MM-ShiftKV: Decode-Aware Prefill-Stage KV Selection for Multimodal Large Language Models

Anonymous ACL submission

Abstract

Key-Value (KV) caching is essential for efficient inference in multimodal large language models (MLLMs), yet its memory footprint grows linearly with context length and becomes a major bottleneck due to the large number of visual tokens. Recent prefill-stage KV selection methods estimate KV importance from prefilling statistics, implicitly assuming that prefilling-time queries are representative of those encountered during decoding.

We show that this assumption breaks down in multimodal inference, where decoding-time queries exhibit substantially larger variance than prefilling-stage representations, leading to unstable KV importance estimation under tight cache budgets. As a result, small ranking errors can disproportionately discard semantically critical visual tokens and degrade grounding and reasoning performance. We propose **MM-ShiftKV**, a training-free, decode-aware and strictly prefill-only KV selection method. MM-ShiftKV approximates decoding-time query behavior during prefilling by constructing variance-expanded *query proxies* and estimates prompt KV importance based on their aggregated attention mass. Experiments on multimodal benchmarks demonstrate that MM-ShiftKV consistently outperforms existing methods under strict KV-cache budgets.[1]

1 Introduction

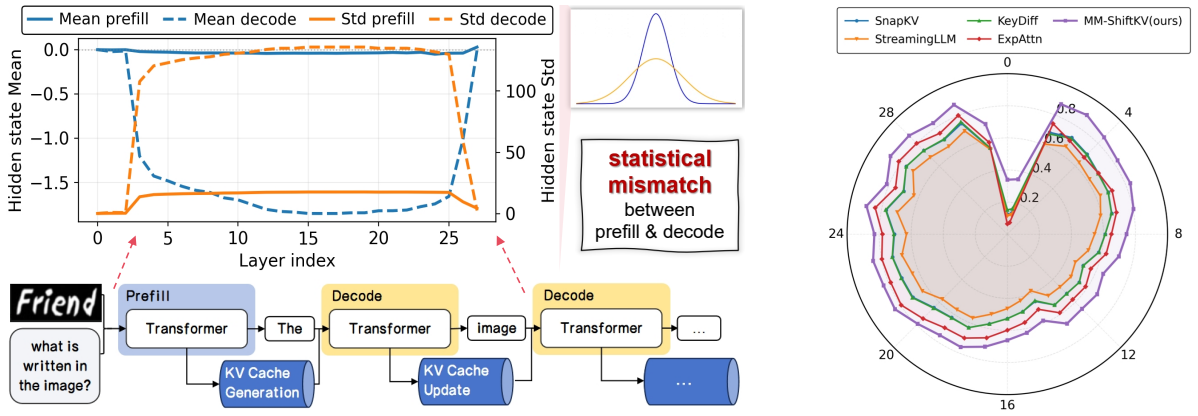
Multimodal large language models (MLLMs) extend text-only language models with the ability to generate language grounded in visual inputs, enabling applications such as optical character recognition (OCR), document understanding, and visual question answering (Li et al., 2024a; Bai et al., 2025). During inference, these models process short textual prompts together with high-resolution visual inputs, which are encoded during prefilling into long multimodal sequences dominated by visual tokens (Arif et al., 2025), followed by autoregressive decoding to generate textual outputs. Efficient decoding relies on *Key–Value (KV) caching*, whose memory footprint grows linearly with the encoded sequence length, making KV cache size and access cost a primary bottleneck for memory consumption and decoding efficiency.

To mitigate this bottleneck, recent work has proposed *prefill-stage KV cache selection*, which retains a subset of KV states after prefilling and reuses them during decoding (Xiao et al., 2023; Li et al., 2024b; Devoto et al., 2025; Park et al., 2025). Compared to decoding-time cache eviction or adaptive cache compression (Xiao et al., 2024), these prefill-only approaches are attractive because they are training-free (Li et al., 2024b), introduce no decoding-time intervention, and remain compatible with advanced attention kernels such as FlashAttention (Dao, 2023). Most methods estimate KV importance from statistics observed during prefilling, implicitly assuming that prefill-stage representations and attention behavior are representative of those encountered during decoding.

This implicit assumption becomes fragile in multimodal inference due to the heterogeneity of multimodal inputs. A large number of visually redundant tokens coexist with a small subset of semantically critical tokens (Tao et al., 2025; Chen et al., 2025), such that small errors in KV ranking may disproportionately remove critical representations. As a result, existing prefill-stage KV selection methods often lead to degraded language grounding, unstable reasoning, and significant performance drops on multimodal tasks (Devoto et al., 2025; Li et al., 2024b; Park et al., 2025; Devoto et al., 2024).

At a more fundamental level, prefilling and decoding correspond to distinct functional stages of multimodal inference. Prefilling primarily empha-

[1] Anonymous code and scripts for reproducing the experiments are available at https://anonymous.4open.science/r/mm-shiftkv.



(a) Feature mean/std (OCRBench)

(b) Attention mass coverage

Figure 1: Prefill–decode statistics and decode-time attention coverage. **(a)** Layer-wise mean and variance of hidden-state representations during prefill and decoding on OCRBench. **(b)** Attention mass coverage (retained prompt), measured as the fraction of decode-time attention probability mass assigned to prompt KV tokens retained after prefilling.

sizes visual perception and cross-modal alignment, whereas decoding shifts toward language generation and reasoning conditioned on previously generated tokens. Although the two stages share identical model parameters, they may induce different distributions of hidden states and attention queries, a phenomenon that we systematically analyze in Section 2. Consequently, importance estimates derived solely from prefilling-stage statistics can be misaligned with decoding-time behavior, leading to unreliable KV selection under strict cache budgets.

In this work, we propose **MM-ShiftKV**, a training-free and strictly prefill-only KV selection framework for multimodal inference. Our core idea is to make prefill-stage KV selection explicitly *decode-aware* by calibrating importance estimates to reflect the distributional properties of decoding-time queries, rather than relying solely on prefill-stage statistics. Decode-aware here does not imply performing KV eviction or re-ranking during decoding, but instead adjusts prefill-based query proxies to better approximate decoding-time behavior. Concretely, MM-ShiftKV performs one-shot KV selection at the end of prefilling by sampling variance-expanded, decode-aligned query proxies from global prefilling statistics and estimating KV importance based on aggregated attention mass. The resulting compact prompt KV cache is reused unchanged during decoding, enabling efficient and robust multimodal inference under strict budgets.

Contributions.

- We identify a persistent prefill–decode *scale mismatch* in multimodal inference, where decoding-time *hidden-state representations* exhibit substantially larger variance than those observed during prefilling.
- We show that this mismatch causes prefill-based query proxies to be under-scaled, leading to distorted query–key relevance estimation and unstable KV selection under strict KV-cache budgets.
- We propose **MM-ShiftKV**, a training-free and strictly *prefill-only* KV selection framework that constructs variance-expanded, decode-aligned *query proxies* to estimate prompt KV importance.
- We demonstrate improved accuracy–memory–latency trade-offs on representative OCR, grounding, and long-context VQA benchmarks under tight KV-cache constraints.

2 Observation: Prefill–Decode Scale Mismatch in Multimodal Inference

.

Recent work has shown that activation statistics in large language models exhibit structured properties that can be exploited in a training-free manner (Liu et al., 2024a). Inspired by the properties, most *prefill-stage* KV selection methods assume that statistics observed during prefilling remain representative of KV usage during decoding.

As discussed in Section 1, this assumption is critical for prefill-stage KV selection, yet it has not been carefully examined in multimodal inference, and we provide detailed theoretical proofs in Appendix A. In this section, we show that the assumption is empirically violated and character-

ize a consistent *prefill-decode statistical mismatch* on **OCRBench** (Liu et al., 2023) and **Qwen2.5-VL-Instruct**. Further results are provided in the Appendix C.

Specifically, we observe that hidden-state representations during prefilling and decoding differ substantially in their statistical *scale*. Figure 1a reports the layer-wise mean and standard deviation of hidden features for **Qwen2.5-VL-Instruct** evaluated on **OCRBench**. Although prefilling and decoding share identical model parameters, decoding-stage representations exhibit consistently larger variance across layers, while mean shifts remain relatively moderate. This indicates that prefilling-stage statistics systematically underestimate the scale of decoding-time representations, a phenomenon that lacks dedicated research in prior work on prefill-stage KV selection.

This statistical mismatch has direct implications for the stability of prefill-stage KV selection. Existing methods typically rely on prefilling-stage signals, such as local attention behavior, sequence-level activation statistics, and KV similarity, to estimate the importance of the token. When these signals are under-scaled relative to true decoding-time queries, the resulting importance estimates become distorted. Figure 1b quantifies this effect using *attention mass coverage*, measured as the fraction of decoding-time attention probability mass assigned to prompt KV tokens retained after prefilling. Following the standard prefill-only protocol, KV selection is performed once at the end of prefilling, and the compressed prompt KV cache is kept fixed during decoding. Across methods, attention coverage is consistently reduced and exhibits high variance under tight cache budgets, indicating that KV sets selected from prefilling statistics fail to reliably capture decoding-time KV usage.

