RIVET: REQUEST-INDEPENDENT VISION KV CACHE FOR ZERO-RECOMPUTATION VLM SERVING

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision-language model (VLM) serving engines redundantly compute and store an image's decoder key-value (KV) cache for every request, even when multiple requests share the same image. Existing solutions fail to fully eliminate redundancy: prefix cache methods reuse KV only when text and positions align exactly, while position-independent approaches still require per-request recomputation, preventing a single physical cache from serving multiple requests. We present Rivet, which builds a request-independent vision KV cache that a serving engine computes once and shares across requests with zero recomputation. For VLMs with multi-dimensional rotary position embeddings (MRoPE), Rivet removes the dependence of vision KV on the request with (1) canonical MRoPE pinning, which replaces the position-dependent offset with a fixed canonical offset, producing identical post-RoPE keys across requests, and (2) sink-preserving isolation that retains the attention sink while blocking context-specific preceding text, making vision-token values request-independent. Together, these techniques produce a vision-KV object that is identical for every request, so one physical copy serves an entire batch through page-table sharing. When a long text precedes the image, Rivet shifts the text positions instead of the image and preserves the accuracy of full recomputation. On Qwen3-VL at 2B, 4B, and 8B scales, Rivet achieves accuracy on par with or even surpassing partial-recomputation baselines such as VLCache and CacheBlend across VQAv2, ChartQA, and TextVQA, while delivering up to 7.4× time-to-first-token speedup over full recomputation and 1.4–2.3× over previous methods.

1 INTRODUCTION

Vision-language models (VLMs) (Alayrac et al., 2022; Liu et al., 2023; Li et al., 2023) frequently serve multiple requests about the same image in multi-turn conversations, retrieval-augmented generation pipelines (Lewis et al., 2020), and reranking workflows. Each request prefills the image's vision tokens through the decoder, and a representative image of 238 vision tokens yields decoder key-value (KV) entries of 26 MiB (Qwen3-VL-2B) to 33 MiB (Qwen3-VL-8B) in bf16. When B requests target the same image, the standard pipeline computes the vision KV B times and stores B identical copies, which causes both **computation redundancy** and **memory redundancy**.

Existing approaches mitigate but do not eliminate these redundancies. Exact-prefix caches such as vLLM's Automatic Prefix Caching (Kwon et al., 2023) and SGLang's RadixAttention (Zheng et al., 2024) reuse vision-KV entries only when the preceding token sequence and positions are identical. A different system prompt, conversation history, or image placement prevents reuse. Position-independent methods such as VLCache (Qin et al., 2025) and CacheBlend (Yao et al., 2025) relax this constraint by selectively recomputing or re-rotating cached entries per request. Even at a low recomputation ratio, the selected tokens pass through every decoder layer, and the resulting request-specific KV entries prevent a single physical copy from serving an entire batch, so both sources of redundancy persist. To our knowledge, **no prior work eliminates both sources of redundancy** by producing vision KV that is independent of the request so that a single copy serves different requests with zero recomputation.

We present Rivet, which makes the vision KV cache **request-independent**, so that a VLM serving engine computes it once per image and shares it across requests with zero recomputation. Rivet

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.

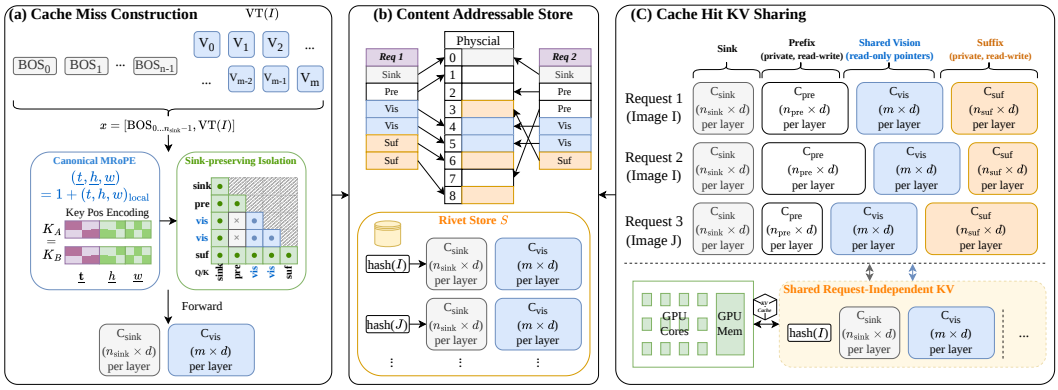

Figure 1: Rivet system overview. **Left:** offline construction computes the image's vision KV with a canonical MRoPE offset and sink-preserving isolation. **Center:** the image hash identifies the object within a prebuilt model-specific store. **Right:** each cache-hit request maps the same read-only KV slots into its page table. Request-specific text is prefilled separately.

targets VLMs with multi-dimensional rotary position embeddings (MRoPE). Rivet addresses two challenges. MRoPE assigns each vision token a **position-dependent** offset, so keys computed at different text positions carry different rotary phases, and causal attention makes vision-token hidden states **context-dependent**, which affects both keys and values. First, **canonical MRoPE pinning** replaces each position-dependent offset with a fixed canonical offset. This preserves every position difference within the image and shifts the distance from later text to the image by a constant that depends only on the text preceding the image. Second, **sink-preserving isolation** blocks vision tokens from attending to context-specific preceding text while retaining the attention sink. Vision tokens derive their representations primarily from attending to each other and to the sink token, so removing the remaining prefix has minimal effect on the resulting KV entries. Together, they produce a **request-independent vision-KV object** that Rivet computes once, indexes by image hash, and maps into the page table of every request, so all B requests share one physical copy without per-request recomputation or correction.

This design explains why zero recomputation can match methods that recompute part of the image. Partial recomputation repairs a request-specific approximation error of the unmodified model for every request. Rivet instead executes a slightly modified model exactly, so the splice introduces no approximation, and the departure of this model from the original depends only on the model and the preceding text (Section 4.4).

We evaluate Rivet on Qwen3-VL at the 2B, 4B, and 8B scales across VQAv2, ChartQA, and TextVQA, with serving benchmarks on an A100 GPU. Rivet remains within 0.8 points of full recomputation and matches or exceeds partial-recomputation baselines such as VLCache and CacheBlend, while delivering up to a 7.4× time-to-first-token speedup over full recomputation and 1.4–2.3× over VLCache and CacheBlend. Our contributions are:

- **A request-independent vision KV cache** for MRoPE VLMs, obtained through canonical MRoPE pinning and sink-preserving isolation, which lets one physical copy serve an entire batch with zero recomputation.
- **An analysis of why this object preserves accuracy**, which shows that the splice is exact, identifies the retained attention sink as the key factor, and traces the loss at long preceding text to the cross-modal shift, and shows that shifted text positions recover this accuracy without recomputation.
- **An end-to-end integration into a VLM serving engine** that combines offline construction, image-hash indexing, and page-table sharing, evaluated on Qwen3-VL at three scales for accuracy, latency, memory, prefix-length sensitivity, and component contributions.

