CACHELINK: FUNCTIONAL KV TRANSPORT FOR ZERO-PREFILL CROSS-MODEL CONTEXT HANDOFF

Anonymous authors
Paper under double-blind review

ABSTRACT

KV caches make long-context serving efficient by reusing past computation, but model switching breaks this reuse and forces the receiver to prefill the context again. We study *zero-prefill cross-model context handoff*: translating a sender cache into a receiver-compatible state so that the receiver can continue directly without prefilling same context. Existing affine transport methods show that receiver KV states can be substantially reconstructed from sender caches. However, through extensive analyses, we find that reconstruction fidelity is a poor proxy for receiver utility. Similar K/V reconstruction errors can lead to substantially different attention patterns, predictive distributions, and downstream accuracy, because conventional reconstruction objectives overlook the query-dependent functional importance of K and V in receiver attention. We thus introduce CACHELINK, which learns a coupled K/V correction through the receiver behavior directly. Across four independently trained sender–receiver pairs and seven evaluation panels, CACHELINK improves the equal-weight mean over all methods for every pair. In matched diagnostics, CACHELINK changes K/V squared error by less than 0.4% while substantially changing receiver attention and next-token behavior. At 32K context, it reduces handoff latency by up to $7.27\times$ relative to receiver re-prefill. Our findings verify that reconstruction fidelity alone is insufficient and that preserving the receiver computation induced by the transported cache is important for effective cross-model KV transport. The anonymized code is available at the accompanying repository.

1 INTRODUCTION

KV caches reduce redundant context computation and improve LLM serving efficiency, especially for long-running agents. Yet, when switching models, the cached states cannot be directly transferred to and reused by the new model, forcing it to prefill the same context again. Besides, Routers and cascades increasingly move requests across models to trade inference cost for capability (Dohan et al., 2022; Chen et al., 2024; Aggarwal et al., 2024; Ong et al., 2025; Hu et al., 2024; Zhang et al., 2026). A model switch therefore creates a recurring systems cost: the sender may have already processed tens of thousands of useful tokens, but the receiver cannot directly reuse its KV cache and must prefill the same context again. Same-model serving systems avoid this repeated work through prefix-KV reuse (Kwon et al., 2023; Gim et al., 2024; Srivatsa et al., 2025; Zheng et al., 2024; Qin et al., 2025). We study *zero-prefill cross-model context handoff*, where the sender processes the reusable prefix once, its KV cache is translated into the receiver's cache space, and the receiver continues without replaying the transferred prefix (Figure 1).

Recent cross-model state-transfer work shows that related models admit useful internal alignments (Chen et al., 2025; Zhao et al., 2025; Fu et al., 2026; Dery et al., 2026). For KV transport, per-head Ridge regression reconstructs a substantial fraction of the receiver cache from sender states (Heo et al., 2026), and CacheBridge further improves affine mapping across models (Qu et al., 2026). These results establish reconstruction as a strong starting point. They do not establish reconstruction fidelity as the right objective. Receiver queries consume the translated K and V jointly through attention, and we find that caches with similar K/V error can induce substantially different attention, predictive distributions, and task accuracy.

This reconstruction–utility mismatch motivates CACHELINK. CACHELINK learns a coupled K/V correction, and trains the transported state through the frozen receiver with functional supervision.

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.

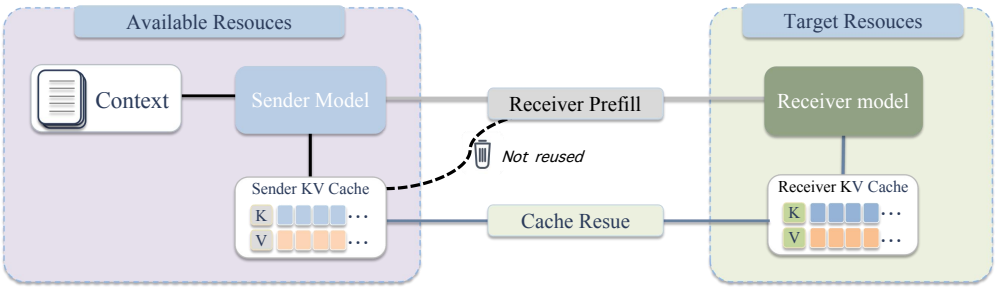

Figure 1: **A model switch breaks native KV-cache reuse.** The sender has already processed the accumulated context and produced a reusable cache. Native switching discards that computation and rebuilds the receiver cache by prefilling the same prefix again. Cross-model handoff translates the sender cache into a receiver-compatible state, allowing the receiver to continue without replaying the transferred context.

Ridge supplies the cross-model structure already captured by affine prediction (Heo et al., 2026); the functional objective directs subsequent correction toward the behavior that the receiver actually uses. Deployment remains zero-prefill: the receiver processes only the new query and generated tokens.

Across four independently trained sender→receiver pairs and seven fixed evaluation panels, CACHELINK achieves a higher equal-weight mean than both Ridge and the native sender for every pair and outperforms Ridge in 26 of 28 pair–panel comparisons. Matched diagnostics sharpen the mechanism: receiver attention and next-token behavior change substantially while K/V squared error changes by less than 0.4%. At a 32K reusable prefix, CACHELINK reduces handoff latency by up to 7.27× relative to receiver re-prefill.

Our main contributions are:

- **Problem and finding.** We formulate zero-prefill cross-model context handoff and conduct extensive diagnostic analyses of cross-model KV reuse. These analyses show that KV reconstruction fidelity does not determine downstream receiver utility: caches with similar reconstruction error can induce markedly different receiver behavior.
- **Method.** We introduce CACHELINK, which learns a coupled K/V correction through functional supervision from a frozen receiver. By optimizing the computation induced by the transported cache, CACHELINK targets receiver behavior directly instead of focusing solely on accurate tensor reconstruction.
- **Evaluation.** Across four sender→receiver model pairs and seven datasets, CACHELINK consistently outperforms Ridge while preserving the latency advantage of zero-prefill handoff, achieving up to 7.27× lower handoff latency than receiver re-prefill at 32K context.

2 RELATED WORK

Cross-model representation and state transfer. Representation-similarity analyses show that independently trained models often share substantial geometric structure (Kornblith et al., 2019; Huh et al., 2024), while model-stitching methods exploit this structure by learning mappings between intermediate states (Bansal et al., 2021; Chen et al., 2025). Recent LLM systems extend this idea to runtime transfer through attention mapping, KV projection, shared latent spaces, selective recomputation, or coordinated prefill (Zhao et al., 2025; Fu et al., 2026; Dery et al., 2026; Liu et al., 2026b; Woo et al., 2026a;b).

Cross-model KV transport. KV caches are especially attractive transfer targets because they contain the state needed to continue autoregressive decoding after a prefix has been processed. Recent work shows that sender KV states can predict a substantial fraction of receiver KV structure using simple affine mappings (Heo et al., 2026), and CacheBridge develops related cross-model

cache mappings for efficient reuse (Qu et al., 2026). Our experiments build on this reconstruction basis: because the receiver consumes K and V jointly through attention, transported-cache utility depends on the computation it induces, not only on elementwise agreement with native tensors. CacheLink therefore starts from an affine receiver-cache estimate and learns a coupled correction through receiver-level functional supervision.