Taken together, these results demonstrate that prefill-stage KV selection based solely on prefilling-stage statistics is inherently brittle in multimodal settings. The observed prefill-decode statistical mismatch highlights the need to explicitly account for decoding-time query behavior while remaining strictly within the prefill-only regime. This insight forms the basis for the decode-aware prefill-stage KV selection approach developed in the next section, which achieves the highest attention coverage shown in Figure 1b.

3 Method

Core Idea. MM-ShiftKV addresses the instability of prefill-stage KV selection in multimodal inference by explicitly approximating how prompt keys will be accessed by *future decoding queries*. Since decoding-time queries are unavailable during prefilling, the core idea is to construct a set of synthetic *query proxies* during the prefill stage that approximate the distributional properties of decoding queries. Prompt keys that are consistently attended by these query proxies are more likely to be important during decoding and should therefore be retained under the limited KV-cache budget.

3.1 Problem Definition

Before describing our method, we formalize the prefill-stage KV selection problem in multimodal inference.

Input Sequence. Given a multimodal input sequence

$$X = (x_1, x_2, \ldots, x_T), \tag{1}$$

where tokens include both visual and textual modalities, and T denotes the sequence length of the *prefill stage*.

Prompt KV cache. During prefilling, the model computes KV representations for all prompt tokens at each transformer layer ℓ and KV head h. The resulting set of KV pairs forms the *prompt KV cache*:

$$\mathcal{C}^{(\ell,h)} = \{(k_t^{(\ell,h)}, v_t^{(\ell,h)})\}_{t\in\mathcal{T}}, \tag{2}$$

where $\mathcal{T} = \{1, \ldots, T\}$ indexes prompt tokens. These prompt KVs are repeatedly accessed during decoding and dominate memory consumption and attention computation in inference.

Prefill-stage KV selection. With a given cache budget $C_{\ell,h}$ for each layer ℓ and KV head h, the objective of the prefill-stage KV selection is to retain a subset

$$\mathcal{C}'^{(\ell,h)} \subseteq \mathcal{C}^{(\ell,h)}, \qquad |\mathcal{C}'^{(\ell,h)}| \le C_{\ell,h}, \tag{3}$$

which is reused throughout the decoding to reduce memory usage and attention cost while preserving generation quality.

In practice, the KV selection is performed independently for each layer and head, and executed once at the end of *prefilling*. Note that the selected prompt KV cache remains fixed during decoding, while KV pairs from newly generated tokens are appended in a standard autoregressive manner.

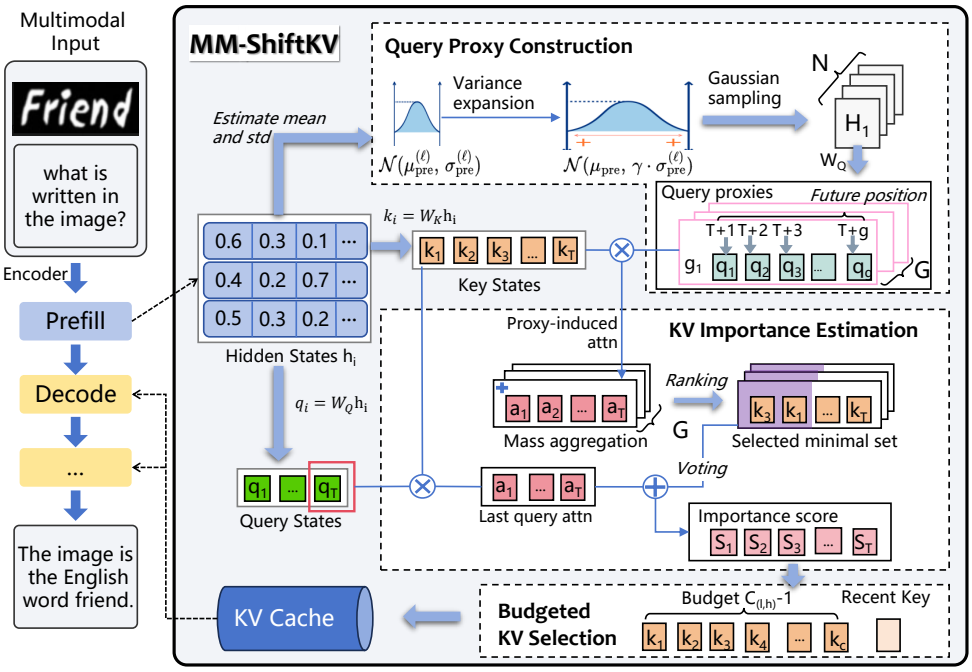


Figure 2: Overview of MM-ShiftKV. The method computes global statistics from prefilling hidden states and constructs variance-expanded, decode-aware *query proxies* during prefilling. These query proxies are used to estimate prompt KV importance via attention-mass aggregation and group-wise voting with a last-token anchor, yielding a compact KV cache under a fixed budget.

3.2 MM-ShiftKV

Figure 2 provides an overview of MM-ShiftKV. The method consists of three steps executed during prefilling: (1) constructing decode-aware query proxies, (2) estimating prompt KV importance via mass-based voting, and (3) performing budgeted KV selection based on aggregated importance scores. All steps are strictly prefill-only and introduce no decoding-time intervention.More implementation details can be found in Appendix B.

3.2.1 Query Proxy Construction

To approximate decoding-time query behavior, MM-ShiftKV constructs a set of synthetic *query proxies* using prefilling-stage statistics.

Prefilling hidden-state statistics. Let $h^{(\ell)} \in \mathbb{R}^d$ denote the hidden states produced at layer ℓ during prefilling. We summarize these representations using element-wise statistics:

$$\mu_{\text{pre}}^{(\ell)} = \text{mean}(h^{(\ell)}), \qquad \sigma_{\text{pre}}^{(\ell)} = \text{std}(h^{(\ell)}), \quad (4)$$

computed across prompt tokens for each feature dimension.

Variance-expanded sampling. As shown in Section 2, decoding-time queries exhibit substantially larger variance than prefilling-stage representations. To approximate this effect, we introduce a variance expansion factor $\gamma > 1$ and sample N *proxy hidden states* $\{\tilde{H}_i^{(\ell)}\}_{i=1}^N$ from

$$\tilde{H}_i^{(\ell)} \sim \mathcal{N}\Big(\mu_{\text{pre}}^{(\ell)},\ \text{diag}\Big((\gamma\sigma_{\text{pre}}^{(\ell)})^2\Big)\Big). \quad (5)$$

Query proxy projection and positioning. Each proxy hidden state is projected into the query space using the model's query projection matrix:

$$q_i^{(\ell,h)} = W_Q^{(\ell,h)} \tilde{H}_i^{(\ell)}, \qquad i = 1, \ldots, N. \quad (6)$$

The resulting vectors $\{q_i^{(\ell,h)}\}$ are referred to as *query proxies*. To reflect that decoding queries attend to prompt keys from future positions, query proxies are assigned synthetic future positions and encoded using the model's rotary positional embedding (RoPE), while prompt keys retain their prefilling-stage positional encodings.

3.2.2 KV Importance Estimation

Given a set of query proxies, MM-ShiftKV estimates the importance of each prompt key based on how consistently it attends across diverse proxies.

Proxy-induced attention. For each query proxy $q_i^{(\ell,h)}$, we compute its attention distribution over

prompt keys:

$$a_t\left(q_i^{(\ell,h)}\right) = \mathrm{softmax}\left(\frac{\left(q_i^{(\ell,h)}\right)^\top k_t^{(\ell,h)}}{\sqrt{d}}\right), t \in \mathcal{T} \tag{7}$$

Mass aggregation and voting. The N query proxies are partitioned into G disjoint groups $\{\mathcal{I}_{g'}\}_{g'=1}^{G}$. For each group g', attention masses are aggregated as

$$\bar{a}_t^{(g')} = \sum_{i \in \mathcal{I}_{g'}} a_t\left(q_i^{(\ell,h)}\right), \quad t \in \mathcal{T} \tag{8}$$

Prompt keys are then ranked in descending order of $\bar{a}_t^{(g')}$, and we define $\mathcal{S}_{g'} \subseteq \mathcal{T}$ as the *smallest* set of keys whose cumulative mass satisfies

$$\sum_{t \in \mathcal{S}_{g'}} \bar{a}_t^{(g')} \geq \tau \sum_{t \in \mathcal{T}} \bar{a}_t^{(g')}, \tag{9}$$

where $\tau \in (0, 1)$ is a predefined mass threshold. Each selected key receives one vote, and the final importance score of key t is given by

$$\mathrm{vote}(t) = \sum_{g'=1}^{G} \mathbf{1}[t \in \mathcal{S}_{g'}]. \tag{10}$$

3.2.3 Budgeted KV Selection

KV selection is performed independently for each layer ℓ and KV head h. Among the prompt keys, we retain the most recent key to ensure decoding stability and select the remaining keys based on their aggregated importance scores. Specifically, the top $C_{\ell,h} - 1$ keys ranked by $\mathrm{vote}(t)$ is used to form the compressed prompt KV cache.

