56.5	65.3	60.6	42.7	70.0	59.9	68.8	74.4	38.5	75.1
	StreamingLLM	36.4†	44.0†	62.2†	46.5†	58.8	41.2	51.8	63.6	48.6	63.8	47.0	58.2	66.7	49.6	68.9	58.8	68.4	72.5	53.4	78.7
	ElasticCache	0.5	1.3	1.0	2.3	1.6	3.6	2.0	0.9	2.1	2.7	2.8	9.3	1.5	2.5	5.0	8.4	30.3	15.8	6.0	18.8
	MixKV	36.3	44.0	64.7	46.5	59.5	70.0	66.7	**74.7**	58.5	83.9	80.4	74.4	**79.2**	**64.0**	**92.6**	87.0	**78.6**	82.4	67.0	97.9
	TGV-KV	35.5	41.6	42.4	44.3	51.0	53.4	53.8	62.5	51.6	68.8	79.9	71.2	76.8	63.2	90.5	87.4	78.6	81.8	68.1	98.2
	CARVE	**66.8**	**66.2**	64.5	**54.2**	**78.2**	**76.7**	**74.2**	71.2	55.7	86.3	79.2	75.3	68.7	56.5	86.9	87.1	78.5	56.5	64.5	89.1
	CARVE-G	63.7	63.4	59.5	53.6	74.7	76.6	72.0	70.2	**59.9**	**86.6**	**80.8**	**75.9**	77.2	61.6	91.9	**87.7**	78.6	**82.7**	**68.8**	**98.8**
LLaVA-NeXT	Full cache	63.7	65.8	53.0	30.5	100	63.7	65.8	53.0	30.5	100	63.7	65.8	53.0	30.5	100	63.7	65.8	53.0	30.5	100
	SnapKV	27.2	42.3	30.6	22.9	57.8	29.0	44.5	29.8	24.4	60.0	42.4	53.6	32.4	25.9	72.5	57.3	60.8	44.7	28.3	89.8
	H2O	30.0	49.3	**34.0**	22.1	63.7	37.5	56.8	38.4	22.2	72.8	43.1	59.3	42.9	24.5	79.7	51.6	62.6	46.8	26.6	88.1
	PrefixKV	29.4	49.4	29.2	17.9	59.1	40.8	57.6	37.5	20.2	73.3	45.7	59.7	40.0	21.5	78.4	52.4	62.1	42.9	23.9	85.1
	StreamingLLM	27.2	42.3	30.6	22.9	57.8	27.9	42.8	29.7	24.4	58.6	28.1	43.1	30.0	24.3	58.9	29.6	44.5	29.8	24.9	60.5
	ElasticCache	1.8	2.5	1.0	1.3	3.1	4.8	12.3	4.3	3.0	11.5	8.9	20.7	6.4	3.8	18.7	15.2	32.3	7.9	4.8	28.3
	MixKV	27.6	42.8	31.5	22.7	58.5	50.6	60.8	43.8	27.5	85.8	57.4	63.9	47.0	28.0	92.2	61.3	65.2	50.8	28.9	96.8
	TGV-KV	26.7	40.4	25.4	23.6	54.6	54.0	61.6	**46.7**	27.7	89.2	58.4	**64.1**	**49.3**	28.1	**93.9**	**61.5**	**65.2**	**51.3**	28.9	**97.2**
	CARVE	**42.6**	**51.7**	31.0	**24.6**	**70.4**	**56.3**	**62.5**	45.8	**28.0**	**90.5**	**58.5**	64.0	48.2	**28.4**	93.5	60.9	65.0	51.0	**29.3**	96.8
	CARVE-G	41.4	50.5	30.2	24.4	68.8	55.7	62.0	45.7	28.0	89.9	57.9	63.6	48.2	28.2	92.9	61.0	65.0	51.2	29.2	96.9
Qwen2.5-VL-7B	Full cache	91.2	76.7	78.2	79.4	100	91.2	76.7	78.2	79.4	100	91.2	76.7	78.2	79.4	100	91.2	76.7	78.2	79.4	100
	SnapKV	37.7	41.7	58.0	53.6	58.7	58.5	41.9	64.1	63.3	70.0	74.6	46.1	61.0	68.2	76.8	85.3	62.9	65.9	73.2	88.2
	H2O	45.2	42.4	57.9	56.5	62.0	58.4	57.2	63.0	60.7	73.5	66.9	62.3	64.9	63.5	79.1	78.5	68.5	73.2	68.9	88.8
	PrefixKV	48.5	46.3	**62.4**	56.7	65.7	64.2	60.7	64.8	61.4	77.1	72.9	65.8	65.8	65.2	82.9	82.1	70.0	68.2	70.1	89.2
	StreamingLLM	37.4	41.6	58.2	53.5	58.6	39.7	41.8	64.1	56.0	61.9	42.9	46.1	61.1	56.1	63.3	53.6	53.5	65.4	59.1	71.1
	ElasticCache	15.4	24.0	7.1	24.6	21.8	13.1	16.1	5.4	19.0	16.5	11.3	16.3	12.7	17.3	17.7	11.5	24.5	14.2	22.7	22.4
	MixKV	39.1	41.1	58.2	55.0	59.4	79.5	41.5	64.5	70.0	78.5	85.7	55.0	61.4	73.4	84.6	89.5	71.8	74.5	76.9	96.0
	TGV-KV	59.0	40.7	52.8	58.3	64.7	**85.0**	50.5	61.0	72.0	82.5	**87.9**	68.1	68.3	75.4	92.1	**89.8**	74.0	78.5	77.3	98.2
	CARVE	**76.3**	**56.9**	59.0	70.1	**80.5**	84.0	**71.6**	71.8	**75.8**	**93.1**	87.5	**74.9**	74.4	76.6	96.3	89.8	**74.9**	79.4	77.4	**98.8**
	CARVE-G	74.6	55.7	58.8	**70.1**	79.6	83.5	70.5	**72.2**	75.7	92.7	87.5	74.3	**75.5**	**76.9**	**96.5**	89.7	73.3	**80.1**	**77.5**	98.5
Qwen3-VL-8B	Full cache	95.7	82.0	85.3	83.0	100	95.7	82.0	85.3	83.0	100	95.7	82.0	85.3	83.0	100	95.7	82.0	85.3	83.0	100
	SnapKV	46.9	37.6	48.4	61.5	56.2	54.6	51.0	71.5	69.0	71.1	75.4	53.9	72.4	73.7	79.6	89.3	64.0	70.6	78.9	87.5
	H2O	50.7	37.7	48.5	61.7	57.4	63.5	58.0	72.2	67.9	75.6	71.2	62.2	74.4	70.4	80.4	82.4	70.2	76.2	74.6	87.7
	PrefixKV	47.5	45.6	48.0	59.7	58.1	62.3	54.0	65.8	63.5	71.0	70.8	58.6	67.4	65.7	75.9	80.3	65.1	73.7	68.8	83.2
	StreamingLLM	46.7	37.5	47.9	61.5	56.0	49.8	50.9	71.5	65.6	68.7	51.8	53.9	72.4	66.9	70.8	62.2	61.1	70.5	68.0	75.7
	ElasticCache	14.4	24.2	43.4	25.0	30.9	15.4	24.5	14.2	25.8	23.1	16.6	24.8	15.0	25.7	23.7	20.8	24.7	13.3	20.4	22.9
	MixKV	47.4	40.2	**67.7**	63.7	63.3	82.4	52.2	70.5	75.5	81.1	90.3	56.9	71.2	78.7	85.9	94.4	78.6	78.6	81.8	96.4
	TGV-KV	67.2	46.1	64.9	67.0	70.8	91.0	58.9	70.6	78.2	86.3	93.4	72.0	74.5	80.5	92.6	94.8	79.5	81.6	82.3	97.7
	CARVE	**76.5**	46.2	64.9	**71.1**	**74.8**	**91.6**	**65.9**	**73.6**	**79.6**	**89.8**	**93.5**	**73.8**	**78.4**	**81.2**	**94.5**	94.8	79.3	81.8	82.4	97.8
	CARVE-G	72.3	**46.2**	64.9	70.3	73.4	90.3	65.7	73.5	79.3	89.3	93.5	73.5	78.1	81.1	94.3	**94.8**	**79.6**	**82.4**	**82.5**	**98.1**