KV reuse, serving, and model routing. Within a single model, modern serving systems preserve prefill work through prefix caching, shared-prefix scheduling, disaggregated serving, and selective cache reuse across repeated or overlapping requests (Gim et al., 2024; Zheng et al., 2024; Srivatsa et al., 2025; Qin et al., 2025; Yao et al., 2025). Complementary work reduces the memory and bandwidth cost of an already-compatible cache through quantization, token selection, and cache compression (Liu et al., 2024; Tang et al., 2024; Li et al., 2024). At the same time, routers and cascades increasingly move requests among models with different capability and cost profiles (Dohan et al., 2022; Chen et al., 2024; Aggarwal et al., 2024; Ong et al., 2025; Zhang et al., 2026). These two trends meet in long-running workloads: context may be reusable, yet the next model selected by the router cannot directly consume the previous model's KV state. Cross-model cache transport turns that already-computed prefix state into something that can survive a model switch, allowing routing and context reuse to coexist without replaying the reusable prefix.

3 Problem Setup

3.1 Zero-prefill cross-model handoff

Let a sender model S process a reusable prefix $x_{1:n}$ and produce

$$C^S(x_{1:n}) = \left\{K_l^S(x_{1:n}), V_l^S(x_{1:n})\right\}_{l=1}^{L_S}, \qquad C^T(x_{1:n}) = \left\{K_l^T(x_{1:n}), V_l^T(x_{1:n})\right\}_{l=1}^{L_T}. \quad (1)$$

A receiver model T would ordinarily prefill the same prefix to construct $C^T(x_{1:n})$. We instead learn :

$$F:\ C^S(x_{1:n}) \to \widehat{C}^T(x_{1:n}), \quad (2)$$

as a transport and inject $\widehat{C}^T$ directly into the receiver. Under the strict zero-prefill constraint used throughout this paper, deployment performs no target-prefix prefill, sparse replay, or receiver-side reconstruction pass over $x_{1:n}$. The complete receiver cache is produced from the sender cache alone.

The transported cache is the state consumed by future receiver queries. At each attention layer, K determines how a query routes probability mass across cached positions, while V determines the information returned along those routes. Handoff quality therefore depends on the receiver computation induced by the paired K/V state.

3.2 Reconstruction baseline

The Ridge baseline treats handoff as supervised reconstruction of the native receiver cache (Heo et al., 2026). For each target layer l, let $\mathcal{S}_l$ contain the k sender layers selected by held-out predictability. After removing source RoPE from keys, selected states are concatenated at each token position,

$$\boldsymbol{x}_{l,t}^K = \mathrm{concat}_{j\in\mathcal{S}_l} K_{j,t}^S, \qquad \boldsymbol{x}_{l,t}^V = \mathrm{concat}_{j\in\mathcal{S}_l} V_{j,t}^S. \quad (3)$$

Independent affine regressors predict each target-layer KV head,

$$\widehat{K}_{l,h,t}^T = \boldsymbol{x}_{l,t}^K W_{l,h}^K + b_{l,h}^K, \quad \widehat{V}_{l,h,t}^T = \boldsymbol{x}_{l,t}^V W_{l,h}^V + b_{l,h}^V. \quad (4)$$

$$W^\star = \arg\min_W \left(\|XW - Y\|_F^2 + \lambda\|W\|_F^2\right). \quad (5)$$

Target RoPE is reapplied to $\widehat{K}^T$ before cache injection. Our matched baseline uses $k = 8$ and the published 500×1024 FineWeb-Edu calibration protocol. Ridge reconstructs a substantial fraction of the receiver KV state and provides a strong structural baseline for cross-model handoff.

Does lower KV reconstruction error produce a more useful receiver cache?

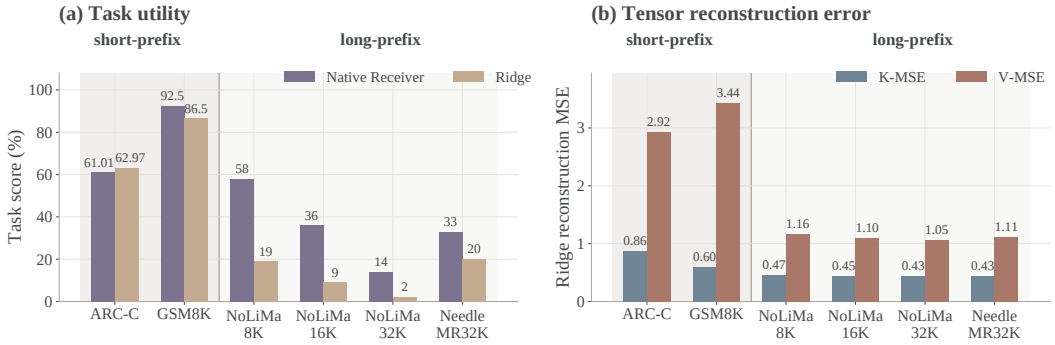

Figure 2: **Long-prefix utility degrades even as KV reconstruction error decreases.** Both panels use the same dataset order and the same Qwen3-14B→32B diagnostic. (a) Native and Ridge task scores. (b) Ridge K/V reconstruction MSE. Across NoLiMa 8K–32K, reconstruction error decreases while task utility falls sharply.

4 Reconstruction fidelity does not determine receiver utility

4.1 Reconstruction fidelity does not predict handoff utility

Ridge reconstructs substantial receiver KV structure from the sender cache, yet its downstream utility varies sharply across tasks. To test whether reconstruction error explains this variation, we use a targeted Qwen3-14B→32B diagnostic; the full seven-panel comparison across all four sender–receiver pairs appears in Appendix Figure 8.

Figure 2 compares ARC-C and GSM8K with NoLiMa 8K/16K/32K and NeedleBench MR32K. Ridge remains relatively close to the native receiver on the short-prefix tasks, while its NoLiMa score drops from 19% to 9% to 2% as the reusable prefix grows from 8K to 16K to 32K; the native receiver scores 58%, 36%, and 14%. Over the same sweep, K-MSE decreases from 0.465 to 0.433 and V-MSE decreases from 1.161 to 1.054. Utility therefore degrades while reconstruction error improves.

Observation 1. Lower K/V reconstruction error does not imply higher receiver utility.

4.2 Receiver behavior can change without better reconstruction

The receiver consumes a translated cache through attention. For a receiver query Q,

$$\mathrm{Attn}(Q, \widehat{K}, \widehat{V}) = \mathrm{softmax}\left(\frac{Q\widehat{K}^\top}{\sqrt{d_k}}\right)\widehat{V}. \tag{6}$$

A perturbation to K changes where the query places attention; a perturbation to V changes what is retrieved under those weights. These effects are query-dependent and coupled. Mean squared cache error instead averages over layers, heads, tokens, and dimensions, without measuring the receiver computation induced by the translated state. We measure that receiver-level computation on a matched Qwen3-14B→32B diagnostic drawn from the same seven evaluation panels. For 50 examples per panel, we compare the native receiver cache, the Ridge base paired with the step-600 CacheLink checkpoint, and the final CacheLink cache. Appendix A gives the sampling and metric definitions.

