KVRELAY: REUSING KV CACHES ACROSS LANGUAGE MODELS WITHOUT RE-PREFILLING

Anonymous authors
Paper under double-blind review

ABSTRACT

Deployed language models reuse context far more often than they compute it: the same document, prompt or session is encoded once and read back many times, which turns its KV cache into stored infrastructure. Model iteration invalidates that infrastructure. The day a deployment upgrades, every cached context still lives in the previous model's attention space, and the new model has to prefill all of it again. We ask whether a target model can *inherit* a KV cache produced by another model and continue inference without re-prefilling the context. The obstacle is that a cache is an interface rather than a portable encoding of text: even between models that share a tokenizer, depth and KV geometry, injecting the source cache directly leaves the target below its own no-context accuracy on four of five pairs. We introduce KVRelay, which treats cross-model reuse as interface adaptation. It supervises a translated cache by the computation it induces inside frozen target attention rather than by element-wise reconstruction of the target's native KV tensors; it starts from a parameter-free, target-calibrated base map and learns a zero-initialized cross-head correction; and it gives the Value branch a short causal mixer over source history. Across five Qwen/Llama model pairs and five context-dependent benchmarks, KVRelay recovers $90.3 \pm 1.0\%$ of the target's native context gain over five seeds with a translator worth 0.3–5.1% of target parameters; replacing functional supervision with direct KV MSE reduces five-pair mean accuracy by 4.88 percentage points. Complete translation costs 13.6% of target prefill on average and reduces end-to-end time to first token by $3.41\times$. KVRelay turns a model upgrade from a full cache rebuild into a cache handoff.

1 INTRODUCTION

Production LLM systems are moving beyond a single, static model. The model behind a service is continually post-trained, specialized, patched, and replaced; at the same time, multi-agent workflows increasingly route a task among models with different capabilities (Ong et al., 2025; Wu et al., 2023; Hong et al., 2024). Across both settings, the models change more often than the context they consume. Documents remain in an index across model releases, and system prompts, tool specifications, retrieved evidence, and dialogue histories flow from one agent to the next.

KV caching preserves the computation invested in such recurring context (Gim et al., 2024), but only as long as the consuming model remains unchanged. After a model upgrade, long-lived document caches are stranded in the old model's attention space; after an agent handoff, the receiver must prefill context that the sender has already processed. This mismatch suggests a natural possibility: rather than repeatedly reconstructing the same context state, can the target inherit the source model's KV cache and continue inference without re-prefilling the context?

Shape compatibility is not interface compatibility. Even checkpoints sharing a tokenizer, depth, KV-head count, and head dimension need not use their caches alike, despite evidence for transferable cross-model representations (Huh et al., 2024; Huang et al., 2025; Bello et al., 2025; Chen et al., 2025). Indeed, raw cache injection is worse than no context on four of our five pairs, lowering mean accuracy from 55.63 to 45.15. Existing approaches either recompute selected layers or tokens (Liu et al., 2026; Yao et al., 2025), fuse into a receiver cache that has already been computed (Fu et al., 2025), or replay context through the target after translation (Lee et al., 2026). We instead require a source-only handoff: the target performs *no context prefill, recomputation, or replay*.

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.