Protocol. All methods use the same total KV cache budget for each model and budget level, with eviction applied at the prefill boundary. CARVE uses unified hyperparameters across all tasks and models, and the gate is signal-driven with fixed thresholds. We verify budget parity by measuring realized KV retention (e.g., 0.0100 measured at the 1% target; Appendix B). Where reported, confidence intervals are the 95% percentile intervals of a paired bootstrap over per-sample scores, using 10,000 resamples per interval.

5 Main results

Overview. The two modes perform best at different budgets. CARVE achieves the best four-benchmark average in 7 of the 8 extreme-budget blocks and CARVE-G in the eighth, and one of the two is best in 13 of the 16 model–budget blocks (Ret. columns, Table 1). At 1%, CARVE exceeds the best baseline average in every model group, by $+10.5$, $+3.6$, $+12.1$, and $+3.4$ (e.g., 62.9 vs. 52.4 for H2O on InternVL3-8B), with the best score in 10 of 16 model–dataset cells; at 3% the gains stay positive in every group, from $+0.6$ to $+8.7$. At larger budgets, CARVE-G achieves the best average on InternVL3-8B at 10% (79.4 vs. 79.0), and the two modes split the four Qwen blocks. The exceptions are the larger LLaVA-NeXT budgets, where the text-guided baseline leads by 0.3–0.4, and 5% on InternVL3-8B, where MixKV leads by 0.6 (74.5 vs. 73.9); both are consistent with the smaller effect of the 2D structure at larger budgets (Section 6.2). The largest DocVQA gain occurs on InternVL3-8B at 1%, where CARVE scores 66.8 and retains 74.4% of the full-cache 89.8, while the strongest text-guided baseline retains 40% (35.5; a gain of $+31.3$,

Table 2: **Analysis of the allocation and the gate.** (a) Allocation ladder at the 1% budget: mean over the four core datasets (scores $\times 100$; InternVL3-8B and Qwen2.5-VL-7B). (b) CARVE-G vs. its conservative fallback on full evaluation sets, derived from Table 1: $\bar{\Delta}$ = four-benchmark mean difference; "worst" = worst single-benchmark difference.

(a) Allocation ladder (1%)

Allocation	InternVL3	Qwen2.5
Token-level top-k (1D)	41.0	52.7
Uniform-2D	42.7	50.8
Random-2D	47.3	51.1
CARVE (w/o switch)	**59.6**	**65.9**

(b) CARVE-G vs. its fallback (full sets)

	LLaVA		InternVL3		Qwen2.5		Qwen3	
Budget	$\bar{\Delta}$	worst	$\bar{\Delta}$	worst	$\bar{\Delta}$	worst	$\bar{\Delta}$	worst
10%	−0.2	−0.6	+0.5	−0.0	+0.3	−0.7	+0.3	+0.0
5%	−0.5	−1.1	+1.1	−1.6	+3.6	−0.5	+1.4	+0.1
3%	+0.3	−1.0	+14.4	+7.7	+8.4	−1.4	+2.6	−0.6
1%	+7.6	+0.8	+19.1	+9.3	+12.1	+6.0	+2.2	+0.0

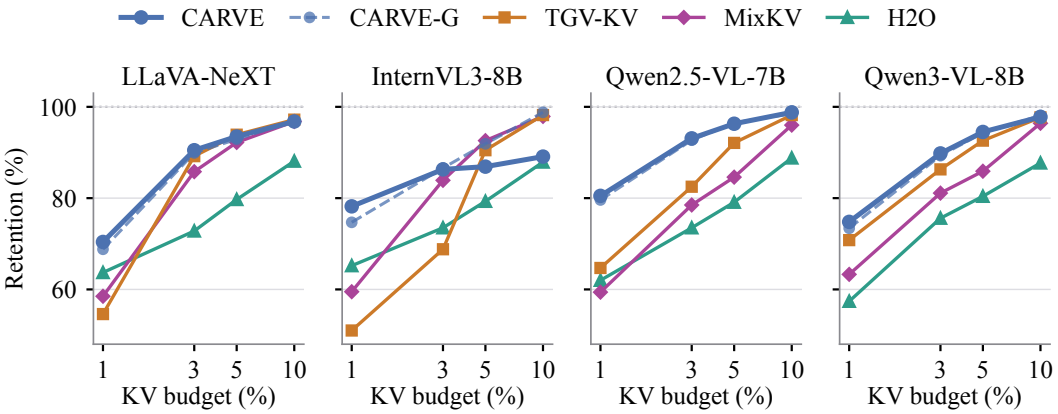

Figure 2: **Graceful degradation vs. cliff.** Four-benchmark retention (% of the full-cache average; Ret. columns of Table 1) as the KV budget shrinks (log axis), one panel per model. Markers are measured points; lines only interpolate. At 1%, CARVE retains 70–81% of the full-cache average, the strongest text-guided baseline 51–71%, and H2O 57–65%. At the ample budgets, the gated mode and the text-guided baseline close the gap, consistent with the regime split of Section 5. ElasticCache (Ret. 2–31) is omitted for scale; all rows appear in Table 1.

95% CI $[+30.1, +32.5]$, and $+26.1$ over H2O at 40.7). Since CARVE uses the same base relevance signal as TGV-KV under the same total budget, this margin mainly reflects where the budget is allocated. The attention-score baselines lose 26–54 points on DocVQA at 1–3% under matched budgets (e.g., SnapKV scores 37.7 vs. the full-cache 91.2 on Qwen2.5-VL-7B at 1%).

Graceful degradation. The baselines degrade sharply at the extreme budgets (below 5% on InternVL3-8B and LLaVA-NeXT, and below 3% on the Qwen models), whereas CARVE degrades gracefully (Figure 2). On InternVL3-8B DocVQA, CARVE loses 23.0 from the full cache at 1%, while TGV-KV loses 54.3 and SnapKV 53.4. On Qwen2.5-VL-7B, moving from 5% to 1% costs TGV-KV 28.9 (87.9 $\rightarrow$ 59.0) but only 11.3 for CARVE (87.5 $\rightarrow$ 76.3). The same ordering holds on LLaVA-NeXT, where the 1% four-benchmark average of CARVE (37.5) retains 70% of the matched full-cache average (53.2), compared with 54.6–63.7% for the other baselines (ElasticCache 3.1%; Ret. columns, Table 1).

Cell-level comparisons. H2O remains ahead on InternVL3 ChartQA (69.2 vs. 64.5 at 1%). There the signal switch of Section 3.5 improves the text-guided base by $+22.1$ and closes the gap to within 4.6 of H2O. On Qwen3-VL-8B, TextVQA and ChartQA at 1% essentially match TGV-KV ($+0.1$ and $+0.0$), while DocVQA and InfoVQA gain $+9.4$ and $+4.1$; the per-head baseline leads on ChartQA at the 1% budget.

