PANORAMA: CROSS-MODEL KV SHARING VIA CONTEXTUAL RECOMPUTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Real-world LLM workloads, such as multi-agent systems and model cascades that escalate requests to larger models, frequently switch a request between models while retaining the same context. Because KV caches are tied to model parameters and cannot be shared across models, each switch forces the target model to re-prefill the entire context that the source model has already processed, incurring substantial computation cost and increasingly delaying time to first token as context length grows. Existing cross-model KV translation methods avoid this re-prefill by mapping the source KV cache into an approximate target KV cache, but translation alone can substantially degrade accuracy. Selective KV recomputation has been shown to repair context-inconsistent KV cache entries produced by the same model, but whether selective recomputation remains effective for a KV cache translated from a different model is unknown. Our key observation is that even when a translated KV cache is insufficient for direct decoding, the translated cache can still provide useful context for target-model recomputation. Based on this observation, we propose **PANORAMA**, a cross-model KV sharing method that recovers high-quality target inference from approximate cross-model KV without full target re-prefill. PANORAMA contextually recomputes a small subset of important target positions over the full KV cache produced by the cross-model mapper, allowing target-model computation to exploit the translated context. To provide the full translated context efficiently, PANORAMA factorizes the mapper into a shared source projection and layer-specific target reconstruction, transferring only a compact latent per prompt position. PANORAMA further refines the mapper under the repaired mixed cache used at inference, removing the training–inference discrepancy between translation-only mapper fitting and mixed-cache decoding. Across model pairs and tasks, PANORAMA achieves the highest accuracy among prior cross-model handoff methods on nearly all benchmarks at comparable handoff cost. Moreover, in the Qwen3 8B-to-32B switching scenario, our vLLM implementation of PANORAMA reduces handoff latency by 5.84× over native re-prefill at 16K tokens with minimal accuracy loss.

1 INTRODUCTION

Modern LLM workloads, such as multi-agent systems and model cascades, increasingly move requests across models while retaining the same conversation or document context (Ong et al., 2024; Wang et al., 2025; Jiang et al., 2026; Yue et al., 2024). This model switch introduces a new serving cost: KV caches are tied to model parameters, so the target model cannot directly reuse the cache already computed by the source model. The target model must therefore re-prefill the retained context before generating its first token, making the handoff cost grow with context length.

Existing cross-model KV sharing methods avoid full target re-prefill by translating the source KV cache into an approximate target KV cache (Heo et al., 2026; Qu et al., 2026). However, translation alone can substantially degrade downstream accuracy despite preserving the full prompt, as shown in Figure 1(a)–(b). Selective KV recomputation can repair context-inconsistent caches by recomputing only a subset of positions (Yao et al., 2024), but existing approaches operate on KV entries originally computed by the same model. Whether selective recomputation remains effective over a KV cache translated from a different model is therefore unclear.

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.

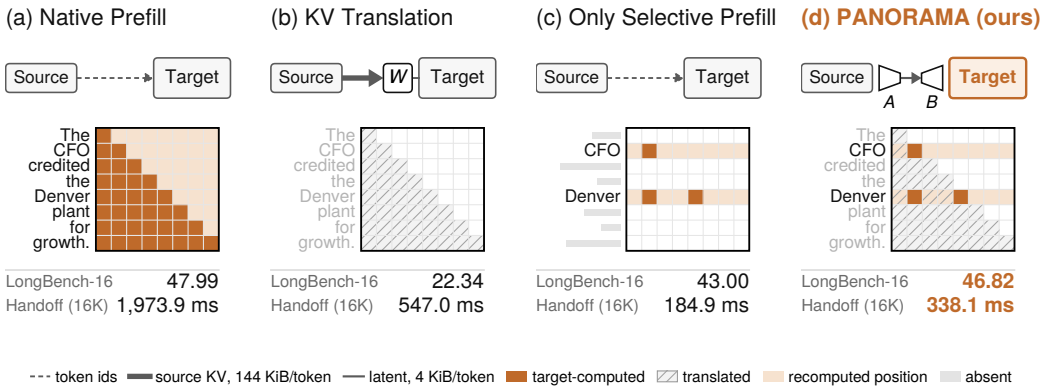

Figure 1: **Cross-model KV handoff strategies on Qwen3 8B→32B.** The top row shows the information transferred from the source to the target, and the middle row illustrates the prompt context visible to target-model computation. **(a) Native Prefill** recomputes the full prompt with the target model. **(b) KV Translation** preserves the full prompt through approximate translated KV, but suffers substantial quality degradation. **(c) Only Selective Prefill** computes target KV only at selected positions, discarding the remaining prompt context. **(d) PANORAMA** retains the full translated context while recomputing only selected target positions, recovering near-native quality at substantially lower handoff cost.

Our key observation is that even when a translated KV cache is insufficient for direct decoding, the translated KV cache can still provide useful context for target-model recomputation. As shown in Figure 1(d), recomputing selected target positions over the translated cache recovers much of the downstream accuracy lost by translation alone, and substantially outperforms recomputation without translated context.Cross-model translated KV therefore need not be accurate enough for direct decoding to remain useful as computational context.

Based on this observation, we propose PANORAMA, a cross-model KV sharing method that recovers high-quality target inference from approximate cross-model KV without full target re-prefill. PANORAMA contextually recomputes only a small subset of important target positions over the full translated KV cache. To provide the translated context efficiently, PANORAMA introduces Latent KV Transfer, which factorizes the mapper into a shared source projection and layer-specific target reconstruction and transfers only a compact latent per prompt position. PANORAMA further introduces Repair-Aware Refinement, which refines the mapper under the repaired mixed cache used during inference, removing the training–inference discrepancy of translation-only mapper fitting. PANORAMA reduces the transferred KV-state payload by $36\times$ relative to dense translation baselines and, with 10% target recomputation on Qwen3 8B-to-32B, reduces handoff latency by $5.84\times$ over native re-prefill at 16K tokens while retaining near-native quality. Across model pairs and tasks, PANORAMA achieves the highest accuracy among prior cross-model handoff methods on most benchmarks, providing a favorable quality–handoff-latency trade-off.

In summary, the main contributions are as follows:

- **Cross-model KV repair as a new empirical observation.** We empirically show that a translated KV cache can remain useful as computational context even when the translated cache is insufficient for direct decoding, enabling effective target-model selective recomputation using cross-model translated KV as context.
- **High-quality and efficient cross-model KV sharing.** We propose PANORAMA, which combines contextual target recomputation, compact latent transfer, and repair-aware refinement to recover high-quality target inference while substantially reducing the computation and communication cost of cross-model handoff.
- **Practical serving gains.** We implement PANORAMA in vLLM and evaluate the system across multiple model pairs and downstream tasks. PANORAMA achieves near-native quality while reducing the transferred KV-state payload by $36\times$ and handoff latency by $5.84\times$ over native re-prefill at 16K tokens.