A small residual changes receiver behavior while leaving reconstruction nearly unchanged. Figure 3 connects the geometry of the learned residual to the computation it induces in the receiver. Panels (a) and (b) show that the correction is small—only 1.52–1.79% of Ridge RMS for K and 1.88–2.40% for V—and is nearly orthogonal to the Native–Ridge gap across panels. Panel (c) then shows that adding this residual barely changes the average K/V reconstruction error. In contrast, panels (d) and (e) show clear changes in the receiver's prefix-attention and next-token distributions. The same small tensor-space correction can therefore have a much larger effect after it passes through receiver attention and prediction. On the separate 200-example short-prefix multiple-choice subset,

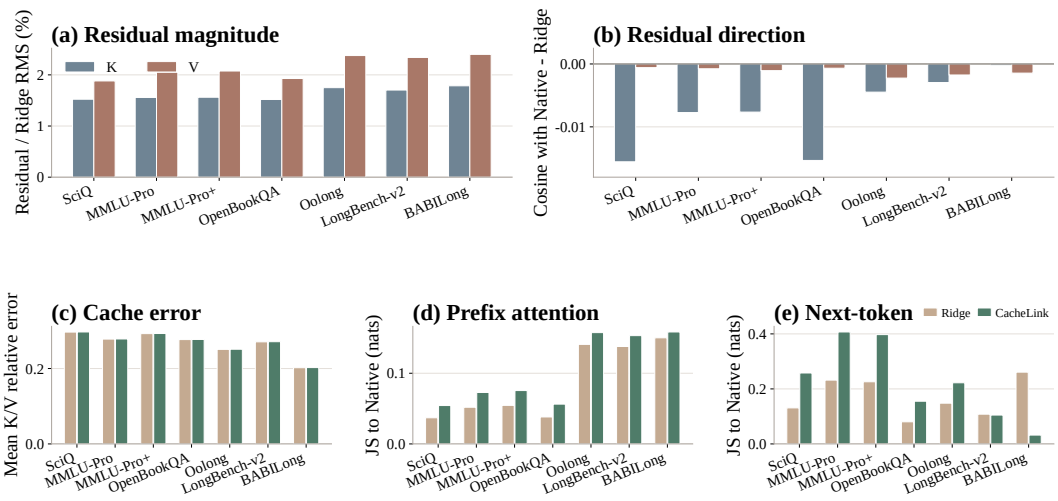

Figure 3: **A small residual changes receiver behavior while leaving reconstruction nearly unchanged.** The top row shows residual size and direction; the bottom row compares cache error, prefix attention, and next-token behavior.

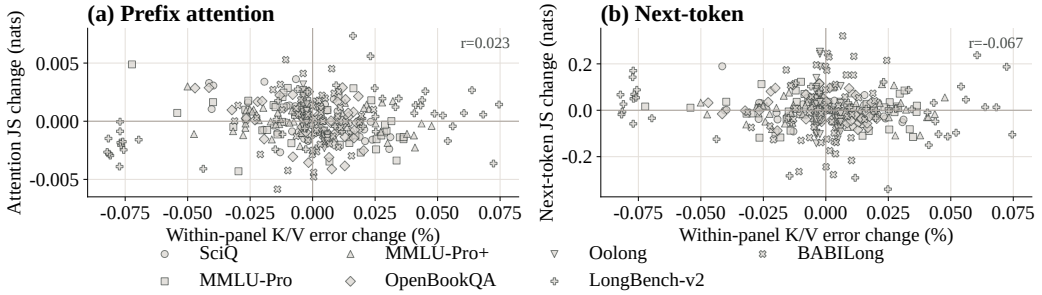

Figure 4: **Cache-error changes do not track receiver-level changes within panels.** Points are matched examples after within-panel centering.

length-normalized accuracy rises from 48% to 51%, with 12 wrong-to-correct and 6 correct-to-wrong flips.

The decoupling persists example by example. Figure 4 removes panel offsets by centering each metric within its panel before pooling examples. The centered Pearson association between mean K/V error reduction and attention-JS change is -0.023; the association with next-token-JS change is 0.067. On the short-prefix panels, the association with normalized gold-answer log-probability change is 0.181. Movement toward the native cache in squared-error space therefore carries little information about how receiver behavior changes.

Observation 2. Receiver-level functional change is largely decoupled from movement toward the native cache in squared-error space.

Observations 1 and 2 assign reconstruction a clear role: it provides useful structure, but receiver behavior must supply the optimization signal.

4.3 K AND V SHARE A COMMON LATENT ORIGIN

At each layer and token, K and V are different projections of the same underlying hidden representation. Suppressing normalization and positional transforms for clarity, they take the form $K = hW_K$ and $V = hW_V$. Cross-model shifts in K and V can therefore share factors inherited from h. Their

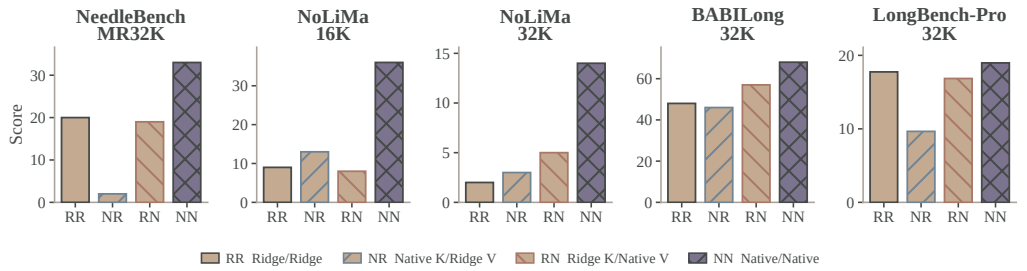

Figure 5: **K and V interact functionally under cross-model cache handoff.** RR, NR, RN, and NN denote Ridge K/Ridge V, Native K/Ridge V, Ridge K/Native V, and Native K/Native V, respectively. Replacing only one component does not produce a uniform improvement across tasks, so the effect of K depends on the paired V state and vice versa.

functional roles are coupled as well: K determines the attention weights applied to V, so the effect of correcting one component depends on the state of the other.

We test this interaction with a targeted source-swap intervention on five tasks. Figure 5 compares Ridge K/Ridge V (RR), Native K/Ridge V (NR), Ridge K/Native V (RN), and Native K/Native V (NN). RR is the deployable Ridge state; NR, RN, and NN are oracle interventions that expose the effect of replacing one or both cache components. Replacing only K or only V does not yield a consistent improvement across tasks. Correcting one component can help or fail depending on the paired state of the other, matching their shared origin and coupled use in attention.

Observation 3. K and V transport errors share latent structure and should be corrected jointly.

Each observation maps directly to a CacheLink design choice. Ridge supplies the structural initialization; receiver-level supervision optimizes functional utility; and separate K/V stems followed by late fusion capture projection-specific and shared structure in a joint correction. Section 5 implements these choices.

5 CacheLink: functional residual KV transport

The three observations in Section 4 define the design of CacheLink. Ridge initializes a complete receiver cache; separate K/V stems preserve projection-specific structure before late fusion forms a joint correction; and functional supervision trains the transported state through the frozen receiver with continuation CE and forward KL.

Ridge as structural initialization. A substantial fraction of receiver KV structure is linearly predictable from the sender cache. In the layerwise diagnostic summarized in Figure 6, mean held-out R^2 across the 64 receiver layers is 0.838 for K and 0.609 for V. Ridge therefore provides a complete, structured starting state; CacheLink uses its additional capacity to optimize the functional correction that reconstruction alone leaves unresolved.