Overhead. Similar to existing prefill-only approaches, MM-ShiftKV introduces additional computation only during the *prefill stage*. For each layer and KV head, it computes attention between N query proxies and $|\mathcal{T}|$ prompt keys once, followed by group-wise aggregation. The time complexity scales as $O(N|\mathcal{T}|)$, while the peak additional memory can be bounded by $O(|\mathcal{T}|)$ using streaming implementations.

4 Experiments

We conduct extensive experiments to evaluate **MM-ShiftKV** under memory-constrained multimodal inference. Following prior work on prompt KV cache compression (e.g., SnapKV (Li et al., 2024b), Expectedattn (Devoto et al., 2025), and KEYDIFF (Park et al., 2025)), we focus on a *one-shot prefilling* setting, where the KV cache is compressed once after prompt encoding and kept fixed throughout decoding.

Our experiments aim to answer the following questions: (i) how MM-ShiftKV compares with existing *prefill-stage* KV compression baselines under strict KV-cache budgets, (ii) how performance degrades as the cache budget decreases, and (iii) what accuracy-memory-latency trade-offs can be achieved while remaining compatible with FlashAttention (Dao, 2023)-style decoding kernels.

4.1 Experimental Setup

Models, Datasets, and Metrics. We evaluate MM-ShiftKV on two representative multimodal large language models, **Qwen2.5-VL-Instruct** (Bai et al., 2025) and **LLaVA-v1.6-Vicuna-7B** (Li et al., 2024a). Experiments are conducted on a diverse suite of multimodal benchmarks covering document understanding, OCR-centric visual question answering, chart reasoning, and image captioning. We employ **OCRBench**(Liu et al., 2023) using exact-match accuracy to measure precise text recognition and **DocVQA**(Mathew et al., 2021) using Average Normalized Levenshtein Similarity (ANLS) as the metric. For chart reasoning and visual question answering, we use exact-match accuracy on **ChartQA**(Masry et al., 2022), **TextVQA**(Singh et al., 2019), and **MMMU**(Yue et al., 2024) to measure the fraction of correctly answered questions. For image captioning, we evaluate **TextCaps**(Sidorov et al., 2020) using caption quality consensus with reference captions (CIDEr) as the metric. For all metrics, higher values indicate better performance.

Baselines. We compare against eviction-free inference (**Full KV**) and strong prefilling-stage KV compression baselines, including **SnapKV** (Li et al., 2024b), **ExpectedAttn** (El Maalouly, 2022), **StreamingLLM** (Xiao et al., 2023), and **KEYDIFF** (Park et al., 2025) (one-shot variant). All baselines are evaluated under identical KV-cache budgets and decoding settings, using their recommended configurations. More details are provided in the Appendix E.

Prefill-Only Evaluation Protocol and KV Budget. Following prior work , we adopt a one-shot prefilling protocol. Given an input prompt consisting of both visual and textual tokens, we first

Method	DocVQA				OCRBench				TextVQA				ChartQA				TextCaps				MMMU				Avg			
	64	128	256	512	64	128	256	512	64	128	256	512	64	128	256	512	64	128	256	512	64	128	256	512	64	128	256	512
FullKV	94.5				82.3				83.0				83.2				58.7				50.8				75.4			
StreamingLLM	41.0	45.1	56.5	69.1	29.3	48.5	65.2	72.0	52.4	60.6	69.4	76.8	68.7	74.4	80.2	82.3	22.1	34.3	45.2	53.2	50.2	50.4	50.2	50.4	44.0	52.2	61.1	67.3
ExpectedAttn	56.3	59.7	66.2	77.6	57.5	67.3	72.2	77.9	64.2	69.4	75.6	81.3	77.5	77.9	80.5	83.0	40.1	48.1	53.7	57.6	50.9	51.0	50.9	50.9	57.8	62.2	66.5	71.4
SnapKV	75.6	88.8	93.0	94.1	53.5	72.1	79.1	81.1	69.8	78.7	81.6	82.9	80.0	84.1	83.6	83.2	29.3	46.6	55.9	58.9	50.7	50.2	50.6	50.4	59.8	70.1	74.0	75.1
KEYDIFF	51.0	61.5	74.7	86.7	41.7	62.7	74.8	79.1	72.1	78.6	82.0	83.0	73.0	80.8	84.1	83.2	42.2	49.9	55.2	59.2	50.9	50.7	50.6	50.4	55.2	64.0	70.2	73.6
MM-ShiftKV	**81.6**	**88.9**	**92.8**	**94.3**	**68.8**	**76.1**	**80.2**	**81.9**	**80.0**	**82.2**	**82.8**	**82.9**	**84.4**	**84.3**	**83.7**	**83.3**	**50.4**	**58.9**	**60.0**	**60.1**	**50.6**	**50.4**	**50.4**	**50.4**	**69.3**	**73.5**	**75.0**	**75.5**

Table 1: Results on multimodal benchmarks under different per-head KV-cache budgets (64/128/256/512) using **Qwen2.5-VL-Instruct**. FullKV denotes standard inference without KV cache compression. Avg is the arithmetic mean over DocVQA, OCRBench, TextVQA, ChartQA, TextCaps, and MMMU.

Method	DocVQA				OCRBench				TextVQA				ChartQA				TextCaps				MMMU				Avg			
	64	128	256	512	64	128	256	512	64	128	256	512	64	128	256	512	64	128	256	512	64	128	256	512	64	128	256	512
FullKV	68.0				52.0				65.0				55.0				73.0				36.4				58.2			
StreamingLLM	27.3	28.6	30.8	35.8	12.9	19.4	25.4	31.8	39.4	41.0	42.8	47.1	23.9	23.9	24.3	27.1	28.8	33.0	36.2	42.1	36.9	36.4	36.4	36.2	28.2	30.4	32.7	36.7
ExpectedAttn	40.6	47.0	54.8	62.1	32.3	37.9	43.5	47.9	54.2	57.2	60.5	62.6	39.0	43.8	49.7	51.8	54.9	60.0	64.6	67.0	36.1	36.6	36.6	36.6	42.9	47.1	51.6	54.7
SnapKV	49.7	59.0	64.3	67.0	33.5	40.8	46.2	51.1	56.1	60.1	62.2	64.1	42.9	45.8	49.8	54.2	44.2	56.1	65.3	69.7	36.4	36.7	36.4	36.7	43.8	49.8	54.0	57.1
KEYDIFF	43.0	49.2	56.5	62.6	29.8	36.8	43.5	48.4	57.7	60.3	63.1	64.0	40.1	44.5	49.6	52.3	59.7	66.3	70.7	73.8	36.3	36.6	36.7	36.8	44.4	49.0	53.4	56.3
MM-ShiftKV	**56.3**	**61.8**	**65.4**	**67.1**	**41.1**	**45.7**	**49.4**	**51.9**	**62.1**	**62.8**	**64.2**	**64.6**	**51.5**	**52.1**	**52.6**	**54.5**	**63.2**	**68.9**	**73.1**	**74.8**	**36.6**	**36.4**	**36.6**	**36.6**	**51.8**	**54.6**	**56.9**	**58.3**

Table 2: Results on multimodal benchmarks under different per-head KV-cache budgets (64/128/256/512) using **LLaVA-v1.6-Vicuna-7B**. FullKV denotes standard inference without KV cache compression. Avg is the arithmetic mean over DocVQA, OCRBench, TextVQA, ChartQA, TextCaps, and MMMU.