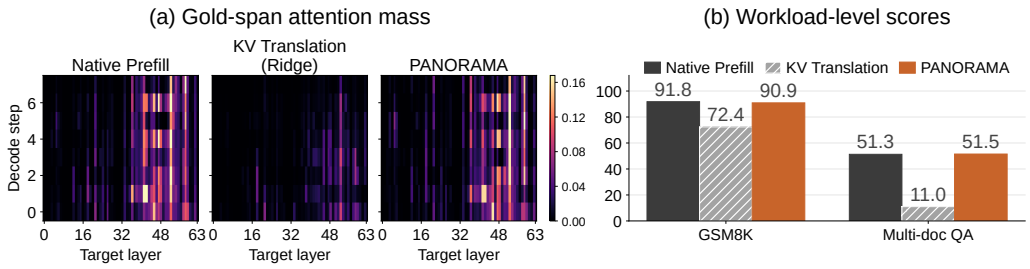

Figure 2: **Reading evidence after cross-model handoff. (a)** Decode-time gold-span attention under the same Native continuation on a 16k-token retrieval prompt. KV Translation (Ridge) weakens retrieval of the gold span, while PANORAMA recovers a Native-like attention pattern. **(b)** GSM8K and multi-document question-answering scores for the same methods on Qwen3 8B-to-32B.

2 Background and Motivation

2.1 Model-Specific KV Caches

LLMs generate tokens autoregressively, with each Transformer layer attending to preceding tokens through their key and value representations. To avoid repeatedly recomputing the prefix at every decoding step, LLM inference stores previously computed keys and values in a KV cache and reuses them throughout decoding.

We denote the source and target models by M_{src} and M_{tgt}, respectively, and omit layer and head indices unless needed. Given the hidden representation h_m of model m, the corresponding key and value are

$$(k_m, v_m) = (W_m^K h_m,\ W_m^V h_m), \qquad m \in \{\text{src}, \text{tgt}\}. \tag{1}$$

Since both h_m and the K/V projections W_m^K, W_m^V are model-specific, source KV cannot be directly consumed by the target model. A model switch therefore requires the target to re-prefill the retained context before decoding, duplicating prompt computation and increasing handoff cost with context length.

2.2 Cross-Model KV Translation

Cross-model KV translation avoids full target re-prefill by reconstructing target KV directly from the source cache (Heo et al., 2026; Qu et al., 2026). Although KV representations are model-specific, source and target KV computed at aligned prompt positions exhibit sufficient correspondence to learn a lightweight transformation between the two representation spaces. Existing methods process the same calibration sequences with both models to collect aligned source–target KV pairs and fit a mapper that predicts target KV from source KV.

We denote the key and value produced by the mapper as k_{map} and v_{map}, respectively. A general linear mapping can be written as

$$(k_{\text{map}}, v_{\text{map}}) = (M_K k_{\text{src}} + b_K,\ M_V v_{\text{src}} + b_V). \tag{2}$$

The mapper is fitted so that $(k_{\text{map}}, v_{\text{map}})$ approximates the native target KV $(k_{\text{tgt}}, v_{\text{tgt}})$. Ridge-based transfer fits the mappings in closed form, while CacheBridge introduces additional structural constraints and attention-aware calibration (Heo et al., 2026; Qu et al., 2026). For notational simplicity, k_{src} and v_{src} denote the source-side KV features consumed by the mapper; practical implementations may aggregate features from multiple source layers. At inference time, the source model processes the prompt once and the mapper reconstructs approximate target KV for every prompt position, replacing expensive target re-prefill with a substantially cheaper mapping operation while preserving the full prompt. However, preserving all prompt positions does not guarantee that the mapped cache retains the information required by the target model. Specifically, Figure 2(a) shows that, even under the same Native continuation, KV Translation (Ridge) substantially weakens

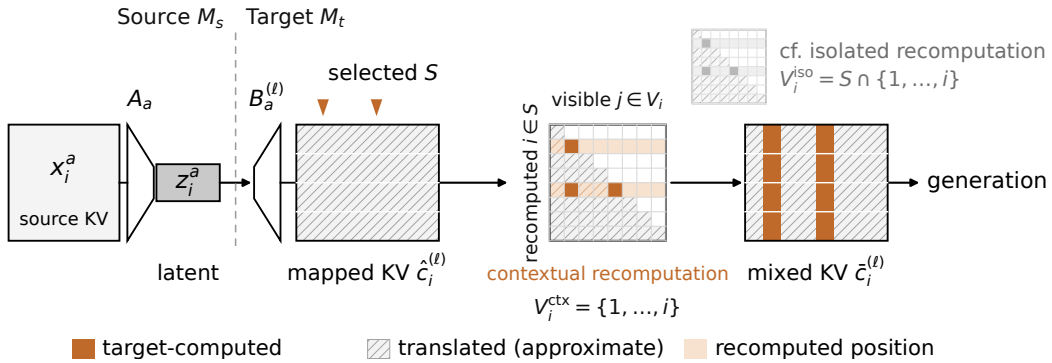

Figure 3: **PANORAMA inference workflow.** The source projects its cache through a shared projection A into compact K/V latents and transfers the latents to the target. Target-side expansion B reconstructs approximate entries for the full prompt. Contextual recomputation updates selected positions while allowing each selected query to attend to the reconstructed causal prefix while isolated recomputation would expose only selected causal positions. Target-computed entries replace the corresponding approximate entries, and generation proceeds from the resulting mixed cache.

attention to the gold span relative to Native Prefill, while PANORAMA recovers a Native-like attention pattern; see Appendix G for the attention measurement protocol.This indicates that translation primarily breaks target-compatible evidence access, while selective recomputation can restore that access without reconstructing the full native cache. This loss of effective context utilization is also reflected in a substantial degradation in downstream accuracy in Figure 2(b).

2.3 MOTIVATION: TRANSLATED KV AS COMPUTATIONAL CONTEXT

Selective KV recomputation provides another way to handle inaccurate cached representations. Prior work such as CacheBlend shows that context-inconsistent KV can be repaired by recomputing selected positions while retaining the remaining cached KV as context (Yao et al., 2024). Importantly, the retained KV in this setting was originally computed by the same high-quality model. The approximation arises from a context mismatch: the cached representations are valid model states, but were computed without the interactions introduced by the new context.

Cross-model translation presents a qualitatively different approximation. The mapped KV in Eq. 2 is not target KV computed under a stale context; instead, the mapped KV is reconstructed from representations produced by a different model and is never directly computed by the target model. It is therefore unclear whether selective repair remains effective in the cross-model setting.

We find that mapped KV can remain useful as computational context even when the mapped cache is insufficient for direct decoding. Figure 2 shows that PANORAMA recovers evidence-directed attention and downstream scores relative to translation alone; Table 2(a) separately tests the effect of letting selected positions read the mapped context. This observation relaxes the requirement on cross-model translation: the mapper need not reconstruct every native target KV accurately enough for direct decoding, but only preserve sufficient contextual information to support high-quality target recomputation.

3 DESIGN OF PANORAMA

PANORAMA is built on the observation that this weaker requirement can be satisfied: cross-model mapped KV can remain useful as computational context even when the same cache is insufficient for direct decoding. PANORAMA exploits this observation in three steps. First, contextual target recomputation constructs selected target states over the full mapped context in Section 3.1. Second, Latent KV Transfer makes the full mapped context inexpensive to communicate and reconstruct in Section 3.2. Finally, Repair-Aware Refinement adapts the mapper to the repaired cache composition actually consumed during inference in Section 3.3.