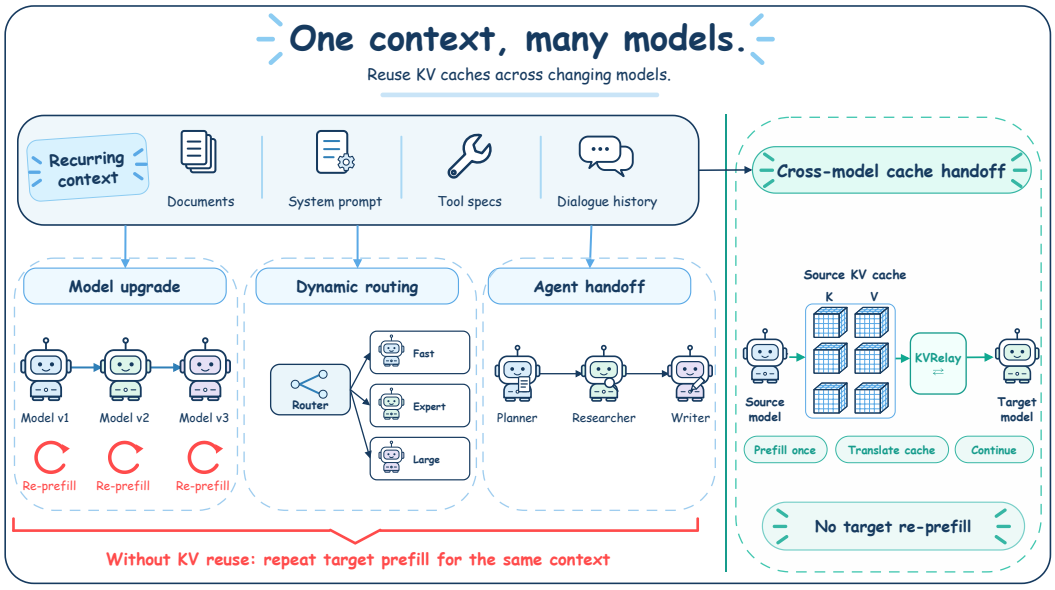

Figure 1: **Cross-model KV cache handoff with KVRelay.** Recurring contexts—such as documents, system prompts, tool specifications, and dialogue histories—are repeatedly processed when models change due to upgrades, dynamic routing, or agent handoffs. Conventional serving re-prefills the shared context for every target model, whereas KVRelay translates the source model's existing KV cache into the target model's attention interface, enabling generation without target-side context prefill or replay.

KVRelay. A cache influences target predictions through the attention computation it induces, so matching its coordinates need not preserve its function. We therefore supervise the attention outputs induced inside the frozen target. Our compact translator starts from a source-conditioned, moment-calibrated base, adds a zero-initialized cross-head correction, and lets the Value branch read a short causal history. Across five Qwen/Llama pairs and five context-dependent benchmarks, it recovers $90.3 \pm 1.0\%$ of the target's native context gain over five seeds with 0.3–5.1% of target parameters. Our contributions are:

- We formulate *cross-model cache handoff* under a source-only protocol with no target context prefill or replay, and show that shape-compatible caches can be functionally incompatible enough to underperform no context.
- We introduce a compact cache-interface adapter that combines canonical positional alignment, attention-output supervision, a calibrated base with zero-initialized cross-head correction, and causal Value mixing.
- Across five model pairs and five benchmarks, we recover $90.3 \pm 1.0\%$ of the target's native context gain while achieving a $3.41\times$ mean end-to-end time-to-first-token speedup; controlled ablations attribute $4.88\,pp$ and $6.10\,pp$ to functional supervision and the calibrated start, respectively.

Organization. We present the method, experiments, limitations, and conclusion in sequence; proofs, detailed results, and implementation details are deferred to the appendix.

2 Related Work

Reusing a cache within one model. Prefix caching and its position-independent extensions address the case where the *same* model re-encounters text it has already encoded: CacheBlend recomputes a few high-deviation tokens to restore cross-chunk attention (Yao et al., 2025), EPIC recomputes only the first tokens of each chunk to repair a spurious attention sink (Hu et al., 2025), and KVLink uses trainable link tokens to connect cached segments (Yang et al., 2025). They keep the

producing model fixed, and are complementary to ours: they make a cache reusable across positions, not across models.

Reusing a cache across models. DroidSpeak recomputes a few sensitive layers when sharing caches among fine-tunes of one base (Liu et al., 2026); ICaRus retrains experts so that they emit identical caches by construction (Woo et al., 2026). Closest to our setting, a closed-form Ridge mapper regresses target KV coordinates from a top-k selection of source layers per head, and observes that attention-output similarity predicts downstream retention better than coordinate R^2 (Heo et al., 2026)—a diagnostic our objective takes literally by *optimizing* that quantity rather than measuring it post hoc. Mixture-of-Translators routes each token to a mixture of translators and corrects the target's hidden trajectory by replaying context (Lee et al., 2026); we keep one translator per pair, need no replay, and place cross-token aggregation *inside* the Value branch.