6 ANALYSIS

6.1 THE ALLOCATION LADDER

Table 2a isolates where the budget is allocated while holding the scorer, the selection rule, and the total budget fixed. We start from the 1D token-level top-k of the base scorer on InternVL3-8B at 1% (four-dataset mean 41.0). Merely imposing 2D structure with query-agnostic budgets already helps: Uniform-2D reaches 42.7 and Random-2D 47.3. These gains do not generalize to Qwen2.5-VL-7B, where the same query-agnostic allocations fall *below* the 1D baseline (50.8 and 51.1 against 52.7). The query-conditioned committed allocation yields substantial gains on both models, reaching 59.6 on InternVL3-8B ($+12.2$ over the best query-agnostic allocation and $+18.6$ over 1D) and 65.9 on Qwen2.5-VL-7B ($+13.2$ over 1D). The signal switch of Section 3.5 adds $+3.4$ more on InternVL3-8B, reaching the four-benchmark average of 62.9 for CARVE (Table 1 cells). The geometry of the funded unit matters beyond the budget split. We hold B_v and the number of cells fixed and shuffle which tokens belong to each cell, so that a funded cell becomes a scattered set instead of a contiguous 2D block. This costs 10.1 and 2.4 points on InternVL3-8B DocVQA and ChartQA at 1%, and 3.1 and 3.8 at 3% (full evaluation sets; region path with the signal switch off).

At a 1% budget, the per-region return is threshold-like, since a region needs a sufficient number of tokens to be legible. Therefore, blind concentration can outperform blind spreading, and the query-conditioning then aims the concentration at the right regions. The ladder attributes the gains of CARVE to the query-conditioned allocation rather than to any single scoring choice.

6.2 PLACEMENT QUALITY UNDER MATCHED BUDGETS

We compare CARVE with Uniform-2D under identical per-sample budgets using 60K matched samples from two models and four benchmarks, with one to three budgets per model–task pair. We group samples by region-score entropy, averaging per-layer normalized entropy for each sample (Figure 3). In the pooled results, the advantage is positive in every quintile ($+1.7$ to $+3.6$). When restricted to the 1% budget, it grows monotonically with the entropy, from $+0.6$ in the sharpest quintile to $+8.9$ in the flattest; on DocVQA, the advantage is $+3.5$ to $+4.8$ throughout. This direction may look opposite to the reliability ordering of Proposition 1, but the two are consistent because they answer different questions. Against blind spreading, the sharp samples are the easy case, since almost any policy retains the peak region, so the commitment adds little. On dispersed samples, uniform allocation starves every region below the number of tokens needed for legibility, and committing to a few regions, even from an imperfect ranking, is what preserves usable evidence. The reliability ordering instead governs when the commitment should beat the token-level top-k of the base scorer, which is exactly the regime that the selection-time switches of Section 3.5 target. The per-task slices follow the same pattern. On LLaVA-NeXT DocVQA at 1%, the advantage grows from $+4.0$ in Q1 to $+9.8$ in Q5, and TextVQA is positive in every quintile ($+2.1$ to $+3.9$; Appendix C).

Reallocating by signal informativeness. When the text-grounded scores are flat, the signal switch of Section 3.5 can move the selection to the accumulated prefill attention. Summed over the four full benchmarks on InternVL3-8B at 1%, the region allocation with the text signal accounts for $+74.4$ of the $+87.8$ improvement of CARVE over the text-guided baseline ($\approx 85\%$), and the switch contributes the remaining $+13.4$, with $+12.8$ of it on chart-style inputs ($+5.1$ at 3%; $+1.3$ on LLaVA-NeXT ChartQA at 1%). This reallocation follows from the choice of the signal and requires no modality parameter. Table 3 shows that neither selection-time constant is the per-benchmark maximizer.

6.3 THE GATE: REGRESSIONS AND GAINS

Table 2b, derived from Table 1, compares CARVE-G with its conservative token-level fallback on the full evaluation sets. Across all sixteen model$\times$budget rows, the worst single-benchmark difference is -1.6 (InfoVQA, InternVL3, 5%; 95% CI $[-2.8, -0.3]$). The mean improvement grows with the severity, reaching $+7.6$ (LLaVA-NeXT), $+19.1$ (InternVL3), $+12.1$ (Qwen2.5-VL), and $+2.2$ (Qwen3-VL) at 1%. The fallback is particularly useful on InternVL3 ChartQA at larger budgets, where it improves over CARVE by $+26.2$ at 10% and $+8.5$ at 5%. The gating can also reduce the gains, most noticeably on Qwen3-VL-8B DocVQA at 1%, where CARVE-G scores 72.3 versus 76.5 for CARVE but still remains $+5.2$ above the base scorer. This result illustrates the cost of the

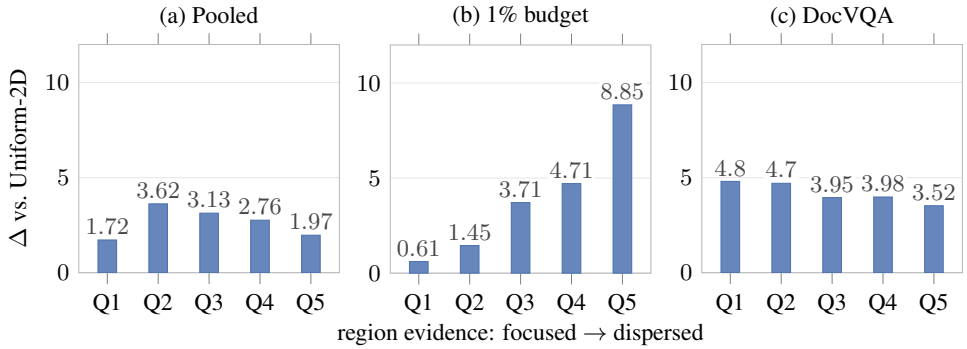

Figure 3: **Committed allocation vs. uniform allocation under matched budgets**, with samples binned by region-evidence concentration (entropy quintiles of the region-score distribution: Q1 = most focused 20%, Q5 = most dispersed 20%); 60,148 matched samples. (a) Pooled over models, tasks, and budgets; (b) 1% slice; (c) DocVQA pooled. The advantage is positive in every bin, and at 1% it grows with entropy. Query-ranked commitment matters most in the bins where the evidence is dispersed and the budgets are tightest.

Table 3: Sensitivity around the shipped constants (bold), InternVL3-8B at 1%; full evaluation sets.

Prompt-length threshold	25	30	**35**	40	45	50
ChartQA	54.8	59.3	**64.5**	66.6	67.0	66.9
DocVQA	48.0	61.4	**66.8**	68.5	69.2	69.4

Funded fraction	0.1	**0.2**	0.3	0.4
DocVQA	63.8	**66.8**	67.5 (best)	67.0

conservative selection. Per task on LLaVA-NeXT at 1%, CARVE-G gains +14.7/+10.1/+4.7/+0.8 (Doc/Text/Chart/Info), all four positive.

6.4 Efficiency

Decode latency is similar across methods: TGV-KV, SnapKV, H2O, StreamingLLM, CARVE, and CARVE-G all decode at 25–26 ms/token on InternVL3-8B at the 1% budget. Realized retention of the prefill context matches the configured budget (measured 0.0100 at the 1% target; Section 4). CARVE adds a host-side routing step during prefill; on InternVL3-8B it costs 30 ms per sample for CARVE (7% of the prefill forward pass) and 60 ms for CARVE-G (14%).