3.1 CONTEXTUAL TARGET RECOMPUTATION

Let S denote the prompt positions selected for target-model recomputation. PANORAMA recomputes only positions in S, while retaining the mapped KV at all remaining positions. When recomputing a selected position i, target attention jointly reads target-recomputed KV at previously repaired positions and mapped KV at the remaining causal positions:

$$o_i^r = \mathrm{Attn}\left(q_{\mathrm{tgt},i}, \left\{ \begin{cases} (k^r_{\mathrm{tgt},j}, v^r_{\mathrm{tgt},j}), & j \in S, \\ (k_{\mathrm{map},j}, v_{\mathrm{map},j}), & j \notin S \end{cases} \right\}_{j \le i}\right), \qquad i \in S. \tag{3}$$

Eq. 3 shows how PANORAMA repairs the mapped cache using the translated context itself. For each selected position, the target model computes a new $(k^r_{\mathrm{tgt},i}, v^r_{\mathrm{tgt},i})$ while attending to both previously recomputed target KV and the remaining mapped KV. The recomputed KV then replaces $(k_{\mathrm{map},i}, v_{\mathrm{map},i})$ at the corresponding position before subsequent target computation proceeds. Repeating this process layer by layer produces a repaired cache that combines target-computed KV at important positions with mapped KV over the rest of the prompt.

The mapped KV therefore plays a dual role in PANORAMA. At unrecomputed positions, $(k_{\mathrm{map}}, v_{\mathrm{map}})$ remains as approximate decoder state; at the same time, the mapped KV provides computational context for constructing the target-recomputed states $(k^r_{\mathrm{tgt}}, v^r_{\mathrm{tgt}})$. Importantly, the target-recomputed KV need not match the KV produced by native full prefill because the recomputed states are constructed over mapped context. PANORAMA instead relies on the weaker requirement identified in Section 2.3: the mapped cache only needs to preserve sufficient contextual information to support effective target-model recomputation.

3.2 LATENT KV TRANSFER

Contextual recomputation requires mapped KV to remain available for every prompt position. Directly applying the dense mapper in Eq. 2, however, introduces substantial mapping and communication overhead. In particular, providing mapped KV across all target layers either requires transmitting a large KV representation or repeatedly transforming high-dimensional source features.

PANORAMA introduces *Latent KV Transfer* (LKT) to make full-context translation efficient. LKT factorizes each dense mapping in Eq. 2 into a shared source projection followed by target-side reconstruction:

$$(c_K, c_V) = (A_K k_{\mathrm{src}},\ A_V v_{\mathrm{src}}), \qquad (k_{\mathrm{map}}, v_{\mathrm{map}}) = (B_K c_K + b_K,\ B_V c_V + b_V). \tag{4}$$

We denote the compact K/V latents as $c_K, c_V \in \mathbb{R}^r$ and collectively write $c = (c_K, c_V)$. Eq. 4 factorizes the dense mappings in Eq. 2 as $M_K \approx B_K A_K$ and $M_V \approx B_V A_V$. The projections A_K and A_V are shared across target layers, whereas each target layer uses its own reconstruction matrices B_K and B_V; target-layer indices are omitted for clarity. The same compact latent c can therefore be reused to reconstruct the mapped KV required by every target layer.

This shared latent directly reduces communication. Dense translation sends the layer-wise source KV cache, whereas LKT sends only $2r$ latent values per prompt token, independent of the number of target layers. With $r = 1024$ and BF16, the two K/V latents require only 4 KiB per token. In the Qwen3 8B-to-32B handoff, this reduces the transferred KV-state payload from 144 KiB/token for dense translation to 4 KiB/token, a $36\times$ reduction as detailed in Section 4.3.

To retain mapping quality under the rank-r bottleneck, we adopt alternating least squares (ALS) (Hastie et al., 2015) to jointly fit the shared projections A_K, A_V and layer-specific reconstructions B_K, B_V on paired source–target KV. ALS alternates least-squares updates between the shared and layer-specific factors, directly optimizing the factorized mapper rather than compressing a pre-fitted dense mapper post hoc. The factorized fitting objective is detailed in Appendix D, with calibration and optimization hyperparameters in Appendix A.1.

3.3 REPAIR-AWARE REFINEMENT

In prior approaches, the mapper is fitted to approximate native target KV (Heo et al., 2026; Qu et al., 2026). PANORAMA, however, decodes from a repaired mixed cache of mapped and target-recomputed KV, and the mapped KV also influences the target-recomputed states through Eq. 3. This creates a training–inference discrepancy that cannot be captured by pointwise KV reconstruction alone.PANORAMA addresses this training–inference discrepancy with *Repair-Aware Refinement* (RAR). For an unlabeled prompt, the source model and mapper first construct the mapped cache, and contextual recomputation then forms the same repaired cache composition used during inference. The target model performs a teacher-forced prefix pass over this cache. Let $p_{\mathrm{tgt},i}$ and $p_{\mathrm{mix},i}$ denote the target output distributions under native target prefill and the repaired mixed cache, respectively. RAR minimizes $\mathcal{L}_{\mathrm{RAR}} = \frac{1}{n}\sum_{i=1}^{n} D_{\mathrm{KL}}(p_{\mathrm{tgt},i} \| p_{\mathrm{mix},i})$.The source and target model parameters remain frozen, and gradients update only the cross-model mapper. Teacher forcing provides dense supervision across the prompt without requiring autoregressive rollouts. RAR therefore refines the mapper according to the target-model behavior induced by the repaired cache, rather than optimizing mapped KV only as standalone approximations of native target states. Consequently, RAR better aligns mapper training with inference-time cache usage and consistently improves transfer quality across the evaluated workloads, as shown in Section 4.3.

4 EVALUATION

4.1 EXPERIMENTAL SETUP

We evaluate PANORAMA from two perspectives: (1) quality after cross-model handoff and (2) handoff latency in our vLLM implementation. For quality, Qwen3-8B→32B is the primary pair and GLM-4-9B→GLM-4-32B provides a second model family (Yang et al., 2025; Zeng et al., 2024). Within each pair, we compare Native Prefill, which re-prefills the full prompt with the target model and serves as the quality reference, against the two existing kinds of handoff. Ridge translation (Heo et al., 2026) and CacheBridge (Qu et al., 2026) translate the full cache and let the target compute nothing; SpecPrefill (Liu et al., 2025) and SpecPC (Galim et al., 2026) let the target compute only the selected 10% of prompt tokens and keep nothing else. Unless stated otherwise, PANORAMA uses a rank-1024 latent (4 KiB/token), source-attention selection, contextual recomputation at $\rho = 10\%$, and Repair-Aware Refinement. Training and scoring details are given in Appendix A, and baseline reproduction is documented in Appendix C.

The benchmarks are chosen so that the two baseline families fail on different ones. On the sixteen LongBench tasks (LB-16) (Bai et al., 2023), a few evidence positions decide the answer, which favors methods that compute those positions exactly. On GSM8K (Cobbe et al., 2021), MATH-500, and eight commonsense reasoning tasks (CSR) (Zellers et al., 2019; Clark et al., 2019; Bisk et al., 2020; Sap et al., 2019; Sakaguchi et al., 2020; Clark et al., 2018; Mihaylov et al., 2018), the answer depends on content spread across a short prompt, which favors methods that keep every position. A method that keeps the full context and computes the selected positions should hold on both groups.

Latency setup. Handoff latency is measured from the moment the source cache is ready to the first nonempty target token. This window includes network transfer but excludes source prefill, online selection, source lookahead, and subsequent generation, all of which complete before the timed interval begins. Quality scores come from separate benchmark requests, whereas the latencies in Figure 4(a,b) are measured on a single 16k-token prompt. To ensure a fair comparison, all methods run on the same producer and consumer GPUs under the settings of Appendix E.