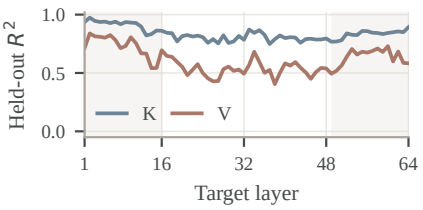

Figure 6: **Linear KV predictability.**

For target layer l, let $\mathcal{S}_l$ contain the same $k = 8$ sender layers selected by the Ridge baseline. At token position t, we form

$$X^K_{l,t} = \text{concat}_{j\in\mathcal{S}_l} K^S_{j,t}, \quad X^V_{l,t} = \text{concat}_{j\in\mathcal{S}_l} V^S_{j,t}. \tag{7}$$

where source keys are represented after kNorm and de-RoPE and values remain in raw cache space. The fitted Ridge maps produce the receiver-cache initialization,

$$B^K_{l,t} = R^K_l(X^K_{l,t}), \qquad B^V_{l,t} = R^V_l(X^V_{l,t}). \tag{8}$$

CACHELINK predicts residual corrections ΔK and ΔV on top of this initialization. The residual output heads are zero-initialized, so every checkpoint begins from the Ridge solution. The fitted Ridge maps remain fixed throughout CacheLink training; only the residual mapper parameters are optimized.

Coupled K/V residual transport. K and V are produced from the same layer input hidden representation through different projections. We preserve their projection-specific structure with separate stems,

$$z_{l,t}^K = \mathrm{GELU}(W_l^K X_{l,t}^K + b_l^K), \qquad z_{l,t}^V = \mathrm{GELU}(W_l^V X_{l,t}^V + b_l^V). \tag{9}$$

The two views are then fused before either correction is produced:

$$u_{l,t} = \mathrm{GELU}\big(W_l^F [z_{l,t}^K; z_{l,t}^V] + b_l^F\big), \tag{10}$$

$$\Delta K_{l,t} = H_l^K u_{l,t}, \qquad \Delta V_{l,t} = H_l^V u_{l,t}. \tag{11}$$

The shared representation $u_{l,t}$ captures latent factors jointly expressed through K and V. The resulting receiver cache is

$$\widehat{K}_{l,t}^T = \mathrm{RoPE}_T(B_{l,t}^K + \Delta K_{l,t}), \qquad \widehat{V}_{l,t}^T = B_{l,t}^V + \Delta V_{l,t}. \tag{12}$$

Functional supervision through the receiver. We inject the translated cache into a frozen receiver and optimize the behavior induced by that state. For a continuation $y_{1:T}$ following the reusable prefix, the training objective is

$$\mathcal{L}_{\mathrm{func}} = -\frac{1}{T}\sum_t \log p_{\mathrm{map}}(y_t) + \lambda_{\mathrm{KL}} \frac{1}{T}\sum_t D_{\mathrm{KL}}(p_{\mathrm{native},t} \,\|\, p_{\mathrm{map},t}). \tag{13}$$

The CE term trains the transported state to support the target continuation; forward KL transfers the predictive behavior of a native receiver that prefills the same prefix. Gradients pass through receiver attention into CACHELINK, while receiver parameters remain frozen. The final objective contains no cache-reconstruction loss. Appendix A reports the loss ablation and implementation details.

Zero-prefill deployment. At deployment, the sender computes the reusable prefix once, CACHELINK translates the resulting cache once, and the receiver processes only the new query and generated tokens. The receiver never replays the transferred prefix.

6 EXPERIMENTS

6.1 EXPERIMENTAL SETUP

Models and zero-prefill protocol. We evaluate four sender→receiver pairs: Qwen3-14B→32B and Qwen3-8B→14B (Yang et al., 2025), Ministral-3-8B-Reasoning-2512→Ministral-3-14B-Reasoning-2512 (Liu et al., 2026a), and Yi-1.5-9B→34B (01.AI et al., 2024). Every handoff method constructs a complete receiver cache from the sender cache and performs zero receiver prefill on the reusable prefix. Keys are translated after source de-RoPE and receive target RoPE before injection; values remain in raw cache space. Within each model pair and evaluation panel, all methods use identical tokenized inputs, prompts, decoding rules, and scorers.

Evaluation panels. We use seven fixed panels spanning short- and long-prefix settings. SciQ (Welbl et al., 2017), MMLU-Pro (Wang et al., 2024), MMLU-Pro-Plus (Taghanaki et al., 2024), and OpenBookQA (Mihaylov et al., 2018) are short-prefix multiple-choice tasks. Oolong-Synth 32K (Bertsch et al., 2025), LongBench-v2 (Bai et al., 2025), and BABILong 32K (Kuratov et al., 2024) use long reusable prefixes and task-specific scorers. MMLU-Pro and MMLU-Pro-Plus share question stems and are reported separately. The equal-weight mean provides a compact within-pair summary of the seven panel scores.

Baselines. We compare CACHELINK with the native sender, native receiver, Ridge-$k8$ (Heo et al., 2026), CacheBridge (Qu et al., 2026), and LatentAlign (Dery et al., 2026). Native receiver re-prefills the reusable prefix and serves as the large-model reference; all cache-transport methods operate under the same zero-prefill protocol.

Table 1: Main results across four sender→receiver model pairs. Scores are percentages. CB and LA denote CacheBridge and LatentAlign, respectively. Native receiver is shown as the large-model reference and is excluded from boldface selection. Bold marks the best among native sender, Ridge-$k8$, CB, LA, and CACHELINK. Equal-weight means summarize the seven panel scores and are not pooled accuracy.

Dataset / panel	Native receiver	Native sender	Ridge-$k8$	CB	LA	CACHELINK
Qwen3 14B→32B						
SciQ	73.30	59.70	74.80	60.20	25.50	**83.90**
MMLU-Pro	48.96	36.36	41.20	36.54	11.10	**47.46**
MMLU-Pro-Plus	36.03	25.08	35.75	29.39	6.40	**40.32**
OpenBookQA	39.20	37.80	40.20	39.00	28.00	**42.20**
Oolong-Synth 32K	29.00	14.23	26.55	25.75	0.00	**38.12**
LongBench-v2	48.28	43.10	42.24	35.34	0.00	**44.83**
BABILong 32K	62.20	54.60	43.60	0.00	0.00	**55.40**
Equal-weight mean	48.14	38.70	43.48	32.32	10.14	**50.32**
Ministral 3 8B→14B						
SciQ	54.30	65.60	59.10	50.90	22.40	**72.60**
MMLU-Pro	40.46	28.10	30.49	21.89	11.69	**34.97**
MMLU-Pro-Plus	46.98	22.17	19.71	12.11	6.57	**23.45**
OpenBookQA	35.60	36.40	35.20	33.60	26.60	**38.80**
Oolong-Synth 32K	42.86	38.20	**42.01**	33.81	0.00	36.14
LongBench-v2	45.69	37.93	**42.24**	38.79	0.00	39.66
BABILong 32K	50.80	**51.60**	41.20	37.80	0.00	45.00
Equal-weight mean	45.24	40.00	38.56	32.70	9.61	**41.52**
Qwen3 8B→14B						
SciQ	59.70	48.10	47.50	48.70	21.80	**81.90**
MMLU-Pro	36.34	40.49	20.78	11.81	11.35	**44.28**
MMLU-Pro-Plus	25.08	**33.50**	15.51	6.61	6.32	30.00
OpenBookQA	37.80	36.60	34.40	31.20	26.40	**39.80**
Oolong-Synth 32K	47.78	41.11	41.20	3.72	0.00	**43.02**
LongBench-v2	39.66	**38.79**	31.90	32.76	0.00	37.93
BABILong 32K	53.80	**51.00**	9.00	0.00	0.00	35.40
Equal-weight mean	42.88	41.37	28.61	19.26	9.41	**44.62**
Yi-1.5 9B→34B						
SciQ	80.00	72.70	62.00	51.80	23.90	**81.00**
MMLU-Pro	41.40	31.57	24.35	11.87	11.52	**34.18**
MMLU-Pro-Plus	41.57	**35.25**	18.09	6.70	6.51	24.69
OpenBookQA	44.20	38.40	33.60	31.60	28.80	**41.20**
Oolong-Synth 32K	20.26	1.61	12.84	3.75	0.00	**26.67**
LongBench-v2	31.90	23.28	22.41	7.76	0.00	**30.17**
BABILong 32K	41.40	**37.40**	8.40	3.40	0.00	21.80
Equal-weight mean	42.96	34.32	25.96	16.70	10.10	**37.10**