Caches as a communication channel. Cache-to-Cache fuses a sharer's projected cache residually into a cache the receiver has already computed (Fu et al., 2025); Latent Cache Flow compresses that channel (Rossi et al., 2026); dense-alignment work formalizes a *context-unaware* protocol and reconstructs the receiver's own cache before generation fine-tuning (Chen et al., 2026); latent-space communication learns a globally shared KV space via per-model adapters (Dery et al., 2026). More generally, semantic alignment has been used to translate between independently learned latent spaces (Maiorca et al., 2023). Ours is pure *source-only translation*: the target never prefills the context, so there is no receiver cache to fuse into and no interlingua to learn; the fixed base map remains source-conditioned and supervision is functional rather than reconstructive.

3 METHOD

3.1 PROBLEM SETUP

Let M_s and M_t be a frozen source and target model sharing a tokenizer, and let $C = (c_1, \ldots, c_N)$ be a context. The source prefills once,

$$KV_s = M_s^{\text{prefill}}(C), \qquad KV_s \in \mathbb{R}^{B\times L_s\times 2\times H_s\times N\times D_s}, \tag{1}$$

where the third axis indexes the Key and Value branches. The returned source Keys already contain source-model rotary position encoding. We invert the source model's native rotary embedding to obtain canonical Keys, translate in that frame, and apply the target model's native rotary embedding before cache injection; Values are unchanged by this positional conversion. A translator $\text{Trans}_\theta : KV_s \mapsto \widehat{KV}_t$ produces a cache installed as the target's context prefix, after which the target processes only the short post-context sequence consisting of the query Q and answer Y: $P_t(Y \mid Q, C) = M_t(Q, Y; \widehat{KV}_t)$. Both models stay frozen; gradients flow through the frozen target attention into θ only. Throughout, $KV_t = M_t^{\text{prefill}}(C)$ denotes the target's *native* cache, available at training time as supervision and unavailable at deployment time — if it were available there would be nothing to translate. We consider pairs with a shared tokenizer and compatible KV geometry, and return to this boundary in Section 6.

3.2 A CACHE IS AN INTERFACE, NOT A TENSOR

Write one attention layer as $O = A(K)V$ with $A(K) = \text{softmax}(QK^\top/\sqrt{d})$. To first order, translation errors $\Delta K, \Delta V$ propagate as

$$\Delta O \;\approx\; D\,\text{softmax}_S\Big[\tfrac{1}{\sqrt{d}}\,Q\,\Delta K^\top\Big]V \;+\; A\,\Delta V, \qquad S = \tfrac{1}{\sqrt{d}}QK^\top, \tag{2}$$

where $D\,\text{softmax}_S[\cdot]$ is the Fréchet derivative of row-wise softmax at S, applied row by row to the bracketed perturbation. Thus Keys reach the output only through a softmax derivative while Values enter linearly: the two branches possess different *equivalence classes*, not merely different difficulty.

Proposition 1 (Routing equivalence) *For any $c \in \mathbb{R}^d$, setting $K'_j = K_j + c$ for all j leaves attention unchanged:* $\text{softmax}(q_i K'^\top) = \text{softmax}(q_i K^\top)$ *for every query q_i.*

The proof is one line (Appendix A): the extra term $q_i^\top c$ is constant along row i, and softmax is shift-invariant. The identifiable object on the Key side is thus an equivalence class. Let $\widehat{K}_t$ denote the translated Key cache, produced from the source cache to approximate the target's native Key cache K_t. For target queries Q_t, the routing discrepancy $\mathcal{D}\big(A(Q_t, \widehat{K}_t), A(Q_t, K_t)\big)$ is therefore a more direct functional error than $\|\widehat{K}_t - K_t\|_2^2$. This invariance applies to a uniform shift of all Keys visible in a given attention window in the injected target-space coordinates; it is not a claim of invariance to an arbitrary shift in the canonical pre-RoPE frame. Element-wise Key MSE penalizes even those differences that provably do not change routing. Values admit no such invariance — $A(V + \Delta V) = AV + A\Delta V$ — so their errors survive unless they lie in the null space of A. We stress what Eq. (2) does *not* say: it does not make Values unconditionally harder than Keys, since large query norms can let a small ΔK change routing substantially. We use the asymmetry only to explain why element-wise KV MSE and functional error are not equivalent; we do not build a strongly asymmetric architecture on it, and Section 5.2 finds such asymmetry empirically irrelevant here.