4.2 BENCHMARK RESULTS

Quality on the primary pair. Table 1 compares all methods on Qwen3-8B→32B. The results show that PANORAMA is the only handoff method that stays close to native on both benchmark groups, scoring 46.82 on LB-16 and 90.90% on GSM8K against 47.99 and 91.81% for Native Prefill. In contrast, Ridge holds 72.40% on GSM8K but drops to 22.34 on LB-16, because it keeps every position only approximately and loses the few positions a retrieval answer depends on; SpecPrefill shows the opposite, 43.00 on LB-16 but 50.80% on GSM8K, because it keeps the selected positions exactly and loses the rest of a prompt whose answer depends on all of it. We attribute the result to

Table 1: **Quality after cross-model KV handoff.** Avg. is the macro average over the sixteen LongBench tasks (LB-16); GSM8K, MATH-500, and CSR are accuracies (%). Native Prefill re-prefills the full prompt with the target model, whereas SpecPrefill, SpecPC, and PANORAMA let the target model compute only 10% of the prompt tokens. Bold marks the best handoff method within each pair.

		LongBench							Math		
Pair	**Method**	Single Doc.	Multi Doc.	**Summ.**	Few Shot	**Synth.**	**Code**	**Avg.**	**GSM8K**	**MATH-500**	**CSR**
Qwen3 8B→32B	Native Prefill	42.18	51.26	27.36	65.30	58.75	46.05	47.99	91.81	85.20	75.70
	Ridge	17.67	10.98	19.46	54.02	16.62	8.93	22.34	72.40	52.20	72.87
	CacheBridge	4.16	1.22	13.40	18.59	0.00	8.82	8.11	1.67	1.60	65.92
	SpecPrefill	40.98	49.49	23.76	62.49	**52.00**	26.95	43.00	50.80	38.80	63.94
	SpecPC	36.53	51.37	23.77	57.92	24.75	24.67	37.94	61.49	51.60	49.47
	PANORAMA	**42.13**	**51.45**	**26.71**	**66.04**	51.75	**43.33**	**46.82**	**90.90**	**82.00**	**73.65**
GLM-4 9B→32B	Native Prefill	42.37	50.75	26.34	71.60	56.50	49.82	49.11	92.87	88.40	78.43
	Ridge	8.13	5.74	16.43	37.03	7.01	12.83	15.10	26.69	25.60	73.52
	CacheBridge	6.10	2.94	14.79	9.33	2.87	11.17	7.97	4.17	3.60	70.40
	SpecPrefill	39.00	47.09	22.04	69.07	44.50	34.68	43.12	56.41	65.60	65.91
	SpecPC	37.31	**51.10**	23.20	68.60	43.50	33.01	43.35	43.52	57.40	50.43
	PANORAMA	**39.67**	42.26	**24.18**	**70.22**	**46.50**	**44.17**	**44.39**	**81.88**	**84.80**	**74.14**

Table 2: **Qwen3-8B→32B ablations.** The target model computes 10% of the prompt tokens in every row. (a) Context available to the selected positions; the italic Ridge row is a dense-mapper control. (b) Cache used for refinement, with contextual recomputation at inference; the last row is RAR. (c) Latent rank; each mapper is fitted independently, so the rank-1024 row is a separate fit from the mapper in (b). Bold marks the maximum in (b) and (c).

	Configuration	**GSM8K**	**LB-16**
(a)	Selective prefill only (SpecPrefill)	50.80	43.00
	Mapped cache only (compact mapper)	25.85	12.03
	Mapped cache, isolated recomputation	61.64	43.83
	Mapped cache, contextual recomputation	85.06	44.33
	Dense Ridge, contextual recomputation (control)	*90.37*	*45.21*
(b)	No refinement	85.06	44.33
	Refinement on the mapped cache	90.52	46.13
	Refinement with native replacement	89.54	46.29
	RAR: refinement on the recomputed mixed cache	**90.90**	**46.82**
(c)	Rank $r = 256$ (1 KiB/token, 144× vs. source KV)	79.68	45.59
	Rank $r = 512$ (2 KiB/token, 72×)	85.82	46.03
	Rank $r = 1024$ (4 KiB/token, 36×)	90.83	46.90
	Rank $r = 2048$ (8 KiB/token, 18×)	**91.43**	**47.33**

the dual role of the mapped KV in Section 3.1: the unselected positions remain available as decoder state, while the selected positions are rebuilt by the target over that context. The same holds on MATH-500, where PANORAMA leads the handoff methods at 82.00%. The gain is smallest where translation already works: on CSR, Ridge is within 3 points of native and PANORAMA adds 0.78. CacheBridge's collapse on this pair is not a reproduction error; on its own Qwen3-14B→32B pair our implementation reaches 98.5% of native CSR (Appendix C). Additionally, with the stronger Qwen3-14B source (Appendix I), Ridge alone is already competitive on the short tasks and stays slightly ahead on MATH-500 and CSR, while PANORAMA keeps its lead on LB-16 at 46.63.

Second model family. We further evaluate GLM-4-9B→32B under the same protocol (Table 1). The same pattern holds with a larger gap to native: PANORAMA scores 44.39 on LB-16 and 81.88% on GSM8K against 15.10 and 26.69% for Ridge and 49.11 and 92.87% for Native Prefill. The gap is wider than on Qwen3 because translation itself is weaker on this pair, so less of the prompt survives as usable context. This pair also exposes the one category where keeping the mapped cache costs accuracy. On multi-document QA, PANORAMA's 42.26 trails SpecPrefill's 47.09 and SpecPC's 51.10, both of which keep only the selected positions; when translation is this weak, the mapped cache held as decoder state hurts on the category that depends most on exact evidence positions.

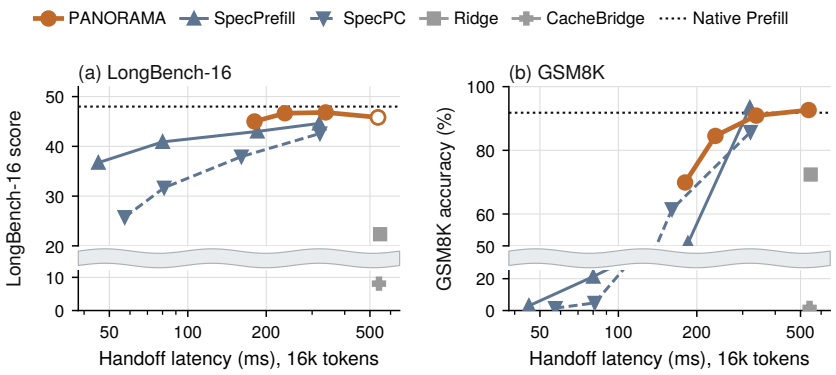

Figure 4: **Quality–latency frontier on Qwen3-8B→32B.** LB-16 and GSM8K scores versus 16k-token handoff latency; SpecPrefill, SpecPC, and PANORAMA are shown at 2.5%, 5%, 10%, and 20% recomputation from left to right. Hollow markers are points that a cheaper budget of the same method also beats in quality, and the wavy bands mark omitted score ranges.