Training. All four CACHELINK checkpoints use receiver continuation CE and forward KL as functional supervision. The Ridge maps are fitted before CacheLink training and remain frozen for every model pair; CacheLink optimizes only the residual mapper. Qwen3-14B→32B trains its late K/V residual on 400 earlier samples plus 200 LongAlign reasoning samples (Bai et al., 2024). Ministral-3-8B→14B trains the six-source residual over its fixed Ridge cache. Qwen3-8B→14B and Yi-1.5-9B→34B train zero-initialized late K/V residuals for 600 optimizer steps on 1,200 examples, with two examples per step. Appendix A gives the complete training sources, cache conventions, and checkpoint dependencies. The experiments test three claims: functional supervision recovers utility missed by reconstruction, the gain transfers across model pairs and prefix regimes, and zero-prefill transport retains its latency advantage.

6.2 FUNCTIONAL SUPERVISION IMPROVES RECEIVER UTILITY

CACHELINK improves the equal-weight seven-panel mean over Ridge for every sender→receiver pair: +6.84 points for Qwen3-14B→32B, +2.96 for Ministral-3-8B→14B, +16.01 for Qwen3-8B→14B, and +11.14 for Yi-1.5-9B→34B. At the panel level, it exceeds Ridge in 26 of 28 comparisons. The same aggregate pattern holds against the native sender, CacheBridge, and LatentAlign: CACHELINK has the higher equal-weight mean for all four model pairs and exceeds the native sender in 21 of 28 pair–panel comparisons. These results match the mechanism in Section 4. Ridge supplies a strong reconstruction-based state, while receiver-level functional supervision recovers utility that reconstruction error does not identify.

6.3 THE GAIN TRANSFERS ACROSS MODEL PAIRS AND PREFIX REGIMES

The improvement is not confined to one model family or context regime. On Qwen3-14B→32B, Qwen3-8B→14B, and Yi-1.5-9B→34B, CACHELINK improves over Ridge on both the short-prefix and long-prefix panels. Ministral-3-8B→14B shows the concentrated exception: CACHELINK improves all four short-prefix panels and BABILong, while Ridge remains stronger on Oolong-Synth and LongBench-v2.

The cross-pair pattern is therefore broad but task-dependent: functional transport improves both short- and long-context handoff across three model families, while the remaining regressions are concentrated in two long-prefix panels for one model pair.

6.4 ZERO-PREFILL TRANSPORT PRESERVES THE LATENCY ADVANTAGE

We measure end-to-end handoff latency in the intended deployment setting: the sender has already processed the reusable prefix and its KV cache is available when a model switch occurs. Native receiver latency includes re-prefilling that prefix, whereas Ridge and CACHELINK translate the available sender cache before continuation. Figure 7 reports latency from 2K to 32K reusable prefixes on Qwen3-14B→32B and Ministral-3-8B→14B.

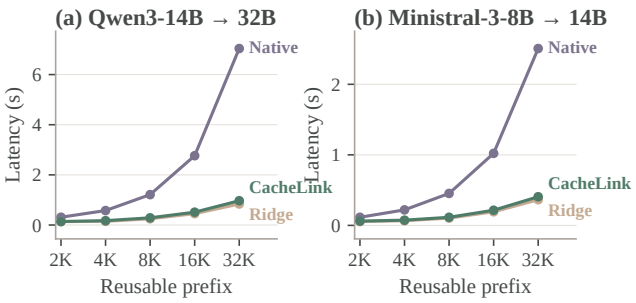

Figure 7: **Zero-prefill transport scales better with prefix length.** At 32K, CacheLink is 7.27× and 6.18× faster than receiver re-prefill on the two model pairs.

The residual correction adds modest overhead to Ridge while preserving most of the transport advantage. At 32K, latency increases from 0.839s for Ridge to 0.969s for CACHELINK on Qwen3-14B→32B, and from 0.364s to 0.406s on Ministral-3-8B→14B. Both remain far below receiver re-prefill, reaching 7.27× and 6.18× speedups respectively. The advantage grows with prefix length because re-prefill scales with the transferred context while cache translation avoids receiver-prefix replay.

7 CONCLUSION

Cross-model KV states can be substantially reconstructed, but reconstruction fidelity does not determine whether the receiver can use the transported state effectively. CACHELINK turns this observation into a transport objective: it starts from a Ridge-initialized receiver cache, couples K/V corrections through late fusion, and trains the transported state through the frozen receiver. Across four sender→receiver pairs and seven fixed evaluation panels, CACHELINK improves the equal-weight mean over both Ridge and the native sender for every pair, exceeding Ridge in 26 of 28 pair–panel comparisons and the sender in 21 of 28. Matched diagnostics show why: CacheLink changes K/V squared error by less than 0.4% while substantially changing receiver attention and next-token behavior. Zero-prefill cross-model handoff is therefore better framed as preserving receiver computation than as reconstructing receiver tensors.

Reproducibility statement

Appendix A specifies each model pair's cache convention, mapper construction, prompts, decoding, and scoring. The anonymized source artifact contains the selected training and evaluation code but does not bundle model weights, Ridge fits, benchmark panels, predictions, or run logs. The manuscript artifact provides the fixed-panel result tables and, for the matched diagnostic, sanitized row identifiers, panel hashes, per-example numeric metrics, and aggregation code; all required base models, checkpoints, and evaluation inputs must be prepared separately as described in the appendix.

AI use statement

Generative AI tools were used to assist with literature search, experiment orchestration, plotting code, and language editing. All numerical claims are derived from stored experiment artifacts; the authors reviewed the generated text and code, checked citations against primary sources, and take responsibility for the final content.

References

01.AI, Alex Young, Bei Chen, Chao Li, Chengen Huang, Ge Zhang, et al. Yi: Open foundation models by 01.ai. *arXiv preprint arXiv:2403.04652*, 2024.

Pranjal Aggarwal, Aman Madaan, Ankit Anand, Srividya Pranavi Potharaju, Swaroop Mishra, Pei Zhou, Aditya Gupta, Dheeraj Rajagopal, Karthik Kappaganthu, Yiming Yang, Shyam Upadhyay, Manaal Faruqui, and Mausam. Automix: Automatically mixing language models. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Yushi Bai, Xin Lv, Jiajie Zhang, Yuze He, Ji Qi, Lei Hou, Jie Tang, Yuxiao Dong, and Juanzi Li. Longalign: A recipe for long context alignment of large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 1376–1395, 2024.

Yushi Bai, Shangqing Tu, Jiajie Zhang, Hao Peng, Xiaozhi Wang, Xin Lv, Shulin Cao, Jiazheng Xu, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. Longbench v2: Towards deeper understanding and reasoning on realistic long-context multitasks. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics*, pp. 3639–3664, 2025.