7 Conclusion and Limitations

Extreme KV eviction of visual contexts is an allocation problem. Treating the image as a 1D token stream discards the geometry that makes visual evidence usable, and preserving this structure is especially important at 1–3% budgets. We proposed CARVE, which commits the budget to a small set of query-ranked 2D regions and selects tokens within them. Selection-time diagnostics decide when this commitment is appropriate, and a signal-driven gate reverts to the conservative eviction when the region evidence is unreliable. Under matched total budgets, CARVE achieves gains up to +31.3 over the strongest text-guided baseline and degrades gracefully where the 1D methods break down. Its gated deployment mode preserves these gains on the full evaluation sets, with a worst single-benchmark regression of 1.6 points relative to the text-guided fallback. As visual contexts keep growing, we expect the cache layer, like the input layer before it, to become spatially structured, and CARVE is a concrete step in this direction.

Limitations. Our evaluation covers four 7B–8B VLMs on single-image question answering with short outputs. Larger models, multi-page documents, videos, ultra-high-resolution inputs, and longer generations remain to be evaluated, and host-side routing is not yet kernel-optimized.

REPRODUCIBILITY STATEMENT

All results are computed on the full evaluation sets with greedy decoding (temperature 0) under the fixed constants listed in Appendix B, which are shared across all tasks, models, and budgets. Section 4 names the models, benchmarks, budgets, and baselines. Appendix B gives the cell construction, the allocation rule with its integerization, the switches, and the gate in algorithmic form, together with the sensitivity of the constants (Table 3) and the budget-parity measurement. Every run records its full runtime configuration and code version alongside its outputs, and the numbers in the tables are read from these records without manual transcription.

USE OF LARGE LANGUAGE MODELS

Large language model assistants were used to polish the writing, to draft figure-plotting scripts, and to cross-check the numerical consistency between tables and text. The method, the experiments, and all claims are the authors'.

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 *Empirical Methods in Natural Language Processing*, pp. 4895–4901, 2023. doi: 10.18653/v1/2023.emnlp-main.298.

Kazi Hasan Ibn Arif, JinYi Yoon, Dimitrios S. Nikolopoulos, Hans Vandierendonck, Deepu John, and Bo Ji. HiRED: Attention-guided token dropping for efficient inference of high-resolution vision-language models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(2): 1773–1781, 2025. doi: 10.1609/aaai.v39i2.32171.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-VL technical report. *arXiv preprint arXiv:2511.21631*, 2025a.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-VL technical report. *arXiv preprint arXiv:2502.13923*, 2025b.

Zefan Cai, Yichi Zhang, Bofei Gao, Yuliang Liu, Yucheng Li, Tianyu Liu, Keming Lu, Wayne Xiong, Yue Dong, Junjie Hu, and Wen Xiao. PyramidKV: Dynamic KV cache compression based on pyramidal information funneling. *arXiv preprint arXiv:2406.02069*, 2024.

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*, volume 15139 of *Lecture Notes in Computer Science*, pp. 19–35. Springer, 2024. doi: 10.1007/978-3-031-73004-7_2.

Tianhao Chen, Yuheng Wu, Kelu Yao, Xiaogang Xu, Xiaobin Hu, and Dongman Lee. Last but not least: Boundary attention CalibratiON for multimodal KV cache compression. *arXiv preprint arXiv:2606.14782*, 2026a.

Xihao Chen, Yangyang Guo, and Roger Zimmermann. Make your LVLM KV cache more lightweight. *Transactions on Machine Learning Research*, 2026b.

Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. FlashAttention: Fast and memory-efficient exact attention with IO-awareness. In *Advances in Neural Information Processing Systems*, pp. 16344–16359, 2022.

Suyu Ge, Yunan Zhang, Liyuan Liu, Minjia Zhang, Jiawei Han, and Jianfeng Gao. Model tells you what to discard: Adaptive KV cache compression for LLMs. In *International Conference on Learning Representations*, 2024.

Yuhang Han, Wenzheng Yang, Yujie Chen, Xiangqi Jin, Yaojie Zhang, Siteng Huang, and Linfeng Zhang. STaR-KV: Spatio-temporal adaptive re-weighting for KV cache compression in GUI vision-language models. *arXiv preprint arXiv:2606.01790*, 2026.

Kung-Hsiang Huang, Haoyi Qiu, Yutong Dai, Caiming Xiong, and Chien-Sheng Wu. GUI-KV: Efficient GUI agents via KV cache with spatio-temporal awareness. *arXiv preprint arXiv:2510.00536*, 2025.

Yingbing Huang, Tharun Adithya Srikrishnan, Steven K. Reinhardt, and Deming Chen. KVCapsule: Efficient sequential KV cache compression for vision-language models with asymmetric redundancy. *arXiv preprint arXiv:2605.16439*, 2026.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7B. *arXiv preprint arXiv:2310.06825*, 2023.

Junwan Kim and Hyunkyung Bae. Reducing peak memory usage for modern multimodal large language model pipelines. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 28518–28526, 2026. doi: 10.18653/v1/2026.findings-acl.1422.

Minsoo Kim, Kyuhong Shim, Jungwook Choi, and Simyung Chang. InfiniPot-V: Memory-constrained KV cache compression for streaming video understanding. In *Advances in Neural Information Processing Systems*, pp. 154137–154167, 2025.

Kunxi Li, Zhonghua Jiang, Zhouzhou Shen, Zhaode Wang, Chengfei Lv, Shengyu Zhang, Fan Wu, and Fei Wu. MadaKV: Adaptive modality-perception KV cache eviction for efficient multimodal long-context inference. In *Annual Meeting of the Association for Computational Linguistics*, pp. 13306–13318, 2025. doi: 10.18653/v1/2025.acl-long.652.

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 *Advances in Neural Information Processing Systems*, pp. 22947–22970, 2024.

Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. LLaVA-NeXT: Improved reasoning, OCR, and world knowledge. `https://llava-vl.github.io/blog/2024-01-30-llava-next/`, 2024a.

Jizhihui Liu, Ruizi Han, Miao Zhang, Rui Shao, Xuebo Liu, Weili Guan, and Yaowei Wang. TGV-KV: Text-grounded KV eviction for vision-language models. In *International Conference on Machine Learning*, 2026a. arXiv:2606.03075.

Sihao Liu, YuFan Xiong, Zhonghua Jiang, Zhaode Wang, Chengfei Lv, and Shengyu Zhang. RetentiveKV: State-space memory for uncertainty-aware multimodal KV cache eviction. In *Findings of the Association for Computational Linguistics: ACL*, pp. 18717–18727, 2026b. doi: 10.18653/v1/2026.findings-acl.934.

Xuyang Liu, Xiyan Gui, Yuchao Zhang, and Linfeng Zhang. Mixing importance with diversity: Joint optimization for KV cache compression in large vision-language models. In *International Conference on Learning Representations*, 2026c.

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

Zuyan Liu, Benlin Liu, Jiahui Wang, Yuhao Dong, Guangyi Chen, Yongming Rao, Ranjay Krishna, and Jiwen Lu. Efficient inference of vision instruction-following models with elastic cache. In *European Conference on Computer Vision*, volume 15075 of *Lecture Notes in Computer Science*, pp. 54–69. Springer, 2024b. doi: 10.1007/978-3-031-72643-9_4.

Ahmed Masry, Do Xuan Long, 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*, pp. 2263–2279, 2022. doi: 10.18653/v1/2022.findings-acl.177.

Minesh Mathew, Dimosthenis Karatzas, and C. V. Jawahar. DocVQA: A dataset for VQA on document images. In *IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 2200–2209, 2021.