4.3 ABLATION STUDIES

Table 2 examines the three components of Section 3 individually on Qwen3-8B→32B, with the target computing 10% of the prompt tokens in every row. Every row of panel (b) uses contextual recomputation at inference, so its first row is the same configuration as the contextual-recomputation row of panel (a).

Contextual recomputation. Table 2(a) compares the context available to the selected positions. The results show that the mapped cache is not usable for decoding on its own but is usable as context: the mapped cache alone scores 25.85% on GSM8K, recomputing the selected positions over it reaches 85.06%, and the same recomputation without reading it reaches 61.64%. We attribute the 23-point difference to the mapped context alone, since the two recomputation rows differ only in what the selected positions attend to; this is the weaker requirement of Section 2.3 measured directly. The same effect appears with the dense Ridge mapper, which rises from 72.40% to 90.37% under contextual recomputation, so the effect is a property of the mapped cache rather than of the compact mapper. On LB-16, in contrast, most of the gain over selective prefill comes from keeping the mapped cache as decoder state (43.00 to 43.83), and reading it during recomputation adds only 0.50, because on long prompts the selected positions already contain the evidence the answer depends on. Additionally, the selected states themselves move with the score: their K/V error relative to native target states is lower under contextual than under isolated recomputation at every prompt length (Appendix F).

Repair-Aware Refinement. Table 2(b) compares the caches on which the mapper is refined. RAR raises GSM8K from 85.06% to 90.90% and LB-16 from 44.33 to 46.82, closing most of the remaining gap to native. Refinement on the other two caches recovers most of the same gain, within 1.4 points on GSM8K and 0.7 on LB-16. We attribute the gain to fitting the mapper through the target model's output distribution, as intended in Section 3.3, rather than to the choice of training cache; RAR remains the highest of the three on both benchmarks, by a small margin. After refinement, the 4-KiB/token mapper slightly exceeds unrefined dense Ridge with contextual recomputation, which sends 144 KiB/token; Ridge was not refined under the same objective, so this is a comparison between two mappers at different payloads rather than a matched control.

Latent KV Transfer. Table 2(c) measures the quality cost of the rank-r bottleneck in Section 3.2 with the selector and refinement fixed. Rank 1024 is where the payload–quality curve flattens: raising the rank from 256 to 1024 lifts GSM8K by 11 points and LB-16 by 1.3, whereas doubling the payload again to rank 2048 adds 0.60 and 0.43. Consequently, PANORAMA sends 4 KiB/token, 36× less than the source KV cache, at a cost of about one point against native on each benchmark.

4.4 HANDOFF LATENCY

Quality at a given latency. PANORAMA reaches every LB-16 quality that selective prefill reaches at lower handoff latency, as measured on a 16k-token prompt across budgets from 2.5% to 20%

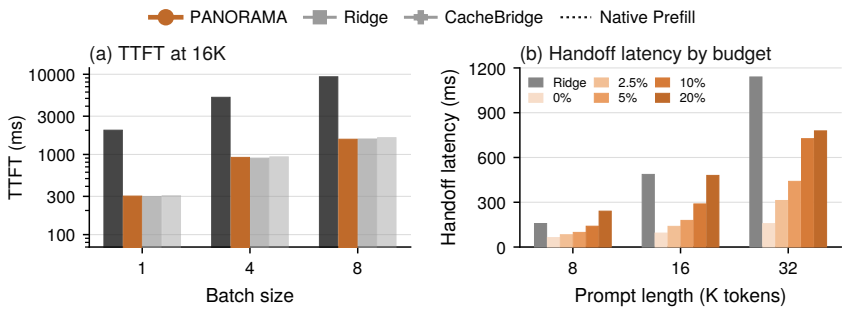

Figure 5: **Serving scaling on Qwen3-8B→32B. (a)** Request-level TTFT (log scale) at a fixed 16K-token prompt across batch sizes 1, 4, and 8; the dark bars are Native Prefill. **(b)** Handoff latency of Section 4.1 at 8, 16, and 32K tokens for Ridge and PANORAMA with 0–20% recomputation; the 0% bars transfer and reconstruct the mapped cache without recomputing any position.

(Figure 4(a)). A key advantage of PANORAMA is that the mapped cache lets a smaller recomputation budget do the work of a larger one. For example, its 2.5% point, 45.00 in 180.3 ms, already exceeds SpecPrefill's best score, 44.61, which needs the 20% budget and 319.5 ms. In contrast, Ridge and CacheBridge remain at 22.34 and 8.11 at about 545 ms, because they move the full cache and still cannot decode from it. The frontier has one exception, on GSM8K rather than LB-16: at 20%, SpecPrefill reaches 93.40% in 319.5 ms, ahead of PANORAMA's 92.65% in 537.1 ms (Figure 4(b)), since at that budget the selected positions already cover the short prompt and the mapped cache adds transfer time without adding accuracy.

Batching. Figure 5(a) compares request-level TTFT at batch sizes 1, 4, and 8. The reduction over native prefill does not shrink with batch size: PANORAMA is 6.0× faster than native prefill at batch 8, within the 5.6–6.6× range measured at batch sizes 1 and 4, because both costs scale with the batch and the 10% recomputation stays a small share of the target's work. At every batch size PANORAMA is also at or below Ridge and within 2.2% of CacheBridge, both of which translate the cache without recomputing any position, so the recomputation costs no more than translation alone.

Prompt length and recomputation budget. Figure 5(b) separates the two parts of the handoff at 8, 16, and 32k tokens. PANORAMA's cost grows with the recomputation budget, whereas Ridge's grows with the size of the cache it moves, so the point at which recomputation pays for itself shifts with prompt length. At the default 10% budget, PANORAMA is below Ridge at every length, by 11% at 8k tokens and by more than a third at 16k and 32k (726.0 vs. 1,139.5 ms). At 20%, the added recomputation costs more than the transfer it saves at 8k, breaks even at 16k, and is cheaper at 32k. The transfer itself is not what grows: the 0% bars, which transfer and reconstruct the mapped cache without recomputing any position, rise slowly with length, and the rest of each bar is recomputation. Source prefill and online selection are outside this measurement.

5 CONCLUSION

In this paper, we presented PANORAMA, an efficient cross-model KV sharing method that reduces the need for full target re-prefill when requests move across models.PANORAMA is motivated by the observation that a translated KV cache can still provide useful context for target-model recomputation, even when translation alone degrades downstream accuracy.By recomputing only a small subset of target positions over the full translated context, PANORAMA recovers much of the quality lost by translation without re-prefilling the full prompt.To make this procedure efficient, PANORAMA factorizes the cross-model mapper so that only a compact latent is transferred and reconstructed into the full mapped context at the target. It further refines the mapper under the repaired mixed cache used during inference, aligning mapper training with how the translated cache is actually consumed. Across model families and a range of long-context and reasoning tasks, PANORAMA achieves quality close to native prefill while reducing the transferred KV-state payload by 36×. Our vLLM implementation further reduces handoff latency by 5.84× over native re-prefill at 16K tokens, with similar trends under batching and longer prompt lengths. Overall, these results suggest that approximate cross-model KV can be effectively reused as context for selective target recomputation, providing an efficient alternative to full target re-prefill in multi-model serving.