2 Background

VLM prefill and the KV cache. A vision-language model encodes an image with a vision encoder, typically a Vision Transformer (Dosovitskiy et al., 2021), projects the resulting vision to-

kens into the decoder's embedding space, and combines them with text tokens to form $\mathbf{x} = [\mathbf{x}_{\text{pre}}, \mathbf{x}_{\text{vis}}, \mathbf{x}_{\text{suf}}]$. Here, $\mathbf{x}_{\text{pre}}$ contains the system prompt and preceding conversation, $\mathbf{x}_{\text{vis}}$ is the vision-token span, and $\mathbf{x}_{\text{suf}}$ contains the text after the image, including the user query. During prefill, the causal decoder (Vaswani et al., 2017) computes at each layer l the keys $\mathbf{K}_l = \text{RoPE}(\mathbf{W}_K^l \mathbf{h}_l, \mathbf{p})$ and values $\mathbf{V}_l = \mathbf{W}_V^l \mathbf{h}_l$ of every token and stores them in the **KV cache**, where $\mathbf{h}_l$ is the hidden state and $\mathbf{p}$ is the position vector. Values receive no direct positional transformation, but they remain indirectly position- and context-dependent because earlier attention operations determine $\mathbf{h}_l$.

MRoPE. Standard RoPE (Su et al., 2024) assigns a scalar position p to each token and rotates each pair of head dimensions by $p\theta_r$, where θ_r is the rotary frequency for pair r. Qwen3-VL (Bai et al., 2025) uses **interleaved multi-dimensional RoPE (MRoPE)**, which assigns three coordinates (t, h, w) to each token and associates the 64 rotary dimension pairs of each head with the temporal, height, and width axes (24, 20, and 20 pairs, respectively). Text tokens receive the same scalar index on all three axes. For vision token q in request i, Qwen3-VL assigns

$$\mathbf{p}_{\text{vis}}^i(q) = (\delta_i + \tau_q,\ \delta_i + h_q,\ \delta_i + w_q)\,, \tag{1}$$

where δ_i is a shared offset determined by the preceding layout and (τ_q, h_q, w_q) are local coordinates on the vision-token grid after spatial merging. For a still image, $\tau_q = 0$ for every token. Text after the image continues from one past the largest vision coordinate, so an image with an $H \times W$ merged grid advances the position index by only $\max(H, W)$ rather than by its token count.

Limitations of prefix caching. Production serving systems reuse cached KV only when a request matches a stored prefix exactly, in both content and position (Kwon et al., 2023; Zheng et al., 2024). Placing the same image after different text changes δ_i, which alters the rotary transformation of the vision keys, and changes what the vision tokens attend to, which alters their hidden states and therefore both keys and values. Matching image content alone is therefore insufficient for direct KV reuse.

3 Related Work

Prefix-only KV sharing. vLLM's Automatic Prefix Caching (Kwon et al., 2023), SGLang's RadixAttention (Zheng et al., 2024), TensorRT-LLM block reuse, and LMCache (Liu et al., 2025) share KV only for an unchanged leading prefix at the same positions. SmartCache (YU et al., 2025) extends reuse to semantically equivalent prefixes but still requires positional alignment, and multimodal image hashing in vLLM indexes images by content but still requires positional identity.

Position-independent reuse. For text, CacheBlend (Yao et al., 2025) and EPIC (Hu et al., 2025) selectively recompute a subset of tokens per request after position-independent splicing, and Block-Attention (Ma et al., 2025) requires fine-tuning. Prompt Cache (Gim et al., 2024) assigns fixed position IDs to reusable text segments, MEPIC (Wang et al., 2025) extends block-level reuse to page-aligned KV, and TurboRAG (Lu et al., 2025) and APE (Yang et al., 2025) assign reused chunks shared position ranges but require fine-tuning or attention rescaling, respectively. These methods target one-dimensional RoPE, and we adapt CacheBlend, the most general of them, to MRoPE vision tokens as a baseline. For multimodal models, MPIC (Zhao et al., 2025) reuses a cached image's KV at any position but recomputes all text tokens and the first k image tokens per request on one-dimensional RoPE models (Liu et al., 2023; Chen et al., 2024b). VLCache (Qin et al., 2025) stores vision KV before RoPE and selectively recomputes early vision tokens at the target positions, and Kamera (Ma et al., 2026) applies a per-reuse RoPE transformation and a low-rank conditioning update. Table 1 compares the incremental image-related work of these methods.

Vision–text relative distance. Several works show that the RoPE distance between vision and text tokens affects the accuracy and hallucination rate of VLMs, and they modify position assignment to control it (Xing et al., 2024; Ge et al., 2025; Wang et al., 2026). Canonical pinning changes this distance by a constant, and Section 4.4 relates the resulting behavior to these findings.

KV eviction (Zhang et al., 2023; Li et al., 2024), quantization (Liu et al., 2024), and vision-token pruning (Chen et al., 2024a) reduce single-request cost and are complementary, since the canonical object can be built from pruned tokens and compressed.

Table 1: Incremental image-related cost for a batch of B requests, excluding request-specific text prefill. L is the number of decoder layers, n_{vis} is the number of vision tokens, and m is the correction rank. The copy count assumes that request-specific transformed entries are materialized.

Method	Image-related work	Copies	MRoPE
MPIC	k token recomputations	B	No
EPIC	k token recomp./chunk	B	No
VLCache	2–5% recomp./layer	B	Yes
CacheBlend	Re-rotation + 16% recomp./layer	B	No†
Kamera	$O(BLn_{\text{vis}}dm)$ correction	B	Yes
Rivet	$O(Bn_{\text{vis}})$ **table entries**	**1**	**Yes**

†CacheBlend targets RAG document chunk reuse. Our evaluation adapts it to vision token reuse (Section 5.1).

4 METHOD

4.1 PROBLEM FORMULATION

Consider N requests $1, \ldots, N$ that contain the same image I after different text prefixes. The input to request i is $\mathbf{x}_i = [T^i_{\text{pre}},\ \text{ViT}(I),\ T^i_{\text{suf}}]$, where T^i_{pre} and T^i_{suf} denote the text tokens before and after the vision span, corresponding to $\mathbf{x}_{\text{pre}}$ and $\mathbf{x}_{\text{suf}}$ from Section 2. Standard prefill independently computes the vision-token keys $\mathbf{K}^{\text{vis}}_l(i) = \text{RoPE}(\mathbf{W}^l_K \mathbf{h}^{\text{vis}}_l(i), \mathbf{p}^i_{\text{vis}})$ and values $\mathbf{V}^{\text{vis}}_l(i) = \mathbf{W}^l_V \mathbf{h}^{\text{vis}}_l(i)$ at every layer l.