Minesh Mathew, Viraj Bagal, Rubèn Tito, Dimosthenis Karatzas, Ernest Valveny, and C. V. Jawahar. InfographicVQA. In *IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 1697–1706, 2022.

Reiner Pope, Sholto Douglas, Aakanksha Chowdhery, Jacob Devlin, James Bradbury, Jonathan Heek, Kefan Xiao, Shivani Agrawal, and Jeff Dean. Efficiently scaling transformer inference. In *Proceedings of Machine Learning and Systems*, 2023.

Yuzhang Shang, Mu Cai, Bingxin Xu, Yong Jae Lee, and Yan Yan. LLaVA-Prumerge: Adaptive token reduction for efficient large multimodal models. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 22857–22867. IEEE, 2025. doi: 10.1109/ICCV51701.2025.02122.

Noam Shazeer. Fast transformer decoding: One write-head is all you need. *arXiv preprint arXiv:1911.02150*, 2019.

Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards VQA models that can read. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8317–8326, 2019.

Dezhan Tu, Danylo Vashchilenko, Yuzhe Lu, and Panpan Xu. VL-Cache: Sparsity and modality-aware KV cache compression for vision-language model inference acceleration. In *International Conference on Learning Representations*, 2025.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pp. 5998–6008, 2017.

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*, pp. 4065–4078, 2024. doi: 10.18653/v1/2024.findings-emnlp.235.

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 *North American Chapter of the Association for Computational Linguistics*, pp. 2485–2497, 2025. doi: 10.18653/v1/2025.naacl-long.125.

Ao Wang, Hui Chen, Jiaxin Li, 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 *Advances in Neural Information Processing Systems*, pp. 104755–104781, 2025a.

Jiahui Wang, Zuyan Liu, Yongming Rao, and Jiwen Lu. SparseMM: Head sparsity emerges from visual concept responses in MLLMs. In *IEEE/CVF International Conference on Computer Vision*, pp. 23177–23187, 2025b. doi: 10.1109/ICCV51701.2025.02152.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. In *International Conference on Learning Representations*, 2024.

Yaoxin Yang, Peng Ye, Xudong Tan, Chongjun Tu, Maosen Zhao, Jia Hao, and Tao Chen. Revisiting multimodal KV cache compression: A frequency-domain-guided outlier-KV-aware approach. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026.

Bowen Zeng, Feiyang Ren, Jun Zhang, Xiaoling Gu, Ke Chen, Lidan Shou, and Huan Li. HybridKV: Hybrid KV cache compression for efficient multimodal large language model inference. In *Annual Meeting of the Association for Computational Linguistics*, pp. 13018–13034, 2026. doi: 10.18653/v1/2026.acl-long.594.

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

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, Zhangyang Wang, and Beidi Chen. H2O: Heavy-hitter oracle for efficient generative inference of large language models. In *Advances in Neural Information Processing Systems*, pp. 34661–34710, 2023.

Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, Zhangwei Gao, Erfei Cui, Xuehui Wang, Yue Cao, Yangzhou Liu, Xingguang Wei, Hongjie Zhang, Haomin Wang, Weiye Xu, Hao Li, Jiahao Wang, Nianchen Deng, Songze Li, Yinan He, Tan Jiang, Jiapeng Luo, Yi Wang, Conghui He, Botian Shi, Xingcheng Zhang, Wenqi Shao, Junjun He, Yingtong Xiong, Wenwen Qu, Peng Sun, Penglong Jiao, Han Lv, Lijun Wu, Kaipeng Zhang, Huipeng Deng, Jiaye Ge, Kai Chen, Limin Wang, Min Dou, Lewei Lu, Xizhou Zhu, Tong Lu, Dahua Lin, Yu Qiao, Jifeng Dai, and Wenhai Wang. InternVL3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*, 2025.

A PROOFS OF PROPOSITION 1 AND COROLLARY 1

Model. We consider $K \geq 2$ regions, one of which, denoted by A, contains the evidence needed to answer the query. A budget $B > 0$ is split into $b = (b_1, \ldots, b_K)$ with $b_r \geq 0$ and $\sum_r b_r = B$. If the evidence region receives budget b_A, the answer is recovered with probability $U(b_A)$.

Assumption 1. *$U : [0, B] \to [0, 1]$ satisfies $U(0) = 0$, is continuous on $[0, B]$, and is twice continuously differentiable on $(0, B]$ with $U' > 0$ and $U'' < 0$ there. The right derivative at zero may be finite or infinite; both cases $U'(0^+) < \infty$ and $U'(0^+) = \infty$ are allowed.*

Concavity encodes diminishing returns. Within a region, the tokens are kept in the relevance order, so the first tokens recover the most evidence and the returns saturate once the evidence is covered. The threshold behavior illustrates the shape. If the retrieval succeeds when b_A exceeds a sample-dependent threshold c, then $U(b) = \Pr[c \leq b]$ is the threshold CDF, and a nonincreasing threshold density makes this CDF concave. If, moreover, the threshold is positive almost surely and has a continuously differentiable density f with $f > 0$ and $f' < 0$ on $(0, B]$, then $U(0) = 0$, $U' = f > 0$, and $U'' = f' < 0$ there, so Assumption 1 holds in full.

The allocator observes scores $s = (s_1, \ldots, s_K)$ and splits the budget through the softmax family $b_r(\tau) = B\,\pi_r(\tau)$ of Equation 1, indexed by the temperature $\tau \in (0, \infty)$.

Proposition 1 (Optimal concentration tracks score reliability)**.** *Under Assumption 1:*

- *(i)* ***Uninformative scores ⇒ uniform.*** *If A is uniform on $\{1, \ldots, K\}$ and independent of s, then every allocation policy $b = b(s)$, a measurable function of the scores alone, satisfies $\mathbb{E}[U(b_A)] \leq U(B/K)$, with equality iff $b \equiv (B/K, \ldots, B/K)$ almost surely. In the softmax family with almost-surely distinct scores, $\mathbb{E}[U(b_A(\tau))] < U(B/K)$ for every finite τ, and $\mathbb{E}[U(b_A(\tau))] \to U(B/K)$ as $\tau \to \infty$: the uniform limit is the unique supremum.*
- *(ii)* ***Perfect scores ⇒ maximal concentration.*** *If $s_A = \mu > 0$ and $s_r = 0$ for $r \neq A$ deterministically, then $\mathbb{E}[U(b_A(\tau))] = U\big(B\,e^{\mu/\tau}/(e^{\mu/\tau} + K - 1)\big)$ is strictly decreasing in τ, and its supremum $U(B)$ is approached as $\tau \to 0^+$.*
- *(iii)* ***Two-region characterization.*** *Let $K = 2$ and let the scorer rank the evidence region first with probability $p \in [\frac{1}{2}, 1]$, with a fixed observed score gap $m > 0$. Under the softmax, the top-ranked region receives the share $q(\tau) = \sigma(m/\tau) \in (\frac{1}{2}, 1)$, where σ is the logistic function and $\tau \mapsto q$ is a strictly decreasing bijection. Writing $F(q; p) = p\,U(Bq) + (1 - p)\,U(B(1 - q))$ for $q \in [\frac{1}{2}, 1]$:*
  - *(a) F is strictly concave in q and has a unique maximizer $q^*(p)$;*
  - *(b) $q^*(\frac{1}{2}) = \frac{1}{2}$ (uniform; $\tau^* \to \infty$), and $q^*(p) > \frac{1}{2}$ for $p > \frac{1}{2}$;*
  - *(c) wherever q^* is interior it satisfies $U'(Bq^*)/U'(B(1-q^*)) = (1-p)/p$ and is strictly increasing in p; equivalently $\tau^*(p) = m/\operatorname{logit}(q^*(p))$ is strictly decreasing in p;*
  - *(d) for $p \in (\frac{1}{2}, 1)$ the maximizer is interior iff $U'(B)/U'(0^+) < (1-p)/p < 1$ (the upper inequality is exactly $p > \frac{1}{2}$; $p = \frac{1}{2}$ is the uniform boundary case of (b)). Consequently, if $U'(0^+) < \infty$ then $q^*(p) = 1$ for all $p \geq \bar{p} := U'(0^+)/(U'(0^+) + U'(B))$; if $U'(0^+) = \infty$ then q^* is interior for all $p \in (\frac{1}{2}, 1)$ and $q^*(p) \to 1$ as $p \to 1$ ($\tau^* \to 0$).*