Yamini Bansal, Preetum Nakkiran, and Boaz Barak. Revisiting model stitching to compare neural representations. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

Amanda Bertsch, Adithya Pratapa, Teruko Mitamura, Graham Neubig, and Matthew R. Gormley. Oolong: Evaluating long context reasoning and aggregation capabilities. *arXiv preprint arXiv:2511.02817*, 2025.

Alan Chen, Jack Merullo, Alessandro Stolfo, and Ellie Pavlick. Transferring linear features across language models with model stitching. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Lingjiao Chen, Matei Zaharia, and James Zou. Frugalgpt: How to use large language models while reducing cost and improving performance. *Transactions on Machine Learning Research*, 2024.

Lucio M. Dery, Zohar Yahav, Henry Prior, Qixuan Feng, Jiajun Shen, and Arthur Szlam. Latent space communication via k-v cache alignment. *arXiv preprint arXiv:2601.06123*, 2026.

David Dohan, Winnie Xu, Aitor Lewkowycz, Jacob Austin, David Bieber, Raphael Gontijo Lopes, Yuhuai Wu, Henryk Michalewski, Rif A. Saurous, Jascha Sohl-Dickstein, Kevin Murphy, and Charles Sutton. Language model cascades. *arXiv preprint arXiv:2207.10342*, 2022.

Tianyu Fu, Zihan Min, Hanling Zhang, Jichao Yan, Guohao Dai, Wanli Ouyang, and Yu Wang. Cache-to-cache: Direct semantic communication between large language models. In *International Conference on Learning Representations*, 2026.

In Gim, Guojun Chen, Seung-seob Lee, Nikhil Sarda, Anurag Khandelwal, and Lin Zhong. Prompt cache: Modular attention reuse for low-latency inference. In *Proceedings of Machine Learning and Systems*, volume 6, 2024.

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.

Qitian Jason Hu, Jacob Bieker, Xiuyu Li, Nan Jiang, Benjamin Keigwin, Gaurav Ranganath, Kurt Keutzer, and Shriyash Kaustubh Upadhyay. Routerbench: A benchmark for multi-llm routing system. In *ICML 2024 Workshops: Agentic Markets*, 2024.

Minyoung Huh, Brian Cheung, Tongzhou Wang, and Phillip Isola. Position: The platonic representation hypothesis. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 20617–20642, 2024.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *Proceedings of the 36th International Conference on Machine Learning*, pp. 3519–3529, 2019.

Yuri Kuratov, Aydar Bulatov, Petr Anokhin, Ivan Rodkin, Dmitry Sorokin, Artyom Sorokin, and Mikhail Burtsev. Babilong: Testing the limits of llms with long context reasoning-in-a-haystack. *arXiv preprint arXiv:2406.10149*, 2024.

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 29th Symposium on Operating Systems Principles*, pp. 611–626, 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. *arXiv preprint arXiv:2404.14469*, 2024.

Alexander H. Liu, Kartik Khandelwal, Sandeep Subramanian, Victor Jouault, Abhinav Rastogi, et al. Ministral 3. *arXiv preprint arXiv:2601.08584*, 2026a.

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*, pp. 319–338, 2026b.

Zirui Liu, Jiayi Yuan, Hongye Jin, Shaochen 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*, pp. 32332–32344, 2024.

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 the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2381–2391, 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. In *International Conference on Learning Representations*, 2025.

Ruoyu Qin, Zheming Li, Weiran He, Jialei Cui, Feng Ren, Mingxing Zhang, Yongwei Wu, Weimin Zheng, and Xinran Xu. Mooncake: Trading more storage for less computation – a kvcache-centric architecture for serving llm chatbot. In *23rd USENIX Conference on File and Storage Technologies*, pp. 155–170, 2025.

Xingyu Qu, Siyuan Lu, Zhiyu Chen, Sheng Wang, and Tao Lin. Cachebridge: Efficient cross-model kv cache transfer. *arXiv preprint arXiv:2609.00891*, 2026.

Vikranth Srivatsa, Zijian He, Reyna Abhyankar, Dongming Li, and Yiying Zhang. Preble: Efficient distributed prompt scheduling for llm serving. In *International Conference on Learning Representations*, 2025.

Saeid Asgari Taghanaki, Aliasgahr Khani, and Amir Khasahmadi. Mmlu-pro+: Evaluating higher-order reasoning and shortcut learning in llms. *arXiv preprint arXiv:2409.02257*, 2024.

Jiaming Tang, Yilong Zhao, Kan Zhu, Guangxuan Xiao, Baris Kasikci, and Song Han. Quest: Query-aware sparsity for efficient long-context llm inference. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 47901–47911, 2024.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, et al. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *arXiv preprint arXiv:2406.01574*, 2024.

Johannes Welbl, Nelson F. Liu, and Matt Gardner. Crowdsourcing multiple choice science questions. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pp. 94–106, 2017.

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. Prefill-share: A shared prefill module for kv reuse in multi-llm disaggregated serving. *arXiv preprint arXiv:2602.12029*, 2026b.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, 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. In *Proceedings of the Twentieth European Conference on Computer Systems*, 2025.

Guibin Zhang, Haiyang Yu, Kaiming Yang, Bingli Wu, Fei Huang, Yongbin Li, and Shuicheng Yan. Evoroute: Experience-driven self-routing llm agent systems. *arXiv preprint arXiv:2601.02695*, 2026.

Yi Zhao, Zuchao Li, and Hai Zhao. Iam: Efficient inference through attention mapping between different-scale llms. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics*, pp. 19522–19533, 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 *Advances in Neural Information Processing Systems*, volume 37, 2024.

A Evaluation and diagnostic protocol

Cache transport. Qwen3-14B is the sender and Qwen3-32B is the receiver for the mechanism diagnostic. The receiver has 64 transformer layers; both models expose eight KV heads of dimension 128. A handoff constructs a complete receiver cache and does not prefill the receiver on the reusable prefix. Keys are mapped in de-RoPE content space and receive the receiver's RoPE after transport; values are mapped in raw cache space.

CacheLink architecture. For Qwen3-14B→32B, concatenating the eight selected source layers gives 8,192-dimensional K and V views. The two stems are $8192 \to 256$, the late-fusion layer is $512 \to 512$, and two $512 \to 1024$ heads emit all eight target K or V heads. The residual mapper has 5,508,096 trainable parameters per target layer and 352,518,144 across the 64-layer receiver. It operates directly on the sender cache and requires neither a source-side solo forward pass nor receiver replay of the transferred prefix.

Training objective	Correct / 100
CE only	37
CE + forward KL	**40**
CE + reverse KL	39
Forward KL only	29
Reverse KL only	23
Ridge	28
Native sender (14B)	36
Native receiver (32B)	59

Table 2: **Loss-function ablation on Qwen3-14B→32B.**

Ridge fits and step-600 checkpoints. The main-table Ridge baseline is calibrated on 500 FineWeb-Edu sequences of 1,024 tokens, with $k = 8$ selected source layers, all eight source KV heads, and a separate affine regressor for each target layer/head and for K/V. For every model pair, these Ridge maps are fitted before CacheLink training and then frozen. The Qwen3-14B→32B step-600 CacheLink checkpoint trains only the late K/V residual on 400 earlier samples plus 200 LongAlign reasoning samples using receiver continuation cross-entropy and forward KL; inference loads the fixed `ridge/` fit together with the residual. The Ministral-3-8B→14B checkpoint trains the original six-source residual over its fixed Ridge cache with receiver cross-entropy plus forward KL. The Qwen3-8B→14B and Yi-1.5-9B→34B checkpoints likewise train zero-initialized late K/V residuals over fixed pair-specific Ridge caches with receiver cross-entropy plus forward KL. Each of the latter two runs uses 1,200 training examples, two per optimizer step for 600 steps, drawn from CommonsenseQA, CosmosQA, SQuAD v2, DROP, MultiDoc2Dial, and ContractNLI. These are training sources, not additional evaluation panels.