Error decomposition. Let $\mathbf{k}^i_l = \mathbf{W}^l_K \mathbf{h}^{\text{vis}}_l(i)$ denote the pre-RoPE key. The key difference $\Delta\mathbf{K}_l = \mathbf{K}^{\text{vis}}_l(i) - \mathbf{K}^{\text{vis}}_l(j)$ between two requests i and j decomposes as:

$$\Delta\mathbf{K}_l = \underbrace{\text{RoPE}\big(\mathbf{k}^i_l,\ \mathbf{p}^i_{\text{vis}}\big) - \text{RoPE}\Big(\mathbf{k}^i_l,\ \mathbf{p}^j_{\text{vis}}\Big)}_{\text{positional error: different offsets}} + \underbrace{\text{RoPE}\Big(\mathbf{k}^i_l,\ \mathbf{p}^j_{\text{vis}}\Big) - \text{RoPE}\Big(\mathbf{k}^j_l,\ \mathbf{p}^j_{\text{vis}}\Big)}_{\text{contextual error: different hidden states}} \tag{2}$$

The value difference $\Delta\mathbf{V}_l = \mathbf{W}^l_V(\mathbf{h}^{\text{vis}}_l(i) - \mathbf{h}^{\text{vis}}_l(j))$ is purely contextual, since values receive no rotary transformation. The positional error arises because δ_i depends on the length of the preceding text, so the same vision token receives a different rotary phase in each request. The contextual error arises because the causal mask allows vision tokens to attend to T^i_{pre}, which makes their hidden states, and therefore both keys and values, depend on the prefix. A physically shareable vision-KV object requires $\mathbf{K}^{\text{vis}}_l(i) = \mathbf{K}^{\text{vis}}_l(j)$ and $\mathbf{V}^{\text{vis}}_l(i) = \mathbf{V}^{\text{vis}}_l(j)$ for all i, j, and l in exact arithmetic, for a fixed model, preprocessing configuration, and image. Rivet obtains this property through two interventions, one for each error term. **Canonical MRoPE pinning** eliminates the positional error, and **sink-preserving isolation** eliminates the contextual error.

4.2 CANONICAL MROPE PINNING

As defined in Eq. 1, Qwen3-VL adds the position-dependent offset δ_i to all three local vision coordinates. **Canonical pinning** replaces δ_i with a fixed offset δ_c (Figure 2):

$$\mathbf{p}_{\text{canon}}(q) = (\delta_c + \tau_q,\ \delta_c + h_q,\ \delta_c + w_q)\,, \qquad \delta_c = 1. \tag{3}$$

The complete position vector is therefore independent of the request prefix, and every request applies the same rotary transformation to the same image.

Effect of canonical pinning. Because RoPE attention scores depend only on relative positions, the effect of pinning is determined by the relative offsets it changes. Pinning preserves every offset within the vision span, including the two-dimensional grid that encodes the image's spatial structure. Text tokens keep their natural positions (Section 4.5), so the relative offset from every later text token to every vision token grows by the same constant $\delta_i - \delta_c$ on all three axes (Table 2). We call this constant the **cross-modal shift**, which only depends on the text *before* the image.

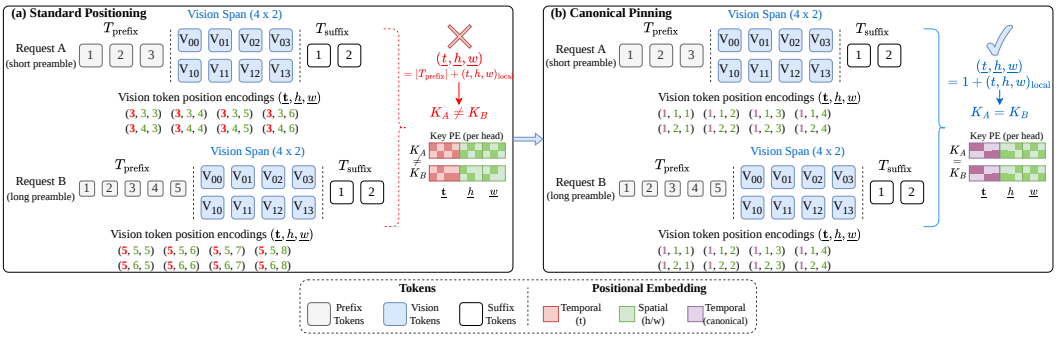

Figure 2: Canonical MRoPE pinning. (a) Standard positioning offsets the vision-token coordinates by the length of the preceding text, so the same image yields different keys after a short and a long preceding text. (b) Canonical pinning assigns every request the offset $\delta_c = 1$, so both requests yield identical keys.

Pinning also lets prefix and vision tokens share position indices, which is benign because vision queries attend only to the sink and to the image, exactly as in the canonical construction, and text queries depend only on relative offsets. Standard Qwen3-VL positioning already assigns overlapping indices, since an image of 238 tokens occupies only 17 distinct values per axis. Section 4.4 analyzes why these changes have little effect on accuracy.

4.3 Sink-Preserving Isolation

Canonical pinning does not remove the dependence of vision-token hidden states on preceding text. Full **isolation** removes this dependence by masking every key before the vision span $[v_0, v_1]$ for vision queries, which makes $\mathbf{h}_l^{\text{vis}}$ independent of T_{pre}. However, it reduces accuracy to 50.2% on 2B and 57.6% on 8B, because vision queries assign most of their attention mass to tokens before the vision span, with the largest share assigned to the attention sink (Xiao et al., 2024). **Sink-preserving isolation** retains the first n_{sink} tokens, including the attention sink, while blocking the remaining preceding text:

$$\mathbf{M}[s,u] = \begin{cases} 0 & \text{if } u < n_{\text{sink}} \text{ or } v_0 \le u \le s \\ -\infty & \text{otherwise} \end{cases} \tag{4}$$

for vision-query rows $s \in [v_0, v_1]$, where $\mathbf{M}$ is the additive attention mask. Text-query rows retain the standard causal mask, and the canonical construction and each target request share the same sink tokens. We use n_{sink}=4 by default. Exact reuse requires that the store construction and the request share these first n_{sink} tokens. In Qwen3-VL they are the three chat-template tokens that open the system turn and the first token of the system prompt, so a deployment either keeps the opening token of its system prompts fixed or uses n_{sink}=1, which performs comparably on 2B and 8B (Table 4).

4.4 Why Zero Recomputation Preserves Accuracy

The argument has two steps. Step 1 shows that reuse itself introduces no error, because pinning and isolation make the vision KV request-independent and the splice reproduces it exactly. Step 2 shows that this request-independent vision KV remains close to the vision KV of standard prefill, with the retained sink as the key factor.

Step 1: the splice is exact. Partial recomputation targets the vision KV of standard prefill and approximates it separately for every request by recomputing selected tokens, so its residual error depends on the request and on the token selection. Rivet instead fixes both the positions of the vision span and the tokens it attends to. Every input to the vision span is then identical across requests, so the spliced object equals the vision KV that recomputing the vision span with pinning and isolation produces for the target request. The first-token KL divergence between the two is about 10^{-3} (Table 4), the same magnitude as the mean discrepancy between two implementations of standard full recomputation (up to 4×10^{-3} between our Transformers and mini-SGLang backends). Accuracy therefore depends only on how far this vision KV departs from that of standard prefill,

Table 2: Position assignment when 15 text tokens precede an image with a 14×17 merged grid. Rivet keeps the text positions, so every offset from a suffix token to a vision token grows by the cross-modal shift $\delta_i - \delta_c = 14$ on all three axes. Shifted text positions subtract 14 from every text position and restore the standard offsets. Offsets within the image are unchanged in all three.