REFERENCES

Yushi Bai et al. LongBench: A Bilingual, Multitask Benchmark for Long Context Understanding. *arXiv preprint arXiv:2308.14508*, 2023.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. PIQA: Reasoning about Physical Commonsense in Natural Language. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. BoolQ: Exploring the Surprising Difficulty of Natural Yes/No Questions. In *Proceedings of NAACL-HLT*, 2019.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think You Have Solved Question Answering? Try ARC, the AI2 Reasoning Challenge. *arXiv preprint arXiv:1803.05457*, 2018.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Rei Nakano, Christopher Hesse, and John Schulman. Training Verifiers to Solve Math Word Problems. *arXiv preprint arXiv:2110.14168*, 2021.

Kevin Galim, Ethan Ewer, Wonjun Kang, Minjae Lee, Hyung Il Koo, and Kangwook Lee. Draft-based Approximate Inference for LLMs. In *International Conference on Learning Representations*, 2026.

Trevor Hastie, Rahul Mazumder, Jason D. Lee, and Reza Zadeh. Matrix completion and low-rank svd via fast alternating least squares. *Journal of Machine Learning Research*, 16(104):3367–3402, 2015. URL `https://jmlr.org/papers/v16/hastie15a.html`.

Taekyung Heo, Rasoul Shafipour, Ritchie Zhao, Maximilian Golub, Mohammad Mahdi Kamani, Ritika Borkar, Makesh Tarun Chandran, Pantea Zardoshti, and Bita Darvish Rouhani. Cross-Model KV Cache Transfer in LLM Families: A Closed-Form Linear Mapping for Prefill Reuse. *arXiv preprint arXiv:2608.03893*, 2026.

Youhe Jiang, Fangcheng Fu, Wanru Zhao, Stephan Rabanser, Jintao Zhang, Nicholas D. Lane, and Binhang Yuan. Cascadia: An efficient cascade serving system for large language models. In *International Conference on Learning Representations*, 2026.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient Memory Management for Large Language Model Serving with PagedAttention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023.

Jingyu Liu, Beidi Chen, and Ce Zhang. Speculative Prefill: Turbocharging TTFT with Lightweight and Training-Free Token Importance Estimation. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 38188–38209. PMLR, 2025.

Yuhan Liu, Yuyang Huang, Jiayi Yao, Shaoting Feng, Zhuohan Gu, Kuntai Du, Hanchen Li, Yihua Cheng, Junchen Jiang, Shan Lu, Madan Musuvathi, and Esha Choukse. DroidSpeak: KV Cache Sharing Across Fine-tuned Model Variants. In *23rd USENIX Symposium on Networked Systems Design and Implementation*, 2026.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a Suit of Armor Conduct Electricity? A New Dataset for Open Book Question Answering. In *Proceedings of EMNLP*, 2018.

Isaac Ong, Amjad Almahairi, Vincent Wu, Wei-Lin Chiang, Tianhao Wu, Joseph E. Gonzalez, M. Waleed Kadous, and Ion Stoica. RouteLLM: Learning to Route LLMs with Preference Data. *arXiv preprint arXiv:2406.18665*, 2024.

Xingyu Qu, Siyuan Lu, Zhiyu Chen, Sheng Wang, and Tao Lin. CacheBridge: Efficient Cross-Model KV Cache Transfer. *arXiv preprint arXiv:2609.00891*, 2026.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. WinoGrande: An Adversarial Winograd Schema Challenge at Scale. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. Social IQa: Commonsense Reasoning about Social Interactions. In *Proceedings of EMNLP-IJCNLP*, 2019.

Junlin Wang, Jue Wang, Ben Athiwaratkun, Ce Zhang, and James Zou. Mixture-of-agents enhances large language model capabilities. In *International Conference on Learning Representations*, 2025.

Sunghyeon Woo, Jaeeun Kil, Hoseung Kim, Minsub Kim, Joonghoon Kim, Ahreum Seo, Sungjae Lee, Minjung Jo, Jiwon Ryu, Baeseong Park, Se Jung Kwon, and Dongsoo Lee. ICaRus: Identical Cache Reuse for Efficient Multi-Model Inference. In *International Conference on Learning Representations*, 2026a.

Sunghyeon Woo, Hoseung Kim, Sunghwan Shim, Minjung Jo, Hyunjoon Jeong, Jeongtae Lee, Joonghoon Kim, Sungjae Lee, Baeseong Park, Se Jung Kwon, and Dongsoo Lee. PrefillShare: A Shared Prefill Module for KV Reuse in Multi-LLM Disaggregated Serving. *arXiv preprint arXiv:2602.12029*, 2026b.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, Ahmed Hassan Awadallah, Ryen W. White, Doug Burger, and Chi Wang. AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation. *arXiv preprint arXiv:2308.08155*, 2023.

An Yang et al. Qwen3 Technical Report. *arXiv preprint arXiv:2505.09388*, 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. *arXiv preprint arXiv:2405.16444*, 2024.

Murong Yue, Jie Zhao, Min Zhang, Liang Du, and Ziyu Yao. Large language model cascades with mixture of thought representations for cost-efficient reasoning. In *The Twelfth International Conference on Learning Representations*, 2024.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a Machine Really Finish Your Sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019.

Aohan Zeng et al. ChatGLM: A Family of Large Language Models from GLM-130B to GLM-4 All Tools. *arXiv preprint arXiv:2406.12793*, 2024.

Yinmin Zhong, Shengyu Liu, Junda Chen, Jianbo Hu, Yibo Zhu, Xuanzhe Liu, Xin Jin, and Hao Zhang. DistServe: Disaggregating Prefill and Decoding for Goodput-optimized Large Language Model Serving. In *18th USENIX Symposium on Operating Systems Design and Implementation*, 2024.

APPENDICES

A EXPERIMENTAL DETAILS

A.1 HYPERPARAMETERS AND REPRODUCIBILITY

Compact mapper. We calibrate on 500 FineWeb-Edu sequences of 1,024 tokens, sampling every fourth position ($\approx$ 128K observations). K and V use separate features from the twelve source layers most frequently selected by the dense Ridge mapper. For Qwen3-8B, each feature vector has width 12,288. We standardize the source features per block and fit a shared rank-1024 projection A with separate target-layer expansions $B^{(\ell)}$. The 40 alternating least-squares updates minimize squared error normalized by the calibration-set variance of each target layer–KV-head block, with ridge penalty $\lambda = 0.01$. Keys are represented before RoPE; target RoPE is applied at the original prompt positions after reconstruction.

Mixed-cache refinement. AdamW minimizes the forward KL objective for one epoch, with both models frozen. The Qwen3 learning rate is 3×10^{-7}; GLM-4 uses 10^{-6}. Training includes a 10% selected-position subset, matching the inference budget. This subset is chosen by a source-attention window surrogate and need not coincide with the inference look-ahead selection.

Selector. The inference selector scores 32-token chunks with source look-ahead attention (8 steps; 4 for GLM-4), width-13 average pooling, a maximum over heads and layers, and a mean over look-ahead steps. It keeps the top $\lceil \rho \cdot \#\text{chunks} \rceil$ chunks, without a trailing local window.

Contextual recomputation. Selected positions are grouped into approximately 32-token chunks and recomputed while attending to the translated prefix. Their new K/V states replace the mapped entries before the next target layer.

Software and hardware. We use vLLM 0.11 with a custom KV connector. Quality is evaluated with lm-evaluation-harness on A100 and H100 GPUs; latency is measured on H100.