Proof. (i). Condition on s. Since A is uniform and independent of s, and $b = b(s)$ is a function of s,

$$\mathbb{E}\big[U(b_A) \,\big|\, s\big] = \frac{1}{K}\sum_{r=1}^{K} U\big(b_r(s)\big) \;\leq\; U\Big(\frac{1}{K}\sum_{r=1}^{K} b_r(s)\Big) = U(B/K),$$

by Jensen's inequality for the strictly concave U; equality holds iff $b_1(s) = \cdots = b_K(s) = B/K$. Taking expectations over s proves the bound and the equality condition. For the softmax family at finite τ, distinct scores yield unequal shares, so the event $\{b(\tau) \neq (B/K, \ldots, B/K)\}$ has probability one and the bound is not attained. As $\tau \to \infty$, $b_r(\tau) \to B/K$ pointwise; since U is continuous and bounded on $[0, B]$, dominated convergence gives $\mathbb{E}[U(b_A(\tau))] \to U(B/K)$.

(ii). Here $b_A(\tau) = B\,e^{\mu/\tau}\big/\big(e^{\mu/\tau} + K - 1\big)$. The map $\tau \mapsto \mu/\tau$ is strictly decreasing on $(0, \infty)$, and $x \mapsto Be^x/(e^x + K - 1)$ is strictly increasing, so $b_A(\tau)$ is strictly decreasing in τ; composing

with the strictly increasing U preserves the monotone decrease. As $\tau \to 0^+$, $b_A \to B$ and $U(b_A) \to U(B)$, the supremum of U on $[0, B]$.

(iii). With two regions and gap m, the softmax gives the higher-scored region the share $\sigma(m/\tau)$; since $m > 0$, $\tau \mapsto \sigma(m/\tau)$ is a strictly decreasing bijection from $(0, \infty)$ onto $(\frac{1}{2}, 1)$. With probability p the top-ranked region is the evidence region (share q), otherwise the evidence region holds the share $1 - q$; hence the expected utility is $F(q; p)$.

(a) For $q \in (\frac{1}{2}, 1)$ both arguments Bq and $B(1-q)$ lie in $(0, B)$, where U'' exists and is negative (Assumption 1), so $\partial^2 F/\partial q^2 = B^2\big[p\,U''(Bq) + (1-p)U''(B(1-q))\big] < 0$ there; since F is continuous on $[\frac{1}{2}, 1]$ (continuity of U at 0), strict concavity extends to the closed interval, and the maximizer is therefore unique.

(b) $\partial F/\partial q = B\big[p\,U'(Bq) - (1-p)\,U'(B(1-q))\big]$. At $q = \frac{1}{2}$ this equals $B\,U'(B/2)\,(2p-1)$, which is 0 at $p = \frac{1}{2}$ (so, with strict concavity, $q^* = \frac{1}{2}$ exactly) and strictly positive for $p > \frac{1}{2}$ (so $q^* > \frac{1}{2}$). Under the bijection of (iii), $q^* = \frac{1}{2}$ corresponds to $\tau^* \to \infty$.

(c) An interior maximizer solves the first-order condition $G(q, p) := p\,U'(Bq) - (1-p)\,U'(B(1-q)) = 0$, i.e. $U'(Bq)/U'(B(1-q)) = (1-p)/p$. Implicit differentiation gives

$$\frac{dq^*}{dp} = -\frac{\partial G/\partial p}{\partial G/\partial q} = \frac{U'(Bq^*) + U'(B(1-q^*))}{-B\big[p\,U''(Bq^*) + (1-p)\,U''(B(1-q^*))\big]} > 0,$$

since the numerator is positive (U' positive) and the bracket in the denominator is negative (U'' negative). Because logit is strictly increasing and $\tau = m/\operatorname{logit}(q)$, τ^* is strictly decreasing in p.

(d) The left side of the first-order condition, $R(q) := U'(Bq)/U'(B(1-q))$, is strictly decreasing on $(\frac{1}{2}, 1)$, from $R(\frac{1}{2}) = 1$ to $R(1^-) = U'(B)/U'(0^+)$. Fix $p \in (\frac{1}{2}, 1)$, so that $(1-p)/p \in (0, 1)$; a solution of $R(q) = (1-p)/p$ interior to $(\frac{1}{2}, 1)$ exists iff $(1-p)/p$ lies in the range of R, i.e. iff $U'(B)/U'(0^+) < (1-p)/p < 1$, and the upper inequality holds automatically ($p > \frac{1}{2}$). The case $p = \frac{1}{2}$ is the boundary solution $q^* = \frac{1}{2}$ of (b). If $U'(0^+) < \infty$, the lower inequality fails exactly when $p \geq \bar{p} = U'(0^+)/(U'(0^+) + U'(B))$, where the maximizer sits at the boundary $q^* = 1$: the derivative $\partial F/\partial q$ is strictly positive on $[\frac{1}{2}, 1)$, and at the threshold $p = \bar{p}$ it vanishes only at the endpoint $q = 1$. If $U'(0^+) = \infty$ then $R(1^-) = 0$ and an interior solution exists for every $p \in (\frac{1}{2}, 1)$; as $p \to 1$, $(1-p)/p \to 0$ forces $R(q^*) \to 0$, hence $q^* \to 1$. □

Corollary 1 (Closed form; concentration grows as the budget shrinks)**.** *Let $U(x) = 1 - e^{-x/c}$ with evidence scale $c > 0$. Then in Proposition 1(iii),*

$$q^*(p, B) \;=\; \min\Big\{1,\; \tfrac{1}{2} + \tfrac{c}{2B}\,\ln\tfrac{p}{1-p}\Big\},$$

and the boundary is reached at $\bar{p} = \sigma(B/c)$. In particular, at fixed reliability $p > \frac{1}{2}$, the optimal concentration q^* is nonincreasing in B: strictly decreasing on the interior (uncapped) regime $p < \sigma(B/c)$, and identically 1 at the cap. The tighter the budget relative to the evidence scale, the more the allocation should concentrate, up to full concentration.