Span	**Standard**	**Rivet**	**Rivet w/ shifted text**
Prefix text (15 tokens)	$0, \ldots, 14$	$0, \ldots, 14$	$-14, \ldots, 0$
Vision token (h_q, w_q)	$(15,\ 15{+}h_q,\ 15{+}w_q)$	$(1,\ 1{+}h_q,\ 1{+}w_q)$	$(1,\ 1{+}h_q,\ 1{+}w_q)$
Suffix token j	$32 + j$	$32 + j$	$18 + j$
Offset from suffix j to vision q	$17{+}j - (0,\ h_q,\ w_q)$	$31{+}j - (0,\ h_q,\ w_q)$	$17{+}j - (0,\ h_q,\ w_q)$

which is determined by the model and the length of the preceding text, rather than by the individual request.

Step 2: the request-independent vision KV remains close to standard prefill. *(i) Isolation removes only an indirect attention path.* The question follows the image, so under the causal mask the vision tokens never attend to it, even in standard prefill. The preceding text reaches the answer along two paths, directly through the attention of later text tokens to the preceding text, and indirectly through the vision representations. Sink-preserving isolation removes only the indirect path. Later text tokens still attend to both the preceding text and the image, so the request context and the image are combined after the vision span, much as the decoder of an encoder–decoder model combines encoder outputs with its own context.

(ii) The retained sink is the key factor. For a vision query, let ϵ denote the attention mass that it assigns to preceding tokens other than the sink. Isolation renormalizes the remaining attention weights, so the attention output changes by $\epsilon(\boldsymbol{\mu}_{\mathrm{rm}} - \boldsymbol{\mu}_{\mathrm{kept}})$, where $\boldsymbol{\mu}_{\mathrm{rm}}$ and $\boldsymbol{\mu}_{\mathrm{kept}}$ are the attention-weighted mean values of the removed and the retained keys. This change is small when ϵ is small or when the removed values resemble the retained ones. Full isolation also removes the sink, which receives the largest share of the attention mass, and therefore perturbs every attention output substantially. Retaining only the first token raises the accuracy under isolation from 50.6% to 89.6% on 2B and from 58.6% to 93.2% on 8B (Table 4), which indicates that vision tokens rely on the sink as a fixed attention target rather than on the content of the request.

(iii) Pinning rotates the image as a whole. Because rotations compose, the natural positions of Eq. 1 apply to every vision query and key the same rotation $\mathbf{G}_{\delta_i}$ in addition to the rotation for its local coordinates. In attention among vision tokens, $\mathbf{G}_{\delta_i}$ acts on both the query and the key and cancels in their dot product, so pinning leaves these attention scores unchanged. For a text query, pinning rotates the entire set of vision keys by the same orthogonal transformation, which preserves their norms and pairwise similarities. The only change that text queries observe is therefore the cross-modal shift. A long preceding text enlarges this shift, and RoPE-based VLMs attenuate attention over long vision–text distances (Xing et al., 2024; Ge et al., 2025; Wang et al., 2026). Because rotary attention depends only on position differences, Rivet removes the shift by assigning every text token its standard position minus $\delta_i - \delta_c$ (**shifted text positions**, Table 2), which restores every text–text and text–vision distance of standard prefill without changing the stored object. On 8B with 1,000 and 2,000 preceding tokens, these positions match full recomputation, whereas standard text positions fall 2.4 to 3.5 points below it (Section 5.4). Appendix B discusses when to apply them.

4.5 Serving Protocol

Figure 1 illustrates the Rivet lifecycle, and Algorithm 1 in Appendix B details the serving protocol. **Store construction.** Rivet runs a fixed source template with the canonical positions of Eq. 3 and the mask of Eq. 4, extracts only the vision-token KV entries, and indexes them by a hash of the encoded image in a store specific to the model, processor, resize rule, and data type. On a cache miss, the engine constructs the object inline and inserts it for future reuse. **Cache hit.** The engine assigns text tokens the same positions as standard prefill and the vision span the fixed offset $\delta_c = 1$ (Table 2). It prefills the preceding text, maps the stored vision slots into the request's page table as read-only context, and prefills the suffix. The attention kernel reads the stored keys without an additional rotary transformation, which corresponds to `pos_encoding_mode=NONE` in FlashInfer (Ye et al., 2025). **Batched sharing.** Requests that reference the same image map their vision spans to the same

Table 3: Accuracy (Acc) and token-level F1 (%) on Qwen3-VL-2B, 4B, and 8B. Bold marks the best reuse method in each column for each model.

		VQAv2		ChartQA		TextVQA	
Model	**Method**	Acc	F1	Acc	F1	Acc	F1
2B	Full recomputation	86.7	87.2	79.5	79.9	81.2	81.4
	VLCache (r=0.05)	86.3	86.8	78.7	79.2	**81.3**	81.3
	CacheBlend (r=0.05)	86.5	87.0	**79.6**	**80.1**	80.9	81.3
	CacheBlend (r=0.16)	86.5	87.1	79.5	80.0	80.8	81.2
	Rivet (ours)	**86.8**	**87.4**	78.7	79.2	**81.3**	**81.5**
4B	Full recomputation	88.4	88.9	82.5	82.9	82.9	83.1
	VLCache (r=0.05)	88.3	88.7	81.8	82.2	**82.8**	**82.9**
	CacheBlend (r=0.05)	**88.9**	**89.3**	82.5	82.8	82.3	82.4
	CacheBlend (r=0.16)	88.7	89.2	82.4	82.8	82.3	82.4
	Rivet (ours)	88.2	88.7	**82.6**	**83.0**	82.6	82.6
8B	Full recomputation	89.5	89.8	84.6	85.0	85.5	85.5
	VLCache (r=0.05)	89.6	89.9	84.8	85.2	85.1	85.1
	CacheBlend (r=0.05)	89.7	90.0	84.8	85.2	**85.5**	**85.4**
	CacheBlend (r=0.16)	**89.9**	**90.1**	84.9	**85.4**	85.4	85.3
	Rivet (ours)	89.7	89.9	**85.0**	**85.4**	85.3	**85.4**

physical slots, analogous to shared-prefix batching (Juravsky et al., 2024). This requires $O(Bn_{\text{vis}})$ page-table entries for B requests but no KV copies. One object occupies $2Ln_{\text{vis}}H_{\text{kv}}db$ bytes for L layers, H_{kv} KV heads, head dimension d, and element size b, which is 26.0 MiB for Qwen3-VL-2B and 33.5 MiB for Qwen3-VL-8B with $n_{\text{vis}} = 238$ in bf16. Because the object does not depend on the request, the store can also be built offline for a known image corpus, such as the documents of a retrieval pipeline, and reused across serving sessions.

5 Evaluation

5.1 Setup

Models and environment. We evaluate Qwen3-VL-{2B, 4B, 8B}-Instruct (Bai et al., 2025), all of which use interleaved MRoPE and grouped-query attention (Ainslie et al., 2023), on one NVIDIA A100 40GB GPU. Accuracy and serving experiments use our Rivet integration in mini-SGLang (SGLang Team, 2025) with FlashInfer 0.6.14, greedy decoding, and n_{sink}=4. The prefix-length and ablation studies use Transformers 4.57.3 with bf16 SDPA attention on the 2B and 8B models, and the prefix-length study uses n_{sink}=1. Image preprocessing preserves aspect ratio and caps the pixel area at 512^2.