Benchmark and scoring contracts. Generation is greedy throughout. LongBench-16 reports a sixteen-task macro average over 3,750 documents. GSM8K uses eight chain-of-thought demonstrations, 1,319 problems, a 512-token generation cap, and the fixed-v2 answer extractor. CSR comprises eight 5-shot log-likelihood tasks (22,419 examples). The MATH-500 comparison uses the same 500 problems and eight non-test mathematics demonstrations for every method. These demonstrations are embedded in the prompt, so harness `num_fewshot=0` adds none; generation is capped at 4,096 tokens. Table 1 uses the final-answer rescore of the fixed generations, with `math-verify==0.9.0` for mathematical equivalence. A missing final answer counts as incorrect. Masks are keyed by the SHA-1 of the full evaluation context, including demonstrations. Chunk rounding can raise CSR's effective keep fraction above 10%, so fixed-capacity SpecPC is not a matched-budget control. LongBench confidence intervals for method differences resample documents within each task as paired units (2,000 draws), then recompute the sixteen-task macro average. Bold denotes the highest point estimate, not a significant difference. The earlier full-response MATH scorer misclassified 18 of 9,000 item–output pairs; the table uses the corrected final-answer-v1 replay. All six methods per pair have 500 outputs with common item fingerprints. CacheBridge's MATH cells are separate full runs of the paper-equation checkpoint under the same task contract, not simultaneous generations. The GLM GSM8K row is an independent prompt-paired 8-shot/512-token run; the older 4-shot results are not pooled with it. On Qwen3-14B CSR, the source-specific. We budget-match SpecPrefill and SpecPC using the same relative keep target.

B Related Work

Multi-model serving and cross-model KV reuse. Model routing, multi-agent serving, prefix caching, and prefill–decode disaggregation reduce redundant inference work, but generally assume that KV is reused by the model that produced it (Ong et al., 2024; Wu et al., 2023; Kwon et al., 2023; Zhong et al., 2024). ICaRus and PrefillShare enable identical cache reuse by training models around a shared cache producer, while DroidSpeak reuses compatible states across closely related model variants (Woo et al., 2026a;b; Liu et al., 2026). Ridge-based transfer and CacheBridge instead translate KV between independently trained models using learned source–target mappings (Heo et al., 2026; Qu et al., 2026). PANORAMA follows this setting but uses mapped KV not only as approximate decoder state, but also as context for selective target-model recomputation.

Selective prefill and KV recomputation. SpecPrefill and SpecPC reduce target prefill by computing only selected prompt positions, leaving the remaining context absent from the target cache (Liu et al., 2025; Galim et al., 2026). CacheBlend instead selectively recomputes context-inconsistent KV when same-model cached chunks are composed (Yao et al., 2024). Unlike CacheBlend, PANORAMA operates over KV translated from a different model, which was never directly computed by the target model. Our results show that such mapped KV can nevertheless provide useful context for target recomputation.

C Baseline Reproduction Fidelity

Ridge. We fit the per-head closed-form mapper with $\lambda = 0.01$, the top twelve source layers per target layer, and RoPE-stripped keys on the same 500 FineWeb-Edu sequences (Heo et al., 2026). On Qwen3-14B→32B, we obtain 90.67% GSM8K; the original paper reports 90.98% with a different evaluation protocol.

CacheBridge. We independently implement the equations of Qu et al. (2026): each target KV head uses the same-index source head in pre-RoPE coordinates, with the paper's attention-guided calibration weights and normalization. We use the same calibration set as Ridge. For the paper's Qwen3-14B→32B pair, Table 5 reports 98.5% of native CSR accuracy, versus 99.83% reported in CacheBridge under its own protocol. The Qwen3-8B and GLM-4 rows are our extensions, not published CacheBridge results. We checked their item identities and failed generations separately from answer extraction.

SpecPrefill. The Qwen-pair configuration matches the selector in Appendix A at $\rho = 10\%$; GLM-4 uses four look-ahead steps. Selection masks are paired with prompts by evaluation-context hash.

SpecPC. We port the official selector: a 64-token weighted-query window from layer 8, max reduction and width-64 max pooling for Qwen pairs, neighboring-token expansion, one source-model draft prefill, and contiguous position renumbering. The official implementation leaves prompts of length $T \leq C_{\max}$ uncompressed; our ratio-budget comparison instead sets $C = \lceil \rho T \rceil$. Draft selection is excluded from the prepared-mask latency clock in Appendix E.

D Latent KV Transfer Fitting

LKT directly fits the factorized mapper introduced in Eq. 4, rather than first fitting a dense mapper and subsequently factorizing the learned mapping. For keys and values separately, the source-side projection is shared across target layers, while each target layer has an independent reconstruction matrix.

For $a \in \{K, V\}$, let x_i^a denote the source feature at position i and $y_i^{(\ell,a)}$ the corresponding native target representation at layer ℓ. We optimize

$$\min_{A_a, \{B_a^{(\ell)}, b_a^{(\ell)}\}} \sum_{\ell} \sigma_{\ell,a}^{-2} \sum_{i} \left\| B_a^{(\ell)} A_a x_i^a + b_a^{(\ell)} - y_i^{(\ell,a)} \right\|_2^2, \tag{5}$$

where A_a is shared across target layers and $B_a^{(\ell)}$ is layer-specific. We solve the factorized objective using alternating least squares (ALS) (Hastie et al., 2015), alternating between least-squares updates of the shared projection and the layer-specific reconstruction matrices. We apply ridge regularization with $\lambda = 0.01$ to the least-squares updates.

For the default configuration, we use rank $r = 1024$ and ridge regularization $\lambda = 0.01$. The mapper is fitted on 500 FineWeb-Edu sequences of 1,024 tokens. Keys are mapped in pre-RoPE content coordinates, and the target positional rotation is applied after reconstruction at the original prompt positions.

E Handoff Latency Protocol

The handoff begins when the source model's prompt cache is ready and ends at the target's first non-empty token (Figure 4). The H100 measurements use separate producer and consumer GPUs, TP1, batch one, greedy decoding, randomized method order, and ten repetitions per method; we report medians after removing client submission delay. The clock includes cache transport but excludes source prefill, online selection/look-ahead, and full generation. Masks are prepared in advance, so the selective-method latencies omit selector cost; SpecPC also omits its draft pass. These are prepared-mask, not end-to-end, times.

Ridge and CacheBridge receive the 144 KiB/token source cache and map on the target. Their 32k mapping is chunked to fit memory. The no-recomputation PANORAMA measurement retains latent transfer and cache expansion but disables selected-token target recomputation. Figure 5(b) reports medians from separate runs at each budget. Payloads are nominal BF16 tensor shapes, not measured network byte counts. Dense Ridge and the compact factor mapper are different learned functions, so the no-recomputation handoff contrast is a pipeline comparison, not a pure mapper-placement ablation.

F CACHE-STATE DIAGNOSTICS

F.1 SELECTED-POSITION STATE ERROR

The GSM8K gain from contextual over isolated recomputation in Table 2(a) is accompanied by a change in the selected states. Table 3 compares both modes and the final mixed-cache-refined rank-1024 mapper with native target states at S ($\rho = 10\%$). At 10k tokens, contextual recomputation lowers relative K/V error from 0.249/0.378 under isolated recomputation to 0.137/0.237; the ordering holds at all five lengths. For each layer, the error is the Frobenius norm over selected positions and heads divided by the native-state norm, averaged across target layers. Keys are measured after key normalization and before RoPE. This diagnostic recomputes all selected positions together at each layer rather than using the chunked serving implementation. With one fixed FineWeb prefix per length, these errors do not establish that state fidelity caused the task-level gain.