Proof. $U'(x) = \frac{1}{c}e^{-x/c}$, so the first-order condition $U'(Bq)/U'(B(1-q)) = (1-p)/p$ reads $e^{-(B/c)(2q-1)} = (1-p)/p$, i.e. $q = \frac{1}{2} + \frac{c}{2B}\ln\frac{p}{1-p}$, capped at 1 where the right-hand side exceeds it; the cap binds iff $\ln\frac{p}{1-p} \geq B/c$, i.e. $p \geq \sigma(B/c)$, matching $\bar{p} = U'(0^+)/(U'(0^+) + U'(B)) = 1/(1 + e^{-B/c})$. On the uncapped regime the interior formula is strictly decreasing in B; once the cap binds, $q^* \equiv 1$. Hence q^* is nonincreasing in B overall. □

Remark 1 (Scope)**.** *Parts (i) and (ii) hold for arbitrary K and pin down both ends of the temperature scale. Part (iii) proves the full monotone characterization in the two-region case. For general K with intermediate noise, we do not attempt a proof; instead, the controlled measurement of Section 6.2, which stratifies 60,148 matched samples by the score sharpness, serves as its empirical counterpart. One further gap remains. The theorem links the concentration to the rank-reliability p of a two-region model, whereas the implemented system reads the entropy, Gini, and top-mass statistics of the observed region scores as empirical proxies for that reliability in its selection-time diagnostics, and we claim no formal link between the two. The gate of Section 3.6 implements the low-reliability end*

of the trade-off. When the score statistics indicate that the region evidence is unreliable, the allocator declines to commit and reverts to the conservative baseline behavior. The budget dependence of Corollary 1 matches the empirical pattern that the gains of the committed allocation grow with compression severity (Sections 5 and 6.2).

B IMPLEMENTATION AND PROTOCOL DETAILS

This appendix specifies the implementation. We state the structural constants below, and one fixed setting is used for every experiment in this paper. Algorithm 1 summarizes the per-layer, per-sample selection, and Algorithm 2 the gate decision.

Cell construction. Each image contributes $\approx R$ cells ($R = 128$; the factorization below yields $rc \geq R$). The native layout of the model exposes a token grid of h rows $\times$ w columns (anyres tile grid for LLaVA-NeXT, dynamic tiling for InternVL3, and native patch grids for the Qwen models), and we add no learned components. We factorize the grid aspect-adaptively into $r \times c$ cells. We set $r = \mathrm{clip}\big(\mathrm{round}\sqrt{R\,h/w},\, 1,\, \min(h, R)\big)$ and $c = \mathrm{clip}(\lceil R/r \rceil,\, 1,\, w)$, and then grow r and c alternately (smaller side first) until $rc \geq R$ or the grid is exhausted. Cell (i, j) contains the token rows $\lfloor ih/r \rfloor$ through $\lfloor (i{+}1)h/r \rfloor - 1$ and the columns $\lfloor jw/c \rfloor$ through $\lfloor (j{+}1)w/c \rfloor - 1$, which is a contiguous rectangle of the input. Inputs with several grids (multi-tile documents) split the R cells across the grids in proportion to the token count, with at least one cell per grid. If the layout metadata is unavailable, the cells fall back to R equal contiguous spans of the flattened visual sequence.

Region scoring. The score of cell r pools the base token relevances by the maximum, $s_r = \max_{i \in \mathcal{C}_r} \alpha_i$.

The implemented allocation rule. Cell scores are z-normalized (zero-variance scores are centered only, which the softmax maps to the uniform branch) and passed through a softmax, giving $\pi = \mathrm{softmax}(z)$ with normalized entropy $\tilde{H} = \mathrm{H}(\pi)/\ln R' \in [0, 1]$, where R' is the realized cell count rc. The diagnostic distribution of Equation 2,

$$p \;=\; \lambda\, u + (1 - \lambda)\, \pi, \qquad \lambda = \tilde{H},$$

with u uniform over cells, is computed per layer. The funded set of the region path depends only on the *ranking* of π, which the interpolation preserves. The distribution p and its statistics serve the selection-time decisions. The normalized entropy $\tilde{H}$ drives the signal switch, and the entropy, Gini, and top-k-mass statistics of p drive the confidence gate below, whose region mode also weights the in-region ranking by the cell probability p_r.

Signal switch. Before any funded-region allocation, the same cells are scored a second time by the accumulated online-prefill attention and passed through the identical pooling–normalization recipe, which gives an attention-side normalized entropy $\tilde{H}_{\mathrm{att}}$ alongside the text-side $\tilde{H}$ above. The entropy gap $\tilde{H} - \tilde{H}_{\mathrm{att}}$ is mapped through a fixed linear ramp, shared by every task, model, and budget, to a switch weight $w \in [0, 1]$ (Section 3.5). If $w > 0.5$, the selection of the layer is attention-based token-level top-k at the same budget, replicating the H2O selection semantics (same protected prefix and same budget accounting), and the region pipeline is bypassed entirely for that layer, including the conservative switches below. Otherwise, the text-grounded region path proceeds, and both a below-threshold gap ($w = 0$) and the mixed band with text majority ($0 < w \leq 0.5$) stay on the text path.

Budget split, integerization, and selection. Of the per-layer budget B (Section 3.1), we reserve a share $f = 0.75$ for the tokens inside the funded cells, $B_v = \mathrm{round}(fB)$. CARVE funds the $\mathrm{round}(0.2B)$ cells at the top of the region ranking (at least one cell, capped at 128 cells) and fills the reserved share with the highest-relevance tokens inside them. The query-agnostic ablations (Uniform-2D and Random-2D) instead convert their allocation distribution into per-cell integer quotas by the largest remainder, $b_r = \min(\lfloor p_r B_v \rfloor, |\mathcal{C}_r|)$. The remaining $B_v - \sum_r b_r$ slots are assigned one at a time to the cell with the largest fractional part $p_r B_v - \lfloor p_r B_v \rfloor$ that still has capacity, with at most one extra slot per cell and ties to the lowest cell index. The top-b_r tokens per cell by relevance are kept. In both cases, the remaining $B - B_v$ slots are filled in global relevance

Algorithm 1 CARVE per-layer selection (one sample, one layer)

Require: token relevances α (base scorer), accumulated-attention relevances, visual span, layout grids, per-layer budget B

partition visual tokens into $\approx R$ layout-aligned cells $\{\mathcal{C}_r\}$, realized count R' ▷ cell construction, R=128
$s_r \leftarrow \max_{i \in \mathcal{C}_r} \alpha_i$; $z \leftarrow \mathrm{zscore}(s)$; $\pi \leftarrow \mathrm{softmax}(z)$
$\tilde{H} \leftarrow \mathrm{H}(\pi)/\ln R'$; $p \leftarrow \tilde{H}\, u + (1 - \tilde{H})\, \pi$ ▷ diagnostic distribution, Eq. 2
score the cells by accumulated attention through the same max-pool/z-score/softmax recipe; let $\tilde{H}_{\mathrm{att}}$ be its normalized entropy
$w \leftarrow$ fixed linear ramp of the gap $\tilde{H} - \tilde{H}_{\mathrm{att}}$ ▷ signal switch
if $w > 0.5$ **then** ▷ full switch: attention-guided selection; all fallbacks bypassed
    return attention-based token-level top-k at budget B
end if
if micro-budget or prompt-length switch triggers **then**
    return exact base-scorer selection at budget B
end if
$B_v \leftarrow \mathrm{round}(0.75\, B)$; fund the top $\mathrm{round}(0.2B)$ cells of the region ranking (quotas b_r by largest-remainder integerization for the query-agnostic ablations) ▷ in CARVE-G, Algorithm 2 is consulted here
keep the highest-relevance tokens inside funded cells (B_v total; any shortfall falls through to the global fill)
fill the remaining slots by global relevance rank (text and visual) so that exactly B entries are kept

Algorithm 2 CARVE-G gate decision (one sample, one layer)

ramp gates: g_B (budget), g_H (entropy), g_G (Gini), g_m (top-5 mass)
$g \leftarrow g_B \cdot \mathrm{mean}(g_H, g_G, g_m)$
if g clears the fixed activation threshold **then** continue Algorithm 1 from the budget-split step, ranking in-region candidates by the hybrid score $(1 - \omega)\bar{\alpha}_i + \omega\, \eta_i$
else exact base-scorer selection at the same budget
end if

order over all unclaimed candidates, including text and visual tokens, so text tokens are retained under the same total KV budget.