Benchmarks. We use 5,000 VQAv2 (Goyal et al., 2017) validation examples, the 2,500-example ChartQA (Masry et al., 2022) test set, and 5,000 TextVQA (Singh et al., 2019) validation examples, and report binary task accuracy and token-level F1 (Appendix A). Accuracy prompts use the default Qwen3-VL system prompt followed by the image and the question. The prefix-length and ablation studies insert longer preceding text and use a held-out VQAv2 subset.

Baselines. **Full recomputation** runs standard prefill for every request and provides the quality reference. **Canonical recomputation** applies canonical pinning and sink-preserving isolation but recomputes the vision span for every request. It computes exactly the vision KV that Rivet reuses (Section 4.4), and therefore separates the effect of pinning and isolation from that of the splice. **VLCache** (Qin et al., 2025) splices source KV and recomputes the prefix and the first $\lceil 0.05\, n_{\text{vis}} \rceil$ vision tokens at target positions. **CacheBlend** (Yao et al., 2025), adapted to MRoPE, re-rotates reused keys to the target positions, runs the first two layers in full, and recomputes a fraction r of the vision tokens at every later layer. We evaluate r=0.16, the ratio of its original evaluation, and r=0.05, which matches the VLCache budget. Appendix A gives implementation details. We do not evaluate MPIC (Zhao et al., 2025) separately because it recomputes the same tokens as VLCache.

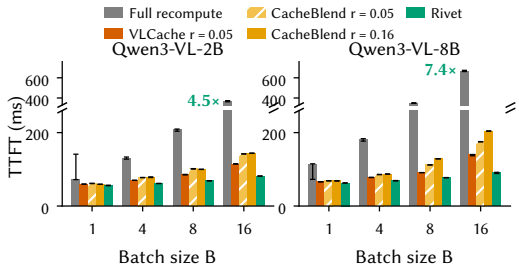

Figure 3: Time-to-first token latency. Bars report median time to first token over 10 measured runs, and error bars report the interquartile range.

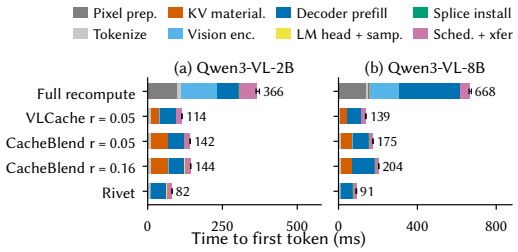

Figure 4: TTFT stage breakdown at B=16 (median over 10 runs, IQR error bars).

5.2 Accuracy

Table 3 reports accuracy and token-level F1, which follow the same trends. Rivet remains within 0.8 points of full recomputation across all nine settings. It matches or exceeds the VLCache baseline in seven of nine settings, despite performing no vision-token recomputation. Rivet also reproduces the answer of full recomputation on 90.1 to 96.7% of examples, more often than VLCache in all nine settings, and its mean first-token KL is 1.4 to 5.8× lower. Over 128 decoding steps of detailed captioning and DocVQA, Rivet lowers the mean teacher-forced KL relative to VLCache by 1.6 to 4.8× (Appendix E). CacheBlend also remains within one point across all settings, though its standalone chunk encoding produces slightly larger shortfalls on TextVQA at the smaller model sizes. These results show that pinning and isolation preserve accuracy across model scales and tasks, and Section 5.4 varies the preceding text.

Table 4: Ablation on a held-out VQAv2 subset (n=500). Δ is relative to full recomputation. Splice KL is the first-token KL divergence between the spliced request and recomputation under the same configuration, which tests whether the splice is exact.

	Qwen3-VL-2B			Qwen3-VL-8B		
Configuration	Acc	Δ	Splice KL	Acc	Δ	Splice KL
Full recomputation (reference)	89.4	–	–	93.8	–	–
Canonical only	88.0	−1.4	0.063	92.4	−1.4	0.082
Full isolation	50.2	−39.2	0.219	57.6	−36.2	0.500
Canonical + full isolation	50.6	−38.8	0.002	58.6	−35.2	0.002
Canonical + retain 1	89.6	+0.2	0.001	93.2	−0.6	0.001
Rivet (canonical + retain 4)	**89.8**	**+0.4**	**0.001**	**93.2**	**−0.6**	**0.001**

5.3 Serving Performance

Shared-image batch latency. Figure 3 reports the median time to first token (TTFT) over 10 measured runs after one warmup iteration, for batches of requests that query the same image. Rivet and the reuse baselines are measured on warm, GPU-resident cache hits, and full recomputation additionally includes image loading, preprocessing on eight CPU workers, and vision encoding. At $B = 16$, Rivet achieves a 4.5× speedup on 2B, from 366 to 82 ms, and a 7.4× speedup on 8B, from 668 to 91 ms. The selective-recompute baselines reach 3.2× (VLCache) and 2.5–2.6× (CacheBlend) on 2B and 4.8× and 3.3–3.8× on 8B, so Rivet is 1.4–2.3× faster.

TTFT composition. Figure 4 decomposes TTFT at B=16. All methods share a floor of tokenization, scheduling, and text prefill (35 ms on 2B, 49 ms on 8B). Partial-recomputation methods add two per-request stages. KV materialization copies or re-rotates the cached vision KV into each request's buffer (30–60 ms), and selective recomputation runs the selected tokens through all decoder layers (27–69 ms), for 31–111 ms of image-induced latency per batch. Rivet removes both stages. Its only image-induced cost is a store lookup and page-table installation for the whole batch, 0.78 ms on 2B and 0.63 ms on 8B, which is 343× and 901× less than that of full recomputation.

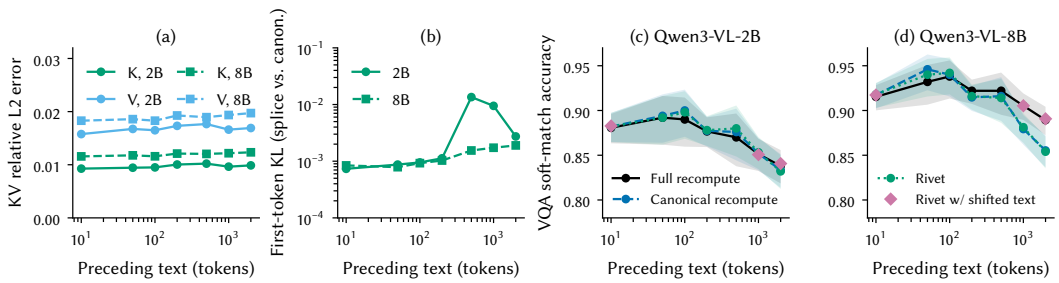

Figure 5: Sensitivity to the length of the preceding text on Qwen3-VL-2B and 8B. (a, b) KV error and first-token KL of Rivet relative to canonical recomputation. (c, d) Accuracy with 95% confidence intervals.