Method	DocVQA				OCRBench				TextVQA				ChartQA				TextCaps				MMMU				Avg			
	64	128	256	512	64	128	256	512	64	128	256	512	64	128	256	512	64	128	256	512	64	128	256	512	64	128	256	512
PyramidKV	49.0	60.0	64.9	67.1	31.2	40.9	47.6	50.1	55.0	60.0	62.6	64.1	41.0	47.1	53.0	54.9	41.6	57.4	64.9	69.5	36.3	36.4	36.4	36.4	42.3	50.3	54.9	57.0
+ours	↑ **8.9**	↑ **2.5**	↑ **1.0**	↑ **0.1**	↑ **11.6**	↑ **6.5**	↑ **1.9**	↑ **1.2**	↑ **6.4**	↑ **3.3**	↑ **1.4**	↑ **0.5**	↑ **8.4**	↑ **5.6**	↑ **1.7**	↑ **0.1**	↑ **23.2**	↑ **11.5**	↑ **8.2**	↑ **4.8**	↑ **0.1**	**0.0**	↑ **0.1**	↑ **0.1**	↑ **9.8**	↑ **4.9**	↑ **2.4**	↑ **1.1**
AdaKV	54.1	60.6	65.2	67.5	35.1	43.0	48.4	50.8	56.0	59.8	61.9	64.2	43.8	45.7	49.1	54.8	47.2	59.2	65.9	70.3	36.1	36.3	36.4	36.4	45.4	50.8	54.5	57.3
+ours	↑ **4.7**	↑ **3.4**	↑ **0.7**	**0.0**	↑ **9.2**	↑ **4.1**	↑ **2.6**	↑ **1.9**	↑ **5.1**	↑ **2.0**	↑ **1.5**	↑ **0.5**	↑ **5.1**	↑ **3.8**	↑ **3.4**	**0.0**	↑ **19.0**	↑ **11.7**	↑ **7.2**	↑ **4.3**	↑ **0.2**	↑ **0.1**	**0.0**	**0.0**	↑ **7.2**	↑ **4.2**	↑ **2.6**	↑ **1.1**
SparseMM	63.7	66.7	67.8	68.0	47.8	50.1	52.3	52.8	63.1	64.1	64.7	64.7	52.7	53.8	53.9	54.8	59.9	70.7	72.8	73.3	36.4	36.6	36.3	36.4	53.9	57.0	58.0	58.3
+ours	↑ **1.2**	↑ **0.3**	↑ **0.1**	**0.0**	↑ **1.3**	↑ **0.7**	↑ **0.1**	**0.0**	↑ **0.4**	**0.0**	**0.0**	**0.0**	↑ **0.1**	↑ **0.1**	↑ **0.5**	**0.0**	↑ **6.0**	**0.0**	↑ **0.1**	↑ **0.1**	**0.0**	↓ **0.2**	↑ **0.1**	**0.0**	↑ **1.5**	↑ **0.2**	↑ **0.2**	**0.0**

Table 3: Results on multimodal benchmarks with different KV-cache *budget allocation* methods under various per-head KV-cache budgets (64/128/256/512) using **LLaVA-v1.6-Vicuna-7B**. **+ours** applies MM-ShiftKV as a prefill-only KV selection module on top of the original allocation strategy, without changing its budget. Avg is the arithmetic mean over DocVQA, OCRBench, TextVQA, ChartQA, TextCaps, and MMMU.

perform standard prefilling to compute the full KV cache. Each method is then applied once at the end of prefilling to score and select a subset of KV pairs according to its selection policy, producing a compressed prompt KV cache that is kept fixed throughout decoding. No decode-time eviction, re-ranking, or decoding-time statistics are used.

KV-cache budgets are defined at the level of individual KV heads for each transformer layer, with per-head budgets $C \in \{64, 128, 256, 512\}$. Unless otherwise specified, the same budget is applied uniformly across all layers and all KV heads. For models with Grouped-Query Attention (GQA) (Ainslie et al., 2023) or Multi-Query Attention (MQA) (Komatsuzaki et al., 2022), where multiple query heads share one KV head, attention statistics from the corresponding query heads are aggregated, and the budget is applied at the head level.

Implementation and Hyperparameters. Unless otherwise specified, we use N=512 decode-aligned query proxies, partitioned into G=32 groups of size g=16 ($N = Gg$), with variance expansion factor γ=10, attention mass threshold τ=0.95, and last-token anchor weight λ=1. All hyperparameters are fixed globally and shared across models, datasets, and budgets. We apply greedy decoding with a maximum generation length of 64. All experiments are conducted on NVIDIA H100 80GB GPUs using CUDA 12.8 and FlashAttention 2.4.1 with mixed-precision (fp16/bf16). Detailed proofs for hyperparameter selection are detailed in the Appendix D.

Task	OCRBench	DocVQA	ChartQA	TextVQA	TextCaps
LLaVA-Series	1700	2433	2270	2376	2376
Qwen2.5-VL-Instruct	1245	4830	642	1024	1024

Table 4: Average number of input tokens across benchmarks. Text instructions are short, and visual tokens constitute the majority of the input sequence.

Results on multi-modal benchmarks. We evaluate **MM-ShiftKV** on a diverse set of multimodal benchmarks under different per-head KV-cache

budgets. Across both backbone models and all tasks, MM-ShiftKV consistently outperforms existing prefill-stage KV selection baselines, with the performance gap becoming more pronounced as the KV budget decreases.

Under extreme compression (64 tokens per KV head), baseline methods often suffer from performance degradation, particularly on OCR- and grounding-centric tasks. Compared to strong prefill-only baselines, MM-ShiftKV achieves relative accuracy improvements on the order of 20%–30% in representative document understanding benchmarks. When compared to decode-agnostic or heuristic KV selection methods, relative performance gains can exceed 50%, highlighting the brittleness of decode-unaware KV management under multimodal inputs.

Consistent trends are also observed on generation-oriented tasks. Under the same extreme budget, MM-ShiftKV yields relative improvements of approximately 40% or more in generation quality compared to prefill-only baselines, indicating substantially better preservation of visually grounded information required for coherent multimodal generation.

In addition to standalone performance, MM-ShiftKV remains complementary to KV budget allocation strategies. As shown in Table 3, integrating MM-ShiftKV with representative budget allocation methods leads to consistent relative improvements, typically in the range of 10%–20% under the most restrictive budgets. These results indicate that correcting the prefill–decode scale mismatch improves KV *selection quality* independently of how KV budgets are allocated across layers or heads.

Overall, the results demonstrate that MM-ShiftKV provides a robust and effective prefill-only solution for multimodal inference under strict KV-cache constraints, offering favorable accuracy–memory trade-offs while remaining compatible with different KV budgeting schemes.

4.2 Ablation Study

We analyze the contribution of individual components in **MM-ShiftKV** through ablation experiments under a fixed per-head KV cache budget. All ablation experiments are conducted using the **Qwen2.5-VL-7B-Instruct** model, following the same prefill-only evaluation protocol as in the main experiments. We start from a lightweight *LastAttn* baseline, which incorporates only the attention induced by the last prefill query, and progressively add decode-aware query sampling, variance expansion, and group-wise voting. Results are reported on **OCRBench** and **TextCaps** in Table 5.

Variant	Samp.	Var. Exp.	G Vote	OCR	TextCaps
LastAttn				52.3	40.8
+ Samp.	✓			54.5	45.5
+ Var. Exp.	✓	✓		59.6	48.6
+ GVote	✓	✓	✓	**68.3**	**50.4**

Table 5: Ablation study under a fixed per-head KV cache budget. **Samp.**, **Var. Exp.**, and **GVote** denote query proxy sampling, variance expansion, and group-wise voting, respectively.

Overall Effect. As shown in Table 5, the *LastAttn* baseline achieves 52.3 OCR accuracy and 40.8 CIDEr on TextCaps, providing a simple but stable prefill-only reference. Enabling decode-aware query sampling (**+ Samp.**) consistently improves performance, increasing OCR accuracy to 54.5 and TextCaps CIDEr to 45.5. This indicates that sampling query proxies aligned with the decoding stage provides a more informative estimate of future attention behavior than relying solely on prefilling-stage statistics.

Adding variance expansion (**+ Var. Exp.**) yields further gains, improving performance to 59.6 on OCRBench and 48.6 CIDEr on TextCaps. This suggests that query proxies derived directly from prefilling statistics are systematically under-scaled, and that scale calibration is critical for capturing the variability of decoding-time queries in multimodal inference.

Finally, incorporating group-wise voting (**+ GVote**) achieves the best overall results, reaching 68.3 OCR accuracy and 50.4 CIDEr on TextCaps. By aggregating attention mass across groups of query proxies, group-wise voting effectively reduces estimation variance and stabilizes KV ranking under tight cache budgets.

Overall, these results demonstrate that decode-aware query sampling, variance expansion, and group-wise voting are complementary components. Starting from a simple last-query attention anchor, progressively introducing decode-aware calibration and variance reduction is necessary to fully realize the performance gains of **MM-ShiftKV** in prefill-only multimodal inference.