Table 3: **Relative K/V error at selected positions versus native target states** on Qwen3-8B→32B (final refined mapper, $\rho = 10\%$; one FineWeb prefix per length).

		K error			V error		
Length	$\lvert S\rvert$	Mapped	Isolated	**Contextual**	Mapped	Isolated	**Contextual**
0.8k	96	0.491	0.259	**0.106**	0.688	0.384	**0.180**
4k	392	0.499	0.300	**0.146**	0.693	0.442	**0.240**
10k	1016	0.490	0.249	**0.137**	0.679	0.378	**0.237**
16k	1576	0.495	0.305	**0.184**	0.691	0.468	**0.306**
32k	3176	0.497	0.270	**0.160**	0.699	0.410	**0.283**

G EVIDENCE-RETRIEVAL DYNAMICS UNDER CROSS-MODEL KV HANDOFF

We use a controlled immediate-answer retrieval stress test to examine how cross-model KV handoff affects the target model's ability to retrieve evidence from the prompt during decoding. We construct seven synthetic retrieval prompts of approximately 16k tokens, wrap them with the Qwen chat template, disable thinking, and explicitly instruct the model to return the answer value immediately. For each prompt, Native Prefill first generates a continuation. Native Prefill, Ridge, and PANORAMA are then teacher-forced with the same first eight Native-generated token IDs. Consequently, all three methods receive identical answer-bearing decode queries, controlling for differences caused by divergent autoregressive trajectories.

For decode step t and target layer ℓ, we define the gold-span attention mass as

$$A_{\ell,t} = \frac{1}{H} \sum_{h=1}^{H} \sum_{j \in G} \alpha_{\ell,h,t,j}, \tag{6}$$

where G denotes the token positions corresponding to the annotated gold answer span, $H = 64$ is the number of query heads, and $\alpha_{\ell,h,t,j}$ denotes the attention probability assigned by query head h at decode step t and layer ℓ to prompt position j. We first combine the two TP2 head shards and then average over all query heads. Each heatmap cell therefore measures the fraction of decode-time attention assigned to the gold span, averaged across all 64 query heads.

The heatmaps visualize the first eight teacher-forced decode queries ($t = 0, \ldots, 7$). Importantly, step 0 does not correspond to the prefill state before generation. Instead, it is the first decode query obtained after feeding the first Native-generated token back into the model.

All three panels share a single linear color scale whose upper bound is set to the pooled 99.5th percentile across all methods. This limits the influence of isolated outliers while preserving direct cross-method comparability.

Under this controlled continuation, Ridge assigns substantially less decode-time attention to the annotated gold span than Native Prefill, whereas PANORAMA recovers a Native-like layer-wise retrieval trajectory. Because all methods receive identical teacher-forced decode queries, these differences cannot be explained by divergent autoregressive trajectories and instead reflect differences induced by the prompt KV cache. Together with the downstream retrieval results, this provides mechanistic evidence that PANORAMA preserves the decoder-compatible KV structure needed to reproduce Native-like evidence-retrieval dynamics after cross-model handoff. Gold-span attention therefore serves as a diagnostic of functional KV compatibility rather than a standalone causal explanation of answer correctness.

H BUDGET SENSITIVITY

PANORAMA's quality gap over SpecPrefill is largest at small recomputation budgets in Table 4. At 2.5%, PANORAMA scores 45.00 on LB-16 against SpecPrefill's 36.72; at 20%, the gap narrows to 45.82 versus 44.61. The advantage is not uniform: at 2.5%, PANORAMA leads in Code (39.07 versus 22.99) but trails slightly on multi-document QA (47.35 versus 48.90). It uses the same trained mapper throughout this sweep, changing only the inference selection budget. Its highest observed LB-16 score is at 10% (46.82); the 1.25% row has no matching latency measurement.

Table 4: **LongBench category quality across recomputation budgets** on Qwen3-8B→Qwen3-32B.

Method	ρ	Single Doc.	Multi Doc.	Summ.	Few Shot	Synth.	Code	LB-16
Native Prefill	–	42.18	51.26	27.36	65.30	58.75	46.05	47.99
PANORAMA	1.25%	38.40	46.38	25.64	62.55	39.50	32.42	41.42
	2.5%	40.74	47.35	26.15	66.39	50.00	39.07	45.00
	5%	40.90	49.44	26.56	68.95	51.50	42.91	46.65
	10%	42.13	51.45	26.71	66.04	51.75	43.33	46.82
	20%	36.45	52.63	26.87	65.92	52.25	41.48	45.82
SpecPrefill	1.25%	27.21	42.14	16.80	44.69	12.75	18.95	28.49
	2.5%	34.99	48.90	19.14	53.79	35.50	22.99	36.72
	5%	38.52	50.30	21.60	60.84	46.00	24.33	40.90
	10%	40.98	49.49	23.76	62.49	52.00	26.95	43.00
	20%	41.12	52.83	25.33	63.29	52.50	30.51	44.61

I QWEN3-14B→32B RESULTS

Table 5 reports Qwen3-14B→32B under the protocol of Table 1. With the larger source model, Ridge alone reaches 39.06 on LB-16 and 90.67% on GSM8K. PANORAMA scores 46.63 and 91.43%, but trails Ridge on MATH-500 (83.60% versus 84.20%) and CSR (74.57 versus 75.62) and SpecPrefill on multi-document QA (51.62 versus 54.02).

On Qwen3-14B CSR, the source-specific SpecPrefill masks keep 8.22–18.36% of the prompt across tasks, versus 10.05–10.47% for relative-budget SpecPC, so the fixed $C_{\max} = 1024$ setting of SpecPC is not a matched-budget control for this table.

Table 5: **Quality after cross-model KV handoff on Qwen3-14B→32B.** Avg. is the macro average over the sixteen LongBench tasks (LB-16); GSM8K, MATH-500, and CSR are accuracies (%). Native Prefill re-prefills the full prompt with the target model, whereas SpecPrefill, SpecPC, and PANORAMA let the target model compute only 10% of the prompt tokens. Bold marks the best handoff method.

		LongBench							Math		
Pair	**Method**	Single Doc.	Multi Doc.	**Summ.**	Few Shot	**Synth.**	**Code**	**Avg.**	**GSM8K**	**MATH-500**	**CSR**
Qwen3 14B→32B	Native Prefill	42.18	51.26	27.36	65.30	58.75	46.05	47.99	91.81	84.80†	75.70
	Ridge	33.34	40.99	23.31	63.65	44.02	26.52	39.06	90.67	**84.20**	**75.62**
	CacheBridge	33.19	44.53	22.05	58.94	**52.79**	34.01	40.61	88.70	81.60	74.53
	SpecPrefill	39.76	**54.02**	23.80	63.04	52.75	28.44	44.02	50.34	39.60	63.92
	SpecPC	35.87	51.83	23.72	59.69	28.75	24.07	38.68	60.42	51.00	49.51
	PANORAMA	**40.52**	51.62	**26.79**	**65.36**	52.75	**43.89**	**46.63**	**91.43**	83.60	74.57

† The Qwen3-32B Native MATH-500 value comes from an independent greedy run of the same target under the stated scoring protocol; Table 1 reports 85.20 from a separate run.