Memory. At B=16 with 238 vision tokens, full recomputation holds 16 vision-KV objects (417 MiB on 2B and 536 MiB on 8B in bf16). VLCache shares the unrecomputed portion as one physical object and stores the recomputed rows per request, totaling $1+rB = 1.8$ objects (47 and 60 MiB). CacheBlend re-rotates keys per request and holds recomputed rows separately, totaling 9.3 to 10.2 objects depending on the recomputation ratio (245–265 and 311–337 MiB). Rivet holds exactly one object (26 and 33 MiB) regardless of the batch size.

5.4 Sensitivity to the Preceding Text

Figure 5 places between 10 and 2,000 text tokens before the image, which varies the cross-modal shift analyzed in Section 4.4. Rivet remains within one point of canonical recomputation at every length, with KV errors below 0.02 and first-token KL of at most 1.4×10^{-2}, so the splice itself introduces negligible error. Relative to full recomputation, Rivet on 2B never falls more than 0.6 points below at any length. On 8B, Rivet with shifted text positions reaches 90.5% and 89.1% at 1,000 and 2,000 tokens, matching full recomputation (90.5% and 88.9%, Figure 5d). With standard text positions it falls to 85.4% at 2,000 tokens, as does canonical recomputation (85.5%), whereas isolation with standard positions reaches 89.2% (Appendix C), so this gap comes from the cross-modal shift rather than from the splice or from isolation.

5.5 Ablation Study

Table 4 ablates the two components of Rivet on 2B and 8B. Canonical pinning alone reduces the mean per-layer relative ℓ_2 key error between source and target from 0.43 to 0.25 on 2B and from 0.59 to 0.30 on 8B, but the remaining contextual error leaves a splice KL of 0.06 to 0.08. Full isolation removes the contextual error but drastically reduces accuracy, because vision tokens assign most of their attention mass to the preceding context, particularly the attention sink. Retaining the sink token recovers nearly all accuracy while keeping the splice KL at approximately 10^{-3}, and Rivet differs from full recomputation by at most 0.6 points. Reusing KV built after a short source template without pinning or isolation (the **uncorrected splice**) yields a splice KL of 0.12 on 2B and 0.35 on 8B, two orders of magnitude above that of Rivet, and falls 8.4 points below full recomputation on 8B with 2,000 preceding tokens (Appendix D).

6 Conclusion

We presented Rivet, a request-independent vision KV cache for zero-recomputation VLM serving. Rivet removes the position-dependent and context-dependent factors of vision KV, so that one object is computed once and shared verbatim across requests. Because the shared object is identical for every request, reuse introduces no per-request error, and shifted text positions preserve accuracy when a long text precedes the image. On Qwen3-VL at 2B, 4B, and 8B scales, Rivet matches partial-recomputation baselines in accuracy while delivering up to 7.4× time-to-first-token speedup over full recomputation and 1.4–2.3× over VLCache and CacheBlend.

AI USE STATEMENT

We used generative AI tools to write evaluation scripts, to assist in interpreting experimental results, and to edit the language of the paper. The authors reviewed and tested all AI-assisted code, checked every interpretation against the raw result files, and computed every reported number with scripts from those files. The authors take full responsibility for the content of this paper.

REPRODUCIBILITY STATEMENT

Section 5.1 specifies the models, hardware, software versions, decoding settings, image preprocessing, benchmark subsets, and baseline configurations, and Appendix A gives the scoring rules and implementation details. We will release the Rivet serving integration, the evaluation harness, the raw per-example result files, and the scripts that compute every table and figure from those files.

REFERENCES

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

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, Andrew Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karen Simonyan. Flamingo: a visual language model for few-shot learning. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=EbMuimAbPbs.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

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*, pp. 19–35. Springer, 2024a.

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 24185–24198, 2024b.

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

Junqi Ge, Ziyi Chen, Jintao Lin, Jinguo Zhu, Xihui Liu, Jifeng Dai, and Xizhou Zhu. V2PE: Improving multimodal long-context capability of vision-language models with variable visual position encoding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025.

In Gim, Guojun Chen, Seung-seob Lee, Nikhil Sarda, Anurag Khandelwal, and Lin Zhong. Prompt cache: Modular attention reuse for low-latency inference. In P. Gibbons, G. Pekhimenko, and C. De Sa (eds.), *Proceedings of Machine Learning and Systems*, volume 6, pp. 325–338, 2024. URL https://proceedings.mlsys.org/paper_files/paper/2024/file/a66caa1703fe34705a4368c3014c1966-Paper-Conference.pdf.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6325–6334, 2017. doi: 10.1109/CVPR.2017.670.

Junhao Hu, Wenrui Huang, Weidong Wang, Haoyi Wang, tiancheng hu, zhang qin, Hao Feng, Xusheng Chen, Yizhou Shan, and Tao Xie. EPIC: Efficient position-independent caching for serving large language models. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=qjd3ZUiHRT.

Jordan Juravsky, Bradley Brown, Ryan Saul Ehrlich, Daniel Y Fu, Christopher Re, and Azalia Mirhoseini. Hydragen: High-throughput LLM inference with shared prefixes. In *Workshop on Efficient Systems for Foundation Models II @ ICML2024*, 2024. URL https://openreview.net/forum?id=Ye3ia34Fpp.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, SOSP '23, pp. 611–626, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400702297. doi: 10.1145/3600006.3613165. URL https://doi.org/10.1145/3600006.3613165.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: bootstrapping language-image pre-training with frozen image encoders and large language models. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

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 *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NIPS '24, Red Hook, NY, USA, 2024. Curran Associates Inc. ISBN 9798331314385.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc.

Yuhan Liu, Yihua Cheng, Jiayi Yao, Yuwei An, Xiaokun Chen, Shaoting Feng, Yuyang Huang, Samuel Shen, Rui Zhang, Kuntai Du, et al. Lmcache: An efficient kv cache layer for enterprise-scale llm inference. *arXiv preprint arXiv:2510.09665*, 2025.

Zirui Liu, Jiayi Yuan, Hongye Jin, Shaochen (Henry) Zhong, Zhaozhuo Xu, Vladimir Braverman, Beidi Chen, and Xia Hu. Kivi: a tuning-free asymmetric 2bit quantization for kv cache. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

Songshuo Lu, Hua Wang, Yutian Rong, Zhi Chen, and Yaohua Tang. TurboRAG: Accelerating retrieval-augmented generation with precomputed KV caches for chunked text. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 6588–6601, 2025.

Bole Ma, Jan Eitzinger, Harald Koestler, and Gerhard Wellein. Kamera: Unified position-invariant multimodal kv cache for training-free reuse. *arXiv preprint arXiv:2606.23581*, 2026.

Dongyang Ma, Yan Wang, and Tian Lan. Block-attention for efficient prefilling. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=7zNYY1E2fq.

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 Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Findings of the Association for Computational Linguistics: ACL 2022*, pp. 2263–2279, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-acl.177. URL https://aclanthology.org/2022.findings-acl.177/.