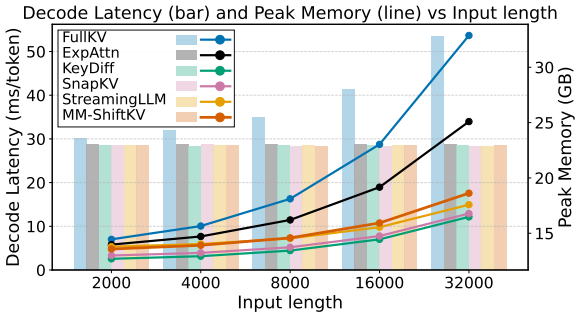


Figure 3: End-to-end decoding latency and peak GPU memory usage under increasing input lengths. Bars denote per-token decoding latency and lines indicate peak GPU memory usage. All methods are evaluated with a fixed per-head KV budget of 256 and a maximum output length of 100 tokens.

4.3 Efficiency Evaluation

Setup. We evaluate inference efficiency under long-context multimodal settings by varying the input length in $\{2K, 4K, 8K, 16K, 32K\}$ while fixing the output length to 100 tokens. Following the main configuration, we apply a per-head KV cache budget of 256 after prefilling-stage compression and report per-token decoding latency and peak GPU memory usage. All experiments use FlashAttention-compatible decoding kernels.

Decoding Latency. As shown in Figure 3, MM-ShiftKV consistently reduces per-token decoding latency compared to FullKV, with the gap widening as the input length increases. While FullKV exhibits steadily growing latency, MM-ShiftKV maintains an almost constant latency profile, achieving up to a $1.9\times$ speedup at 32K input length.

Memory Cost. By bounding the prompt KV cache before decoding, MM-ShiftKV also substantially reduces peak GPU memory usage. At 32K input length, peak memory consumption is reduced from approximately 32.9 GB to 18.6 GB, corresponding to a reduction of about 43%. Overall, MM-ShiftKV offers a more favorable latency–memory trade-off than existing prefilling-stage compression baselines.

5 Related Work

Current research on KV cache compression can be broadly categorized into two types, static pruning (Li et al., 2024b; Jiang et al., 2024; Devoto et al., 2024, 2025; Park et al., 2025) and dynamic eviction (Chen et al., 2025; Child et al., 2019), based on the intervention stage. Static pruning methods are primarily deployed during the prefill stage, where the importance of each KV pair is evaluated using predefined metrics, and lower-scoring KV pairs are pruned to reduce the initial memory footprint. Dynamic eviction methods continuously discard tokens during the decode stage to maintain a fixed-size cache. There are three paradigms for KV cache eviction: first, attention-based eviction (Li et al., 2024b; Devoto et al., 2025); second, value-based analysis (Devoto et al., 2024; Park et al., 2025); third, heuristic-based structural eviction (Cai et al., 2024b; Xiao et al., 2023).

A key limitation of these methods lies in their implicit assumption of "distributional homogeneity" (Devoto et al., 2025; Cai et al., 2024a). However, MM-ShiftKV finds that in multimodal scenarios, there is a significant numerical distribution shift between these two stages. MM-ShiftKV explicitly analyzes the differences in numerical behavior between the prefill and decode stages in multimodal contexts. By introducing a group voting mechanism, MM-ShiftKV achieves more robust and accurate KV selection.

Additionally, some research efforts cover techniques such as model quantization, early exiting, and speculative decoding (Lin et al., 2024; Elhoushi et al., 2024; Child et al., 2019; Liu et al., 2024b; Xu et al., 2025; Su et al., 2025), while hardware methods represent another important direction for efficient inference (Dao, 2023; Zheng et al., 2025). In multimodal scenarios, specialized strategies have been proposed in existing works (Lin et al., 2025), including the dynamic allocation of attention head budgets (Wang et al., 2025; Yang et al., 2025a; Wan et al., 2024) and token pruning within visual encoders (Shen et al., 2024; Yang et al., 2025b; Zhang et al., 2025). MM-ShiftKV is complementary to these methods and can be combined with multiple aforementioned approaches simultaneously to obtain better performance.

6 Conclusion

We studied a previously overlooked aspect of multimodal inference: the systematic mismatch between prefill and decoding stages. By making *prefill-stage* KV selection decode-aware, MM-ShiftKV provides a simple, training-free solution that aligns prompt KV management with decoding-time query distributions. Our results highlight decode-aware KV management as a key design principle for scalable multimodal inference.

7 Ethical Considerations

All experiments in our work are conducted using open-source datasets and models. Our research is solely aimed at enabling efficient inference of multimodal large models (MLLMs), and does not involve any human subject, sensitive data, or commercial applications.

8 Limitations

While MM-ShiftKV exhibits prominent advantages in decoding latency optimization and inference accuracy performance under resource-constrained conditions, it still has certain limitations: First, the additional computation introduced by the voting scoring for KV eviction processing in the prefilling phase incurs a certain overhead. Although this overhead is negligible compared to the overall inference latency, there remains room for further acceleration in the prefilling phase. Second, MM-ShiftKV performs KV eviction only in the prefilling phase. Although the KV cache in the prefilling phase accounts for the largest proportion in most visual question answering tasks, this design poses certain challenges in some tasks that require long-context reasoning. Third, our method primarily focuses on image and video benchmarks (which generate a large amount of KV cache), but it may face new challenges for other tasks and modalities (e.g., audio-based multimodal large models or long-context reasoning scenarios).

Nevertheless, MM-ShiftKV is compatible with multiple future extension directions, including combination with attention head budget allocation methods, integration of offloading techniques in the decoding phase, and joint application with other compression paradigms (e.g., model quantization, speculative decoding, etc.). We believe these directions will further enhance the generality and scalability of MM-ShiftKV.

References

Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebrón, and Sumit Sanghai. 2023. GQA: Training generalized multi-query transformer models from multi-head checkpoints. *Preprint*, arXiv:2305.13245.

Kazi Hasan Ibn Arif, JinYi Yoon, Dimitrios S. Nikolopoulos, Hans Vandierendonck, Deepu John, and Bo Ji. 2025. HiRED: Attention-guided token dropping for efficient inference of high-resolution vision-language models. In *Proceedings of the Thirty-Ninth AAAI Conference on Artificial Intelligence*.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, and 1 others. 2025. Qwen2.5-vl technical report. *Preprint*, arXiv:2502.13923.

Tianle Cai, Yuhong Li, Zhengyang Geng, Hongwu Peng, Jason D. Lee, Deming Chen, and Tri Dao. 2024a. Medusa: Simple LLM inference acceleration framework with multiple decoding heads. *Preprint*, arXiv:2401.10774.

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

Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang. 2025. An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models. In *Computer Vision – ECCV 2024*, pages 19–35, Cham. Springer Nature Switzerland.

Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. 2019. Generating long sequences with Sparse Transformers. *Preprint*, arXiv:1904.10509.

Tri Dao. 2023. Flashattention-2: Faster attention with better parallelism and work partitioning. *Preprint*, arXiv:2307.08691.

Alessio Devoto, Maximilian Jeblick, and Simon Jégou. 2025. Expected attention: Kv cache compression by estimating attention from future query distributions. *Preprint*, arXiv:2510.00636.

Alessio Devoto, Yu Zhao, Simone Scardapane, and Pasquale Minervini. 2024. A simple and effective l_2 norm-based strategy for KV cache compression. *Preprint*, arXiv:2406.11430.

Nicolas El Maalouly. 2022. Exact matching: Algorithms and related problems. *Preprint*, arXiv:2203.13899.

Mostafa Elhoushi, Akshat Shrivastava, Diana Liskovich, Basil Hosmer, Bram Wasti, Liangzhen Lai, Anas Mahmoud, Bilge Acun, Saurabh Agarwal, Ahmed Roman, Ahmed A. Aly, Beidi Chen, and Carole-Jean Wu. 2024. LayerSkip: Enabling early exit inference and self-speculative decoding. *Preprint*, arXiv:2404.16710.

Huiqiang Jiang, Yucheng Li, Chengruidong Zhang, Qianhui Wu, Xufang Luo, Surin Ahn, Zhenhua Han, Amir H. Abdi, Dongsheng Li, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. 2024. MInference 1.0: Accelerating pre-filling for long-context LLMs via dynamic sparse attention. In *Advances in Neural Information Processing Systems (NeurIPS 2024)*, volume 37, pages 52481–52515.

Aran Komatsuzaki, Joan Puigcerver, James Lee-Thorp, Carlos Riquelme Ruiz, Basil Mustafa, Joshua Ainslie, Yi Tay, Mostafa Dehghani, and Neil Houlsby. 2022. Sparse Upcycling: Training Mixture-of-Experts from dense checkpoints. *Preprint*, arXiv:2212.05055.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, and 1 others. 2024a. Llava-onevision: Easy visual task transfer. *Preprint*, arXiv:2408.03326.

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

Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Wei-Ming Chen, Wei-Chen Wang, Guangxuan Xiao, Xingyu Dang, Chuang Gan, and Song Han. 2024. AWQ: Activation-aware weight quantization for on-device LLM compression and acceleration. In *Proceedings of Machine Learning and Systems (MLSys)*, volume 6, pages 87–100.