Conservative switches. Two signal-driven switches revert a sample to the token-level top-k on the base scores at the same budget before any region allocation is attempted. (i) *Micro-budget switch*: the layers whose keep budget is at most 16 tokens form the micro regime. Per sample, we compare the median across the micro layers of the maximum cell-score z-score to a fixed global threshold. When one cell towers over all the others at a budget of a dozen tokens, the global top-k already retains it and the region quotas add only risk, so the sample reverts. (ii) *Prompt-length switch*: the region routing engages only when the post-image prompt is short, at most 35 text tokens. On longer prompts, the sample reverts to the exact base selection.

The gate (CARVE-G). The gate maps four statistics through fixed smooth ramps. The first is a budget gate g_B that activates only when the budget is restrictive, and the other three are evidence gates g_H, g_G, and g_m that rise with the sharpness of the diagnostic distribution p of Equation 2, measured by its entropy, Gini coefficient, and top-5 mass. The gate value is the budget gate multiplied by the mean of the three evidence gates, $g = g_B \cdot \mathrm{mean}(g_H, g_G, g_m)$. It is evaluated per layer using that layer's region statistics and visual retention. If g exceeds the activation threshold, the layer uses the region mode with blend weight ω. Otherwise, it uses the token-level top-k of the base scorer at the same budget. In the region mode, the in-region candidates are ranked by the hybrid score $(1 - \omega)\bar{\alpha}_i + \omega\, \eta_i$, where $\bar{\alpha}$ is the rank-normalized token relevance and η distributes each cell's probability p_r over its tokens in proportion to the normalized in-cell relevance. Visual tokens outside every cell keep their base score.

Unified hyperparameters. The structural constants are as follows: $R = 128$ cells with max pooling, a funded set of $\mathrm{round}(0.2B)$ cells with committed share $f = 0.75$, and two conservative reversion rules. The reversion rules hand a sample back to the token-level selection at budgets of at most 16 retained tokens (too few to give any funded region a viable number of tokens; z-score threshold 8.0) and for prompts longer than 35 text tokens. The detection constants are as follows: the signal-switch entropy-gap ramp $(0.30, 0.45)$; the gate smoothstep knots $(0.02, 0.08)$ for the budget gate, $(0.45, 0.85)$ for the entropy, and $(0.25, 0.65)$ for the Gini and top-5-mass gates; the gate activation threshold 0.65; and the blend weight $\omega = 0.75$. Every constant is fixed once and shared across all tasks, models, and budgets. The detection constants were set from 128-sample development probes (InternVL3-8B ChartQA and DocVQA) and were never revisited on the full sets. Table 3 reports the sensitivity around the shipped values, which are not the per-benchmark maximizers. The full-set CARVE-G results appear in Table 1 for all four models.

Budget parity. We measure realized retention of the prefill context rather than assuming it. On InternVL3-8B with TGV-KV, the measured retention is 0.100, 0.050, 0.030, and 0.010 at the 10%, 5%, 3%, and 1% targets, respectively, and all methods are matched on the same configured budget.

Gate-vs-fallback matrix. Table 2b is derived from Table 1. For each model, budget, and core benchmark (full evaluation sets), the entry compares the CARVE-G row with the conservative token-level mode, i.e., the fallback target of the gate, whose selection coincides with the strongest text-guided baseline. Therefore, $\bar{\Delta}$ is the mean difference between the CARVE-G and text-guided-baseline rows of Table 1. It measures the value that the gate adds by intervening and does not compare against CARVE. $\bar{\Delta}$ averages the four benchmark differences, and "worst" is the smallest single-benchmark difference in the row. All differences are computed from the unrounded values behind the printed cells of Table 1, so every input can be derived from the printed table to within rounding.

Full-cache references. Each model's full-cache reference is measured once, on the same stack as its compressed runs, and shared across all budget groups.

Coinciding baseline selections. At the 1% budget on InternVL3-8B, the SnapKV and StreamingLLM selections coincide on all four core benchmarks (shared cells marked †), and the mechanism is a regime property. SnapKV always keeps its 64-token observation window, and at 1% the per-layer selection quota falls below that window, so its kept set provably reduces to sinks-plus-recency, exactly StreamingLLM's selection. At extreme budgets the observation-window heuristic exhausts the entire budget; the two methods separate only once the budget clears the window (at 10% they differ fully, 81.9 vs. 58.8 DocVQA).

ElasticCache adaptation. We verified our ElasticCache adaptation line-by-line against the reference implementation. Its sharp drop in Table 1 is intrinsic to the method. The cross-position K-mean merging of post-RoPE keys is incompatible with the budget-matched eviction regime, and its scores fall below every budget-matched baseline in every model–budget group where it is evaluated. On LLaVA-NeXT, the scores decrease monotonically as the budget tightens ($32.3 \to 20.7 \to 12.3 \to 2.5$ on TextVQA), which is the signature of a systematic incompatibility rather than run-to-run noise; on InternVL3-8B, the decay is the same ($30.3 \to 9.3 \to 2.0 \to 1.3$ on TextVQA), with every extreme-budget cell in $[0.5, 3.6]$. This behavior is corroborated independently. Liu et al. (2026a) likewise report ElasticCache as the weakest baseline on vision-dominant tasks even in an easier regime (1K-token visual contexts, post-prefill eviction), e.g., 3.9 on LLaVA-1.5 ChartQA and 19.9 on LLaVA-NeXT DocVQA at a 5% budget, against 13.8 and 58.5 for their own method.

C PLACEMENT-QUALITY SLICES

Table 4 lists the per-slice values of the matched-budget comparison of Section 6.2 (CARVE − Uniform-2D, in percentage points, by the region-score-entropy quintile).

The pooled and 1% rows are the two panels of Figure 3. The per-task rows follow the same pattern. TextVQA is positive in every quintile ($+2.1$ to $+3.9$) and peaks in the most dispersed one. On LLaVA-NeXT DocVQA at 1%, the advantage ranges from $+4.0$ to $+9.8$ and is again largest in Q5, as in the pooled 1% slice. On LLaVA-NeXT ChartQA at 3%, it is small but positive ($+1.2$ to $+2.8$).

InternVL3 ChartQA at 10% is the only slice where committed allocation loses to uniform allocation, by 1.2 to 5.6 points in every quintile. At this budget, uniform allocation already keeps enough tokens in each region, and chart evidence is spread across the plot, the axes, and the legend. This is also the setting in which the gate of Section 3.6 gains the most over CARVE (+26.2 at 10%; Section 6.3).

Table 4: Matched-budget advantage of committed over uniform allocation by entropy quintile (Q1 sharpest). The ChartQA slices are available at one budget per model.

Slice	Q1	Q2	Q3	Q4	Q5
Pooled (all models/tasks/budgets)	+1.72	+3.62	+3.13	+2.76	+1.97
1% budget	+0.61	+1.45	+3.71	+4.71	+8.85
DocVQA (pooled)	+4.80	+4.70	+3.95	+3.98	+3.52
TextVQA (pooled)	+2.16	+2.76	+2.46	+2.10	+3.86
LLaVA-NeXT × DocVQA × 1%	+3.98	+6.13	+4.68	+6.88	+9.77
LLaVA-NeXT × ChartQA × 3%	+2.00	+2.20	+2.40	+2.80	+1.20
InternVL3 × ChartQA × 10%	−1.20	−4.80	−5.60	−4.80	−5.60