Shengling Qin, Hao Yu, Chenxin Wu, Zheng Li, Yizhong Cao, Zhengyang Zhuge, Yuxin Zhou, Wentao Yao, Yi Zhang, Zhengheng Wang, et al. Vlcache: Computing 2% vision tokens and reusing 98% for vision-language inference. *arXiv preprint arXiv:2512.12977*, 2025.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pp. 2383–2392, 2016.

SGLang Team. Mini-SGLang: A compact implementation of SGLang. https://github.com/sgl-project/mini-sglang, 2025.

Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards vqa models that can read. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8317–8326, 2019.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Chengcheng Wang, Jianyuan Guo, Hongguang Li, Yuchuan Tian, Ying Nie, Chang Xu, and Kai Han. Circle-RoPE: Cone-like decoupled rotary positional embedding for vision-language models. In *Forty-third International Conference on Machine Learning*, 2026.

Qian Wang, Zahra Yousefijamarani, Morgan Lindsay Heisler, Rongzhi Gu, Bai Xiaolong, Shan Yizhou, Wei Zhang, Wang Lan, Ying Xiong, Yong Zhang, et al. Mepic: Memory efficient position independent caching for llm serving. *arXiv preprint arXiv:2512.16822*, 2025.

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

Yun Xing, Yiheng Li, Ivan Laptev, and Shijian Lu. Mitigating object hallucination via concentric causal attention. In *Advances in Neural Information Processing Systems*, 2024.

Xinyu Yang, Tianqi Chen, and Beidi Chen. APE: Faster and longer context-augmented generation via adaptive parallel encoding. In *International Conference on Learning Representations*, 2025.

Jiayi Yao, Hanchen Li, Yuhan Liu, Siddhant Ray, Yihua Cheng, Qizheng Zhang, Kuntai Du, Shan Lu, and Junchen Jiang. Cacheblend: Fast large language model serving for rag with cached knowledge fusion. In *Proceedings of the Twentieth European Conference on Computer Systems*, EuroSys '25, pp. 94–109, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 9798400711961. doi: 10.1145/3689031.3696098. URL https://doi.org/10.1145/3689031.3696098.

Zihao Ye, Lequn Chen, Ruihang Lai, Wuwei Lin, Yineng Zhang, Stephanie Wang, Tianqi Chen, Baris Kasikci, Vinod Grover, Arvind Krishnamurthy, and Luis Ceze. Flashinfer: Efficient and customizable attention engine for LLM inference serving. In *Eighth Conference on Machine Learning and Systems*, 2025. URL https://openreview.net/forum?id=RXPofAsL8F.

Chengye YU, Tianyu Wang, Zili Shao, and Song Jiang. Smartcache: Context-aware semantic cache for efficient multi-turn LLM inference. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=lMxuq0GNeC.

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

Shiju Zhao, Junhao Hu, Rongxiao Huang, Jiaqi Zheng, and Guihai Chen. Mpic: Position-independent multimodal context caching system for efficient mllm serving. *arXiv preprint arXiv:2502.01960*, 2025.

Lianmin Zheng, Liangsheng Yin, Zhiqiang Xie, Chuyue Sun, Jeff Huang, Cody Hao Yu, Shiyi Cao, Christos Kozyrakis, Ion Stoica, Joseph E. Gonzalez, Clark Barrett, and Ying Sheng. Sglang: efficient execution of structured language model programs. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NIPS '24, Red Hook, NY, USA, 2024. Curran Associates Inc. ISBN 9798331314385.

A Experimental Details

Scoring. We report binary task accuracy rather than the official fractional VQA score. For VQAv2 and TextVQA, a prediction is correct when at least one lowercased, alphanumeric-normalized gold answer equals or occurs within the normalized prediction. For ChartQA, a numeric prediction is correct within 5% relative error, and nonnumeric answers require normalized equality or a complete-token match. Token-level F1 is the harmonic mean of token precision and recall between the normalized prediction and a gold answer, maximized over gold answers (Rajpurkar et al., 2016), and a numeric ChartQA prediction within 5% relative error scores 1. Accuracy runs use a batch size of 16 (32 for the 4B model), and serving runs use batch sizes of $\{1, 4, 8, 16\}$. Vision-token counts vary by image, and memory calculations use a representative count of 238.

VLCache. Our implementation follows the uniform-ratio behavior of the released proof-of-concept. Post-RoPE source KV is spliced without an additional rotation, and the prefix and the first $\lceil 0.05\, n_{\text{vis}} \rceil$ vision tokens are recomputed at target positions.

CacheBlend. Our implementation adapts the semantics of the released artifact to MRoPE. The image is encoded as a standalone chunk, reused keys are re-rotated by the position delta (shared across the temporal, height, and width bands), and the tokens with the highest value deviation at layer 1 are recomputed at every subsequent layer. Its values are copied verbatim and shareable, but its re-rotated keys are request-specific whenever target positions differ.

Serving measurements. Rivet measurements are warm, GPU-resident cache hits that include request tokenization, page-table installation, text prefill, and first-token generation, and exclude image loading, pixel preprocessing, and offline object construction. For the partial-recomputation baselines, each request performs a materialization pass over the vision KV, which is a verbatim copy for VLCache and a key re-rotation plus a value copy for CacheBlend, followed by selective recomputation. For VLCache, this covers the first $\lceil r\, n_{\text{vis}} \rceil$ vision tokens through all layers. For CacheBlend, it covers two full layers over all vision tokens plus $r\, n_{\text{vis}}$ tokens on the remaining layers.

B Serving Protocol

Algorithm 1 summarizes the serving protocol of Section 4.5 together with the shifted text positions of Section 4.4. Shifted text positions depend on the total length of the text before the image, so a cached KV of that text serves only requests that share all of it rather than a common beginning such as the system prompt. The protocol therefore keeps standard text positions when the preceding text

Algorithm 1 Rivet Serving Protocol

Require: Image I, preceding text T_{pre}, subsequent text T_{suf}, store $\mathcal{S}$, length threshold L
Ensure: Generated response

$h \leftarrow \text{Hash}(I)$
if $h \notin \mathcal{S}$ **then**
  Construct canonical vision KV and insert into $\mathcal{S}[h]$
end if
$\mathbf{C}_{\text{vis}} \leftarrow \mathcal{S}[h]$
$\mathbf{p} \leftarrow \text{ComposePositions}(T_{\text{pre}}, I, T_{\text{suf}}, \delta_{\text{c}}{=}1)$
if $|T_{\text{pre}}| > L$ **then**
  Subtract $\delta_i - \delta_{\text{c}}$ from every text position in $\mathbf{p}$ {shifted text positions}
end if
$(\mathbf{p}_{\text{pre}}, \mathbf{p}_{\text{suf}}) \leftarrow$ text slices of $\mathbf{p}$
$\text{PageTable} \leftarrow \text{Prefill}(T_{\text{pre}}, \mathbf{p}_{\text{pre}})$
Install $\mathbf{C}_{\text{vis}}$ at the vision-span page-table entries
$\text{SuffixPrefill}(T_{\text{suf}},\ \mathbf{p}_{\text{suf}},\ \text{PageTable})$
return $\text{Decode}(\text{PageTable})$

has at most L tokens, where Rivet on 8B remains within 0.8 points of full recomputation up to 500 tokens, and assigns shifted positions beyond this threshold. Because the cross-modal shift depends only on the text before the image, multi-turn conversations that place the image in the first user turn keep standard positions as they grow.