Junyan Lin, Haoran Chen, Yue Fan, Yingqi Fan, Xin Jin, Hui Su, Jinlan Fu, and Xiaoyu Shen. 2025. Multi-layer visual feature fusion in multimodal LLMs: Methods, analysis, and best practices. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4156–4166.

Jiahui Liu, Praveen Ponnusamy, Tianle Cai, Hanlin Guo, Yoon Kim, and Ben Athiwaratkun. 2024a. Training-free activation sparsity in large language models. *Preprint*, arXiv:2408.14690.

Yuliang Liu, Zhang Li, Mingxin Huang, Biao Yang, Wenwen Yu, Chunyuan Li, Xucheng Yin, Cheng-lin Liu, Lianwen Jin, and Xiang Bai. 2023. OCRBench: On the hidden mystery of OCR in large multimodal models. *Preprint*, arXiv:2305.07895.

Zirui Liu, Jiayi Yuan, Hongye Jin, Shaochen Zhong, Zhaozhuo Xu, Vladimir Braverman, Beidi Chen, and Xia Hu. 2024b. KIVI: A tuning-free asymmetric 2-bit quantization for KV cache. *Preprint*, arXiv:2402.02750.

Ahmed Masry, Do Xuan Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. 2022. ChartQA: A benchmark for question answering about charts with visual and logical reasoning. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2263–2279, Dublin, Ireland. Association for Computational Linguistics.

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

Junyoung Park, Dalton Jones, Matthew J. Morse, Raghavv Goel, Mingu Lee, and Chris Lott. 2025. Keydiff: Key similarity-based kv cache eviction for long-context llm inference in resource-constrained environments. *Preprint*, arXiv:2504.15364.

David Peer, Philemon Schöpf, Volckmar Nebendahl, Alexander Rietzler, and Sebastian Stabinger. 2024. ANLS* – a universal document processing metric for generative large language models. *Preprint*, arXiv:2402.03848.

Xiaoqian Shen, Yunyang Xiong, Changsheng Zhao, Lemeng Wu, Jun Chen, Chenchen Zhu, Zechun Liu, Fanyi Xiao, Balakrishnan Varadarajan, Florian Bordes, Zhuang Liu, Hu Xu, Hyunwoo J. Kim, Bilge Soran, Raghuraman Krishnamoorthi, Mohamed Elhoseiny, and Vikas Chandra. 2024. LongVU: Spatiotemporal adaptive compression for long video-language understanding. *Preprint*, arXiv:2410.17434.

Oleksii Sidorov, Ronghang Hu, Marcus Rohrbach, and Amanpreet Singh. 2020. TextCaps: A dataset for image captioning with reading comprehension. In *Computer Vision – ECCV 2020*, pages 742–758, Cham. Springer International Publishing.

Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. 2019. Towards VQA models that can read. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Zunhai Su, Zhe Chen, Wang Shen, Hanyu Wei, Linge Li, Huangqi Yu, and Kehong Yuan. 2025. RotateKV: Accurate and robust 2-bit KV cache quantization for LLMs via outlier-aware adaptive rotations. *Preprint*, arXiv:2501.16383.

Keda Tao, Can Qin, Haoxuan You, Yang Sui, and Huan Wang. 2025. DyCoke: Dynamic compression of tokens for fast video large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18992–19001.

Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. 2014. CIDEr: Consensus-based image description evaluation. *Preprint*, arXiv:1411.5726.

Zhongwei Wan, Ziang Wu, Che Liu, Jinfa Huang, Zhihong Zhu, Peng Jin, Longyue Wang, and Li Yuan. 2024. LOOK-M: Look-once optimization in KV cache for efficient multimodal long-context inference. *Preprint*, arXiv:2406.18139.

Jiahui Wang, Zuyan Liu, Yongming Rao, and Jiwen Lu. 2025. Sparsemm: Head sparsity emerges from visual concept responses in mllms. *Preprint*, arXiv:2506.05344.

Chaojun Xiao, Pengle Zhang, Xu Han, Guangxuan Xiao, Yankai Lin, Zhengyan Zhang, Zhiyuan Liu, and Maosong Sun. 2024. InfLLM: Training-free

long-context extrapolation for LLMs with an efficient context memory. In *Proceedings of the 38th International Conference on Neural Information Processing Systems (NeurIPS 2024)*, Red Hook, NY, USA. Curran Associates Inc.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. 2023. Efficient streaming language models with attention sinks. *Preprint*, arXiv:2309.17453.

Jiaming Xu, Jiayi Pan, Yongkang Zhou, Siming Chen, Jinhao Li, Yaoxiu Lian, Junyi Wu, and Guohao Dai. 2025. SpecEE: Accelerating large language model inference with speculative early exiting. In *Proceedings of the 52nd Annual International Symposium on Computer Architecture (ISCA '25)*, pages 467–481, New York, NY, USA. Association for Computing Machinery.

Cheng Yang, Yang Sui, Jinqi Xiao, Lingyi Huang, Yu Gong, Chendi Li, Jinghua Yan, Yu Bai, Ponnuswamy Sadayappan, Xia Hu, and Bo Yuan. 2025a. TopV: Compatible token pruning with inference time optimization for fast and low-memory multimodal vision-language model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19803–19813.

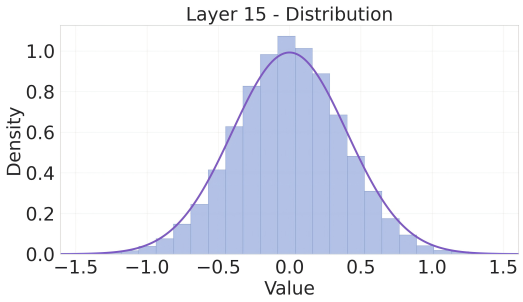
Senqiao Yang, Yukang Chen, Zhuotao Tian, Chengyao Wang, Jingyao Li, Bei Yu, and Jiaya Jia. 2025b. VisionZip: Longer is better but not necessary in vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19792–19802.

Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, Cong Wei, Botao Yu, Ruibin Yuan, Renliang Sun, Ming Yin, Boyuan Zheng, Zhenzhu Yang, Yibo Liu, Wenhao Huang, and 3 others. 2024. MMMU: A massive multi-discipline multimodal understanding and reasoning benchmark for expert AGI. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9556–9567.

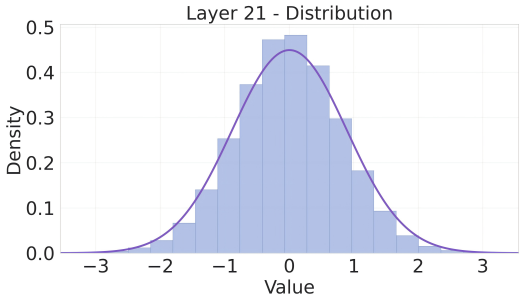
Kaichen Zhang, Bo Li, Peiyuan Zhang, Fanyi Pu, Joshua Adrian Cahyono, Kairui Hu, Shuai Liu, Yuanhan Zhang, Jingkang Yang, Chunyuan Li, and Ziwei Liu. 2024. LMMs-Eval: Reality check on the evaluation of large multimodal models. *Preprint*, arXiv:2407.12772.

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. 2025. SparseVLM: Visual token sparsification for efficient vision-language models inference.

Simeng Zheng, Chih-Hui Ho, Wenyu Peng, and Paul H. Siegel. 2025. Flash-Gen: Spatio-temporal generator for flash memory systems. *IEEE Transactions on Communications*, 73(2):1100–1113.



(a) Layer 15



(b) Layer 21

Figure 4: The figure shows the overall numerical distribution of hidden states across different layers of the **LLaVA-NeXT-Vicuna-7B** model during inference. We flattened the hidden states to calculate their overall numerical distribution, where the x-axis represents specific values and the y-axis denotes distribution density (with the calculation formula as follows: $\text{density}(x) = \frac{\text{freq}(x)}{N \cdot \Delta x}$,where $\text{density}(x)$: the empirical density (height of the histogram) at value x;$\text{freq}(x)$: the number of samples falling into the interval centered at (or starting from) x;N: the total number of all samples;Δx: the width of the interval (bin).).

A Theoretical Analysis of Prefill–Decode Scale Mismatch

This appendix provides a concise theoretical explanation of the prefill–decode scale mismatch and motivates variance-expanded, probe-based KV selection as a principled decode-aware approximation under prefill-only execution.

A.1 Distributional Shift Between Prefilling and Decoding

As shown in Figure 4, the overall numerical distribution of the hidden states across different layers of the model during inference follows a Gaussian distribution. We model hidden states at a given layer as Gaussian random variables. Empirically, decoding-time hidden states and query projections exhibit larger variance than those observed during

prefilling:

$$h_{\text{pre}} \sim \mathcal{N}(\mu, \Sigma_{\text{pre}}), \quad h_{\text{dec}} \sim \mathcal{N}(\mu, \Sigma_{\text{dec}}) \tag{11}$$

where

$$\Sigma_{\text{dec}} \succ \Sigma_{\text{pre}}. \tag{12}$$

Through linear query projection, this induces a corresponding variance gap in query distributions, causing prefilling-based query proxies to underestimate the support of true decoding-time queries.