Why CE + forward KL? We selected receiver continuation cross-entropy plus forward KL as the shared functional objective based on an earlier controlled loss ablation for Qwen3-14B→32B. This pilot used the earlier CacheLink architecture for 150 optimizer steps while holding the training data, fixed Ridge fit, model structure, and optimization settings constant. All variants were evaluated on the same 100-question NeedleBench MR32K set with a 4,096-token output cap and strict boxed-answer matching. Table 2 reports the archived results.

The ablation separates the roles of the two functional terms. CE alone already improves over Ridge and the native sender, showing that continuation supervision directly teaches the transported state to support the target task. KL alone is much weaker, but adding forward KL to CE improves the score from 37 to 40, suggesting that matching the native receiver distribution supplies useful supervision beyond the gold continuation. Forward KL also performs slightly better than reverse KL in the matched CE setting (40 versus 39). KL without CE performs substantially worse. We therefore use CE + forward KL as the functional objective in the main experiments for all four model pairs.

Fixed evaluation panels. All reported evaluation results use the seven panels in Table 1. SciQ, MMLU-Pro, MMLU-Pro-Plus, and OpenBookQA use their complete selected test splits. Oolong-Synth uses the fixed 100-question 32K panel for Qwen3-14B→32B and Ministral-3-8B→14B, and a fixed 400-question panel for Qwen3-8B→14B and Yi-1.5-9B→34B. LongBench-v2 uses the pair-specific frozen 8K–32K tokenized panel; BABILong uses the 500-example 32K QA1–QA5 panel. Exact tokenized-panel hashes and completion records are retained with the evaluation artifacts. MMLU-Pro and MMLU-Pro-Plus share question stems. Different task metrics, panel sizes, tokenizers, and preprocessing mean cross-panel means are descriptive summaries, not pooled accuracy.

Cross-pair Ridge utility. Figure 8 reports the complete seven-panel comparison for all four sender–receiver pairs. Ridge is competitive with the native receiver on some panels and falls below the native sender on others despite using the same mapping family throughout. Averaged across the four short-prefix and three long-prefix panels, the receiver-minus-Ridge gaps are 1.38 and 9.03 points for Qwen3-14B→32B, 8.21 and 4.63 for Ministral-3-8B→14B, 10.18 and 19.71 for Qwen3-8B→14B, and 17.28 and 16.64 for Yi-1.5-9B→34B. The variation is therefore pair- and task-dependent rather than a uniform long-context failure.

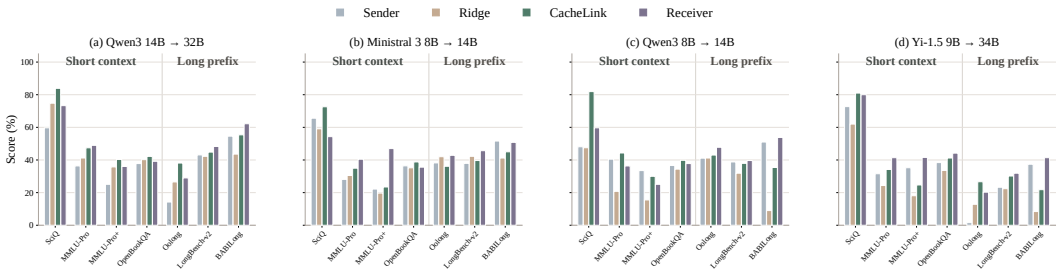

Figure 8: **Receiver utility under cross-model handoff is strongly task- and pair-dependent.** Each subplot shows the native sender, Ridge-$k8$, CacheLink, and native receiver on the same seven fixed panels. SciQ, MMLU-Pro, MMLU-Pro-Plus, and OpenBookQA are short-prefix multiple-choice panels; Oolong-Synth 32K, LongBench-v2, and BABILong 32K use long reusable prefixes.

In-house baseline scope. We reproduce CacheBridge (Qu et al., 2026) and LatentAlign (Dery et al., 2026) as cross-model KV transport baselines under the same zero-prefill evaluation protocol used for Ridge and CACHELINK. Table 1 reports their matched results on the same model pairs, inputs, prompts, decoding rules, and scorers.

Execution. Custom per-layer cache injection is not supported by the standard vLLM generation interface used in our environment. We therefore evaluate with Transformers and FlashAttention-2. The 14B sender and 32B receiver reside on separate H100 GPUs; a one-slot pipeline overlaps preparation of the next sender cache with receiver work on the current item. This changes throughput, not the tokenized inputs, decoding configuration, or scorer.

Matched mechanism diagnostic. For the Qwen3-14B→32B pair, we select 50 examples from each of the seven fixed evaluation panels with seed 20260926, stratifying by category or task where metadata permits. This is a diagnostic subset and does not replace the complete main-table scores. On each example we compare the native receiver cache, the fixed Ridge cache used to initialize CacheLink, and the final cache after applying the step-600 residual.

We compute post-RoPE K and raw-V mean squared error, relative squared L_2 error, and global cosine similarity against the native cache. For the learned correction $\Delta = C_{\mathrm{CacheLink}} - C_{\mathrm{Ridge}}$, we report its RMS relative to the Ridge cache, its cosine with $C_{\mathrm{Native}} - C_{\mathrm{Ridge}}$, and the fraction of K/V squared error removed. We also compare final-query-token attention over reusable-prefix positions after renormalizing those probabilities within the prefix, using Jensen–Shannon divergence, entropy, and top-10 attention mass. This conditional distribution does not measure the total fraction of attention allocated to the prefix versus query tokens. The next-token distributions are compared by forward KL, Jensen–Shannon divergence, and top-1 agreement. These are token- and cache-level diagnostics: they do not measure full-generation accuracy. Multiple-choice answer scores are additionally computed for the four short-prefix panels using both raw and length-normalized continuation likelihood.

B SUPPORTING DIAGNOSTIC ANALYSES

The following analyses provide additional evidence for three findings developed in the main text: Ridge already captures substantial linear sender–receiver KV structure; tensor reconstruction quality can remain nearly unchanged while task utility changes sharply; and a functionally useful residual need not behave like another step toward the native receiver cache. These results are used as targeted diagnostics rather than aggregate benchmark results.