C Shifted Text Positions

Table 5 repeats the prefix-length study of Section 5.4 with two additional configurations. Isolation with standard positions recomputes the vision span for every request with the mask of Eq. 4 but without pinning, so it measures the effect of isolation alone. Shifted text positions splice the same stored object as Rivet but place every text token at its standard position minus $\delta_i - \delta_{\text{c}}$. The text before the image then receives negative positions, which rotary embeddings define in the same way as positive ones, and every text–text and text–vision distance equals that of standard prefill. On both models, Rivet with shifted text positions remains within 0.3 points of full recomputation at every length. On 8B, Rivet with standard text positions falls 2.4 and 3.5 points below full recomputation at 1,000 and 2,000 tokens, while isolation with standard positions matches it, so this gap comes from the cross-modal shift. On 2B, Rivet with standard text positions remains within 0.6 points of full recomputation, and isolation with standard positions falls 0.8 and 1.1 points below it at 1,000 and 2,000 tokens, whereas the shifted positions, which splice an object built with the sink adjacent to the image, do not. At 1,000 and 2,000 tokens, shifting the text positions reduces the first-token KL relative to full recomputation from 0.68 and 0.46 to 0.12 and 0.07 on 2B, and from 0.62 and 0.61 to 0.07 and 0.09 on 8B.

Table 5: Accuracy (%) on the held-out VQAv2 subset by the number of text tokens before the image (n=1,500). All configurations retain one sink token, so Rivet here corresponds to canonical + retain 1 in Table 4.

	Qwen3-VL-2B			**Qwen3-VL-8B**		
Configuration	10	1,000	2,000	10	1,000	2,000
Full recomputation	88.1	85.1	83.8	91.5	90.5	88.9
Canonical recomputation	88.2	85.2	83.3	91.7	87.9	85.5
Isolation, standard positions	88.2	84.3	82.7	91.5	90.7	89.2
Rivet	88.2	85.3	83.2	91.8	88.1	85.4
Rivet w/ shifted text	88.3	85.1	84.1	91.7	90.5	89.1

D Ablation with Long Preceding Text

Table 6 repeats the 8B ablation of Table 4 with 2,000 text tokens before the image. The uncorrected splice falls 8.4 points below full recomputation with a splice KL of 1.73. Rivet with standard text positions falls 6.6 points below full recomputation on this subset, more than the 3.5 points measured on the 1,500-example subset of Section 5.4, while its splice KL remains near 10^{-3}, so this gap arises in the model itself rather than in the splice. With shifted text positions (Appendix C), Rivet reaches 90.6% on the same 500 examples, within 1.2 points of full recomputation (McNemar p=0.21), and its first-token KL relative to full recomputation falls from 0.73 to 0.09. With a single retained sink token, shifted text positions reach 90.4% (p=0.12).

Table 6: Ablation of Qwen3-VL-8B on the held-out VQAv2 subset (n=500) with 2,000 text tokens before the image. Columns follow Table 4. Rows with shifted text splice the same stored object as the corresponding row without them. No single prefill reproduces their positions, because the stored object was built with the sink next to the image, whereas shifted text positions keep the sink at its standard distance from the image, so their splice KL is not defined.

Configuration	Acc	Δ	Splice KL
Full recomputation (reference)	91.8	–	–
Uncorrected splice	83.4	−8.4	1.734
Canonical only	83.8	−8.0	0.249
Full isolation	48.4	−43.4	0.722
Canonical + full isolation	49.0	−42.8	0.004
Canonical + retain 1	85.2	−6.6	0.002
Rivet (canonical + retain 4)	85.2	−6.6	0.002
Canonical + retain 1 w/ shifted text	90.4	−1.4	–
Rivet w/ shifted text	**90.6**	**−1.2**	–

E Generation Divergence

Short-answer accuracy can conceal changes in the output distribution. For example, with 2,000 preceding tokens the 8B uncorrected splice has a first-token KL of 1.73, yet its task accuracy falls only to 83.4% (Table 6). We therefore evaluate teacher-forced fidelity over 128 fixed decoding steps on 250 detailed-captioning prompts and 250 DocVQA examples. This diagnostic uses Transformers with n_{sink}=1 and a short system prompt before the image, and compares Rivet with VLCache (r=0.05) and CacheBlend (r=0.16). The fixed-length diagnostic includes steps after an end-of-sequence token. For each example, full recomputation generates the reference continuation. We measure $D_{\text{KL}}(p_{\text{recompute}} \| p_{\text{method}})$ at each reference token and the number of initial positions whose greedy predictions match the reference.

Rivet yields a lower mean teacher-forced KL and a longer matching prefix than the VLCache baseline in all four settings (Figure 6). For captioning, the mean KL is 0.010 versus 0.031 at 2B and 0.014 versus 0.067 at 8B, corresponding to reductions by factors of 3.0 and 4.8. The mean matching-prefix lengths are 24.3 versus 14.9 tokens at 2B and 24.3 versus 11.7 tokens at 8B. For DocVQA, the mean KL is 0.013 versus 0.024 at 2B and 0.025 versus 0.041 at 8B, corresponding to factors of 1.8 and 1.6. These measurements indicate that Rivet more closely preserves the reference next-token distributions over extended decoding. The CacheBlend baseline attains the lowest KL in all four settings (0.006 to 0.011). This result is expected because it recomputes 16% of the vision tokens plus a full pass over the first two layers for every request, whereas Rivet recomputes nothing and shares one canonical copy across requests.

F Deployment Scope

Rivet targets requests with one reusable vision segment, because independently constructed objects contain no attention between images. Images that recur together, such as the pages of a document or the frames of a video clip, can be canonicalized jointly as one segment that preserves their mutual attention, which we leave to future work.

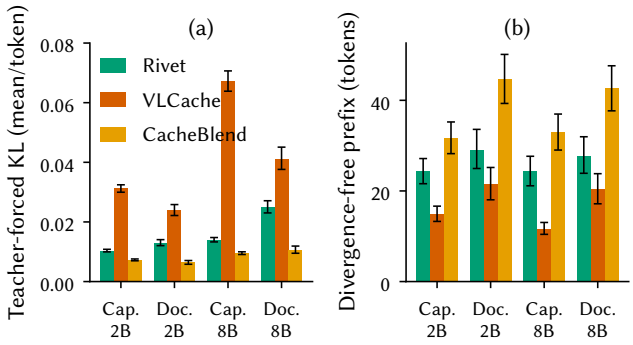

Figure 6: Teacher-forced fidelity over 128 fixed decoding steps. (a) Mean per-token $D_{\mathrm{KL}}(p_{\mathrm{recompute}} \| p_{\mathrm{method}})$ and (b) matching-prefix length, with bootstrap 95% intervals. Relative to the VLCache baseline, Rivet reduces mean KL by factors of 1.6 to 4.8. The CacheBlend baseline achieves the lowest divergence but incurs a per-request recomputation cost of 16% of the vision tokens plus two full layers.