A.2 Impact on Attention Estimation

For a fixed key k and Gaussian query $q \sim \mathcal{N}(\mu_q, \Sigma_q)$, the expected unnormalized attention score admits the closed form

$$\mathbb{E}\left[\exp\left(\frac{q^\top k}{\sqrt{d}}\right)\right] = \exp\left(\frac{\mu_q^\top k}{\sqrt{d}} + \frac{k^\top \Sigma_q k}{2d}\right). \tag{13}$$

Underestimating Σ_q therefore systematically underestimates expected attention mass, particularly for keys aligned with high-variance query directions, leading to biased KV ranking under constrained budgets.

A.3 Variance-Expanded and Probe-Based Approximation

To compensate for this bias without accessing decoding-time signals, we introduce a variance-expanded proxy distribution

$$\tilde{q} \sim \mathcal{N}(\mu_q, \gamma^2 \Sigma_{q,\text{pre}}), \qquad \gamma > 1. \tag{14}$$

This expansion enlarges the support of prefilling-based queries while preserving their mean structure.

KV importance is then estimated by sampling a finite number of query probes, yielding a Monte Carlo approximation of expected attention mass. Group-wise aggregation further reduces estimator variance without materializing attention matrices.

A.4 Summary

This analysis explains why uncalibrated prefill-based KV selection fails under distributional scale shift and why variance-expanded, probe-based attention estimation provides an effective and training-free decode-aware alternative for memory-constrained multimodal inference.

B Implementation Details

Our method supports Grouped-Query Attention (GQA) models such as Qwen2.5-VL-7B-Instruct and LLaVA-v1.6-Vicuna-7B. In GQA, query states have shape (B, L, H_q, d) and key-value states stored in the KV cache have shape (B, L, H_{kv}, d), with $H_q = H_{kv} \times G$. For attention computation, we repeat the key and value states along the head dimension to restore an MHA-like layout, yielding attention scores of shape (B, H_q, L_q, L_k). Although attention scores are computed at the query-head level, KV cache budgeting is performed at the key-value head level by aggregating the scores of query heads belonging to the same key-value head. KV importance is estimated during prefilling using synthetic query probes sampled from a recent context window ($C = 512$), with $N = 512$ samples per layer, generated either by a lightweight statistical predictor or a diagonal Gaussian fallback. The probe queries are grouped into $G_{\text{groups}} = 32$ groups to stabilize estimation, and tokens are ranked by aggregated attention scores. For each key-value head, we select the smallest set of tokens whose cumulative attention mass exceeds a fixed threshold (0.95), while always preserving the most recent token and incorporating the attention score from the last real query token. The selected tokens are restored to their original temporal order, concatenated with the most recent token, and inserted into the KV cache using the standard update interface, ensuring full compatibility with FlashAttention-style kernels. All experiments are conducted in a purely inference-time setting without additional training or fine-tuning.

C Additional Visualization Results

We present additional visualizations to complement the statistical observations in Section 2 and to provide intuitive evidence of the prefill–decode *scale mismatch* in long-context multimodal inference. All visualizations are obtained using **Qwen2-VL** on representative document understanding benchmarks, including **DocVQA** and **SynthDog**.

As shown in Figure 5, FullKV exhibits steadily increasing per-token decoding latency and prompt KV cache size as the input length grows. In contrast, MM-ShiftKV bounds the prompt KV cache after prefilling, resulting in near-constant decoding latency and significantly reduced memory usage. This visualization illustrates how prefill-only KV selection decouples decoding cost from the original

Algorithm 1 MM-ShiftKV (Prefill-only, Decode-aware KV Selection)

Require: Prefill hidden states $\{h_t^{(\ell)}\}_{t=1}^{T}$, prompt KVs $\{(k_t^{(\ell,h)}, v_t^{(\ell,h)})\}_{t=1}^{T}$, budget $C_{\ell,h}$
Ensure: Compressed KV cache $\mathcal{C}'^{(\ell,h)}$

```
Compute prefill statistics (μ_pre^(ℓ), σ_pre^(ℓ))
Sample N=Gg hidden states with variance expansion γ
Project samples to query proxies and apply future-position RoPE
Partition query proxies into G groups of size g
for each group g' do
    Aggregate attention mass over prompt keys
    Select minimal set covering fraction τ
    Vote selected tokens
end for
Add last-query anchor: s_t ← vote(t) + λ a_t(q_last^(ℓ,h))
Always retain token T and select top-C_{ℓ,h} − 1 tokens by s_t
Restore temporal order and return C'^(ℓ,h)
```

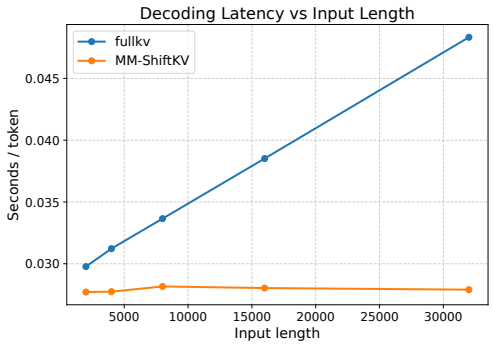
input length under long-context multimodal inputs.

Figure 6 provides additional evidence of the prefill–decode *representation scale mismatch* observed in Section 2. Specifically, Figure 6a visualizes layer-wise representation statistics on **DocVQA**, showing that decoding-stage hidden states exhibit consistently larger variance than those observed during prefilling, despite sharing identical model parameters.

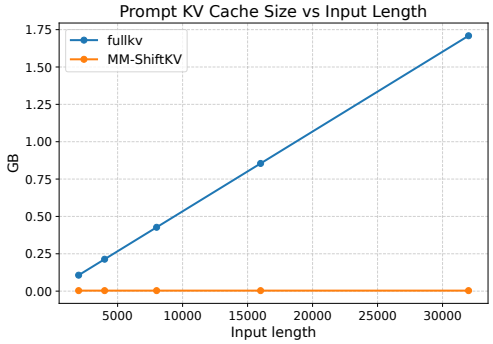
Figure 6b shows that the same variance expansion effect persists on **SynthDog**, indicating that the prefill–decode scale mismatch is not specific to a single dataset but instead reflects a systematic property of multimodal inference under long contexts. Together, these visualizations demonstrate that prefilling-stage statistics systematically underestimate the scale of decoding-time representations across different document understanding benchmarks.

D Sensitivity Study on Hyperparameters

We conduct a sensitivity study to examine the robustness of **MM-ShiftKV** with respect to its key hyperparameters. Unless otherwise specified, all experiments in this section are performed on **Qwen2.5-VL-7B-Instruct** under a fixed per-head KV cache budget of C=64. We report results on three representative multimodal benchmarks: **OCRBench** and **TextVQA** (accuracy), and **TextCaps** (CIDEr). When analyzing one hyperparameter, all others are held fixed at their default values (γ=10, N=512, G=32, τ=0.95, λ=1).



(a) Per-token decoding latency vs. input length.



(b) Prompt KV cache size vs. input length.

Figure 5: Visualization of decoding latency and prompt KV cache size under increasing input lengths.

D.1 Variance Expansion Factor γ

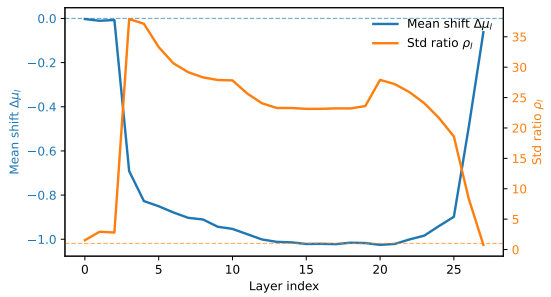
The variance expansion factor γ controls the scale calibration between prefilling-stage query proxies and decoding-time query distributions. As discussed in Section 2, decoding-time queries exhibit substantially larger variance than those observed during prefilling, motivating the use of $\gamma > 1$.

Table 6 shows that setting γ=1, corresponding to no variance expansion, leads to consistently degraded performance across all benchmarks. Increasing γ significantly improves performance, indicating that scale calibration is critical for effective KV importance estimation. Performance peaks around γ=10, while further increasing γ yields diminishing returns. Based on this observation, we fix γ=10 as the default value in all experiments.