Ridge provides a strong reconstruction base. The NeedleBench MR32K layerwise diagnostic measures how much receiver KV structure can be predicted linearly from the sender cache before any functional residual is introduced. Across the 64 receiver layers, mean held-out R^2 is 0.838 for K and 0.609 for V, with mean NMSE 0.108 and 0.364, respectively. The important point is not that Ridge reconstructs every layer perfectly, but that a large fraction of receiver-cache structure is already

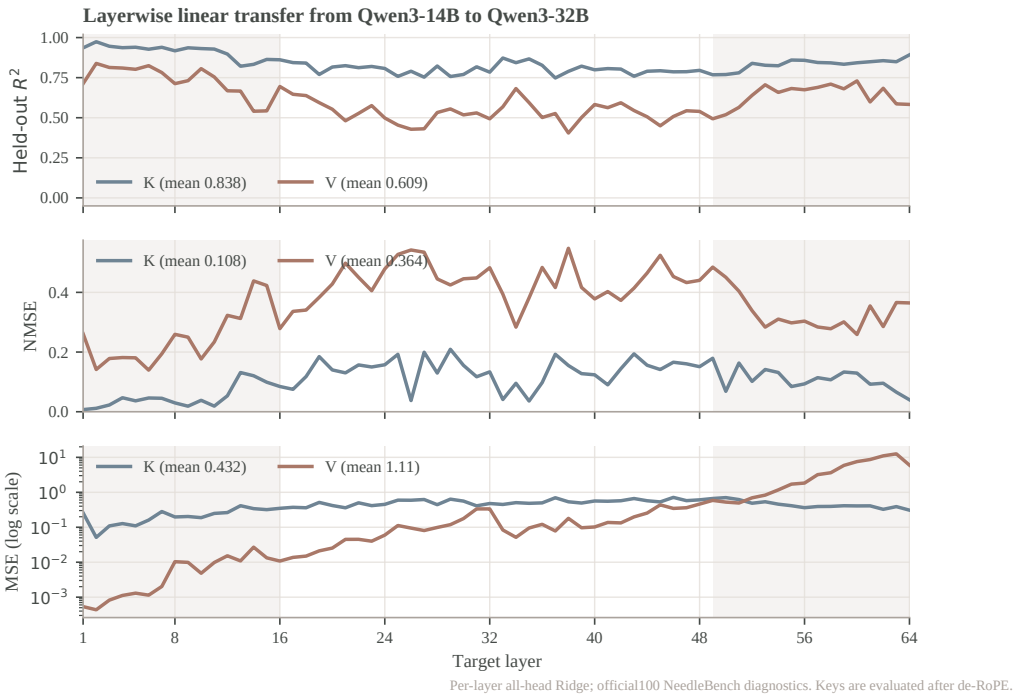

Figure 9: **Layerwise cache predictability for Qwen3-14B→32B on NeedleBench MR32K.** The plot reports held-out K/V reconstruction statistics by receiver layer.

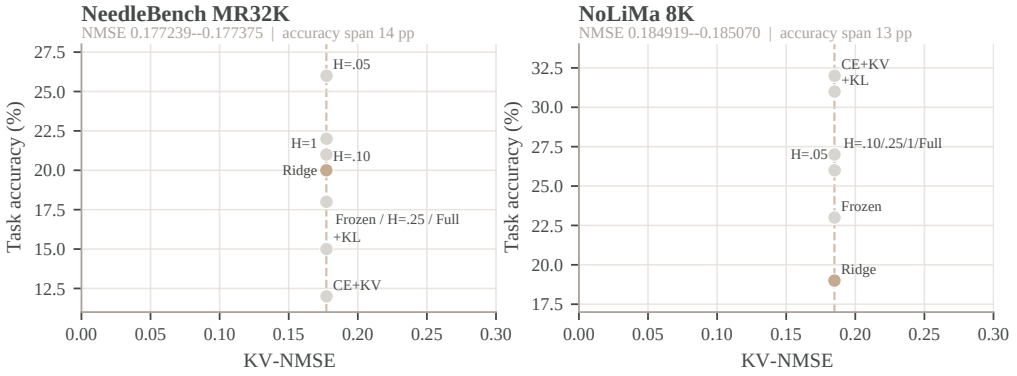

Figure 10: **Mapper-variant comparison on NeedleBench MR32K and NoLiMa 8K.** Similar aggregate KV-NMSE can accompany substantially different task scores.

captured by a simple affine map. This observation motivates CacheLink's residual design: preserve the strong Ridge reconstruction and learn only what receiver behavior still requires.

Nearly identical tensor error can correspond to very different task utility. The second pilot compares nine archived mapper variants while holding the evaluation task fixed. On NeedleBench MR32K, their KV-NMSE values occupy only 0.177239–0.177375, yet accuracy ranges from 12% to 26%. On NoLiMa 8K, KV-NMSE occupies 0.184919–0.185070 while accuracy ranges from 19% to 32%. Thus a mapper can move substantially in task quality without a correspondingly visible change in aggregate tensor error. This pilot supplied the first direct evidence for the reconstruction–utility mismatch; the matched seven-panel diagnostics in Section 4.2 test the same idea more systematically.

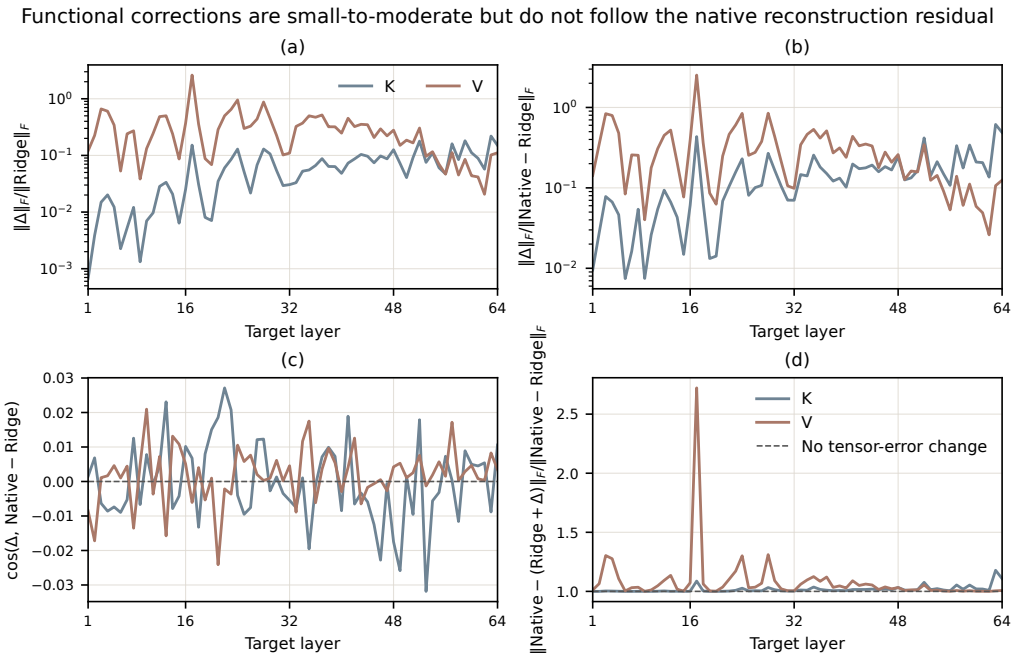

Figure 11: **Layerwise residual analysis on four held-out 32K LongABC documents.** The panels show correction magnitude, alignment with the Native-minus-Ridge direction, and the change in Euclidean error.

Functional residuals need not be reconstruction residuals. The third pilot asks whether receiver-level training merely takes an additional step from Ridge toward the native receiver tensors. Its four panels measure the learned correction relative to the Ridge cache, relative to the Native-minus-Ridge gap, its cosine with the Native-minus-Ridge direction, and the post-/pre-correction Euclidean error ratio. If functional training were simply improving reconstruction, the correction would be expected to align with Native-minus-Ridge and systematically reduce the final tensor error. Instead, the layerwise analysis shows that useful corrections can remain small-to-moderate while not consistently following that reconstruction direction. This observation motivated the cleaner fixed-panel residual-direction analysis in Figure 3.