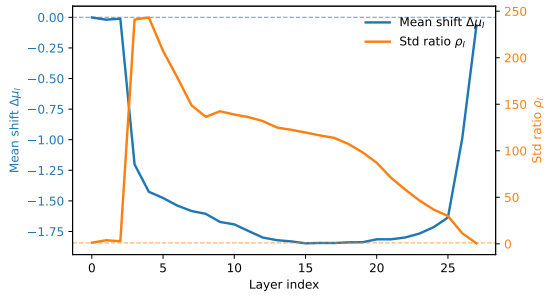
D.2 Attention Mass Threshold τ

The attention mass threshold τ determines the minimum cumulative attention mass preserved when selecting prompt KV tokens. A smaller threshold may discard occasionally important tokens, whereas an overly large threshold approaches FullKV behavior and weakens compression.

As shown in Table 7, τ=0.95 consistently achieves the best trade-off between performance



(a) DocVQA



(b) SynthDog

Figure 6: Additional visualizations of the prefill–decode *scale mismatch* in multimodal inference across different datasets. **(a)** Layer-wise representation statistics on **DocVQA**, showing that decoding-stage hidden states exhibit substantially larger variance than those observed during prefilling. **(b)** The same prefill–decode variance mismatch observed on **SynthDog**, indicating that the scale mismatch is consistent across datasets.

γ	OCRBench	TextVQA	TextCaps	Avg
1	55.7	66.5	51.8	58.0
2	56.1	69.2	55.1	60.1
5	65.4	73.5	58.6	65.8
10	**68.8**	**80.0**	**63.2**	**70.7**
20	68.5	79.3	60.1	69.3

Table 6: Sensitivity study on the variance expansion factor γ. OCRBench and TextVQA are evaluated with accuracy, and TextCaps with CIDEr. Avg denotes the macro-average over the three benchmarks. The per-head KV cache budget is C=64. Higher is better.

τ	OCRBench	TextVQA	TextCaps	Avg
0.90	62.4	78.1	42.4	61.0
0.95	**68.8**	**80.0**	**63.2**	**70.7**
0.99	66.7	79.2	49.0	65.0

Table 7: Sensitivity study on the attention mass threshold τ. Metrics and averaging follow Table 6. The per-head KV cache budget is C=64.

and compression across all evaluated benchmarks. Lower thresholds result in noticeable performance drops, while higher thresholds provide limited additional benefit. We therefore adopt τ=0.95 as a stable default setting.

D.3 Number of Query Proxies N

The number of query proxies N controls the quality of the Monte Carlo approximation of expected attention mass. Larger N reduces estimator variance but increases prefilling-stage computation.

Results in Table 8 show that performance improves as N increases from 128 to 512 and stabilizes at N=512. Using fewer proxies leads to noisier importance estimates, while larger values offer limited additional gains relative to the increased overhead. We therefore set N=512 as a balanced choice between accuracy and efficiency.

D.4 Summary

Overall, these sensitivity studies demonstrate that **MM-ShiftKV** is robust to moderate variations in its hyperparameters. The selected default values correspond to stable operating points that consistently balance accuracy, memory efficiency, and prefilling-stage overhead across OCR-centric, multimodal question answering, and image-conditioned generation tasks.

E Details of Baselines and Dateset

This appendix presents the implementation details and specific parameter configurations of the baselines, along with the detailed content and tasks of the datasets.

E.1 Details of Baselines

For our experiments, we use four methods, namely StreamingLLM, SnapKV, KeyDiff, and Exceptattention, as our test baselines. We also compare the performance differences between all these methods and FullKV under budget-constrained conditions.

StreamingLLM is a classic heuristic KV eviction method. It introduces the concept of attention sink, retains the initial KV pairs statically, and leverages a sliding window mechanism to continuously preserve the KV pairs within the most recent window during the decoding stage. This method discards a large number of redundant intermediate KV pairs. It achieves favorable performance in text reasoning due to its streaming inference paradigm. But in multimodal scenarios, it discards a large number of critical visual KV pairs, leading to performance collapse. In the experiments on StreamingLLM, we adopt the optimal parameters specified in its

N	OCRBench	TextVQA	TextCaps	Avg
128	63.1	78.6	49.8	63.8
256	63.5	78.9	47.0	63.1
512	**68.8**	**80.0**	**63.2**	**70.7**

Table 8: Sensitivity study on the number of query probes N. Metrics and averaging follow Table 6. The per-head KV cache budget is C=64.

original paper: to retain 4 attention sinks, set the window size to budgets-4, and to ensure fairness of comparative experiments, we do not perform KV eviction during the decoding stage.

SnapKV is a strong baseline method for large language models. It compares the similarity between the query attention scores of the final window in the prefilling stage and those in the decoding stage. It uses the final window of the prefilling stage to score the prefix KV pairs based on attention, selects the KV pairs with high attention scores, and statically retains the final window to maintain the characteristics of streaming inference. Following the optimal parameters provided by SnapKV, we set the window size to 32, the convolution size to 5, and the inter-group pooling to average pooling. SnapKV achieves outstanding performance on large language models and also has certain generality in multimodal scenarios, but it performs poorly under ultra-low budget conditions. We conduct a theoretical analysis of this issue: the statically retained window in SnapKV involves some waste and fails to truly evaluate the actually required queries, while reducing the window size leads to instability in KV selection.

KeyDiff achieves state-of-the-art metrics on large language models. It evaluates the relationship between the cosine similarity of Keys in large language models and the magnitude of the attention scores they receive. It proposes a query-agnostic method that scores Keys based on the cosine similarity between them, and the KV pairs with low cosine similarity are retained. In this experiment, since our dataset features streaming inference, we statically retain the last 1 token in accordance with KeyDiff's handling of streaming inference and for the fairness of the experiment. KeyDiff is also perturbed in multimodal scenarios. The relationship between the cosine similarity of Keys for tokens in multimodal scenarios and the attention scores is inconsistent with that in text-only unimodal scenarios, which is the main reason for the decrease in KeyDiff's accuracy in multimodal scenarios.

Exceptattention also leverages the regularity of numerical distributions, but it assumes the distribution homogeneity between the prefilling and decoding stages. However, in multimodal scenarios, there is a certain deviation between the distributions of these two stages, which results in relatively low performance. In this experiment, we set its sampling count to 512 to ensure fairness, and we also configure the static retention of 4 attention sinks, consistent with the setup in its original paper.

E.2 Details of Dataset

To comprehensively evaluate MM-ShiftKV , we utilize the lmms-eval (Zhang et al., 2024) evaluation framework. We employ OCRbench to assess OCR tasks and cross-modal text understanding tasks: this dataset covers multi-scenario images such as document scans and street view recognition, and we adopt accuracy as the corresponding evaluation metric. TextVQA, which spans real-world scenarios including restaurant menus and street signs, is used to evaluate image-text information question answering tasks, with performance measured by the Exact Match (El Maalouly, 2022) metric. Each image in TextCaps is annotated with 3-5 reference descriptions that contain key text; we leverage this dataset to assess semantically consistent text-image captioning tasks, with CIDEr (Vedantam et al., 2014) serving as the performance metric. ChartQA primarily composed of numerical charts is used to evaluate chart data understanding and reasoning-based question answering tasks, with performance assessed using relative error and the Exact Match metric. DocVQA covers a large volume of document images, and we use the **ANLS** (Peer et al., 2024) to evaluate MM-ShiftKV's performance on document image question answering tasks. MMMU is a dataset for calculation, geometric proof, and logical reasoning tasks. We use it to evaluate MM-ShiftKV's impact on reasoning capabilities and whether it generates hallucinations, with accuracy as the evaluation metric.

F Case Study

We present a qualitative case study on the **TextCaps** dataset using **LLaVA-v1.6-Vicuna-7B** under an **extreme KV-cache budget of 64 tokens per KV head**, to illustrate the behavior of different prefill-only KV selection methods under severe

Figure 7: In the left example, the image contains two people wearing **green shirts** with the printed text **"Bossa Nova"**. While the full KV model produces an accurate caption, several baselines degrade significantly after KV compression. **SnapKV** and **KeyDiff** identify the two people but omit the color attribute, and **StreamingLLM** further loses the shirt-related information. In contrast, **MM-ShiftKV** preserves both the color and the textual content, yielding a caption consistent with the reference.In the right example, the image shows a **Lone Star Beer can** with visible branding.Under the same tight per-head budget, **SnapKV** and **KeyDiff** misclassify the object as a beer bottle or miss the embedded text, whereas **MM-ShiftKV** correctly captures both the object type and textual content. These examples demonstrate that decode-aware KV selection enables MM-ShiftKV to remain robust even under extreme KV-cache compression, consistent with its quantitative gains on TextCaps.

memory constraints. Figure 7 shows two representative examples requiring accurate recognition of visual attributes and embedded text.Red denotes missing information, green indicates that key information is captured, and orange signifies that more information is captured compared to FullKV.