3.3 Functional alignment

Let $\Phi_l(K, V) = \mathrm{SelfAttn}_l(Q_l, K, V)$ be the complete output of the target's l-th attention module, output projection included. We run the frozen target twice on the same post-context sequence—the query followed by the gold answer under teacher forcing—once with its native cache KV_t (*target path*, stop-gradient) and once with $\widehat{KV}_t$ (*trans path*). The objective is a layer-wise, energy-normalized discrepancy over all valid post-context positions,

$$\mathcal{L}_{\mathrm{attn}} = \frac{1}{L_t} \sum_{l=1}^{L_t} \frac{\sum_{b,n} m_{bn} \big\| \Phi_{l,bn}^{\mathrm{trans}} - \Phi_{l,bn}^{\mathrm{target}} \big\|_2^2}{\sum_{b,n} m_{bn} \big\| \Phi_{l,bn}^{\mathrm{target}} \big\|_2^2 + \varepsilon}, \tag{3}$$

where m_{bn} masks padding. Per-layer normalization prevents high-energy layers from dominating. The complete loss of the main method is $\mathcal{L} = \mathcal{L}_{\mathrm{attn}}$: we use neither KV regression nor answer-logit distillation. The objective is functional in a precise sense — it constrains the cache only through the computation it induces — and it turns out to be the single largest source of accuracy in our ablations.

3.4 Base-and-correct parameterization

Figure 2 sketches the resulting pipeline. Let $R_s(p)$ and $R_t(p)$ denote the source and target rotary operators at position p. The Key-side translator consumes $\bar{K}_s(p) = R_s(p)^{-1} K_s(p)$, produces a canonical $\widehat{\bar{K}}_t(p)$, and injects $\widehat{K}_t(p) = R_t(p) \widehat{\bar{K}}_t(p)$. We call this de-RoPE/translate/re-RoPE procedure *canonical positional alignment*. It uses each model's native rotary module, including configured scaling, rather than reimplementing its frequency schedule. For target layer l, branch $u \in \{K, V\}$ and source layer $\pi(l)$, we decompose the translated cache in the canonical space for Keys (and the ordinary value space for Values) into a fixed base mapping and a learnable residual correction:

$$\widehat{U}_l^u = \mathcal{B}_l^u \Big(U_{s,\pi(l)}^u \Big) + \alpha_l^u \, \sigma_{t,l}^u \odot \Delta_l^u, \tag{4}$$

where the residual is

$$\Delta_l^u = \mathrm{reshape}_t \Big[W_l^u \Big(\mathcal{T}^u \Big(\mathrm{flat}_s(U_{s,\pi(l)}^u) \Big) \Big) \Big]. \tag{5}$$

Here, $\mathcal{B}_l^u$ denotes the moment-matched base mapping, α_l^u is a learnable residual scale, and $\sigma_{t,l}^u$ places the correction on the target cache scale. The operator $\mathrm{flat}_s : \mathbb{R}^{B \times H_s \times N \times D_s} \to \mathbb{R}^{B \times N \times H_s D_s}$ concatenates all source KV heads at each position, while $\mathrm{reshape}_t$ arranges the linear output according to the target KV-head geometry; it is not the inverse of flat_s. All five pairs satisfy $L_s = L_t$, $H_s = H_t$, and $D_s = D_t$, so the two terms in Eq. (4) have the same geometry and $\pi(l) = l$. Unequal KV geometries would additionally require mapping the base into the target geometry.

Moment-matched base map. The base map matches the first two marginal moments of the target cache channel-wise (up to the numerical floor below),

$$\mathcal{B}_l^u(U) = \mu_{t,l}^u + \sigma_{t,l}^u \odot \frac{U - \mu_{s,\pi(l)}^u}{\max\big(\sigma_{s,\pi(l)}^u, \varepsilon\big)}, \tag{6}$$

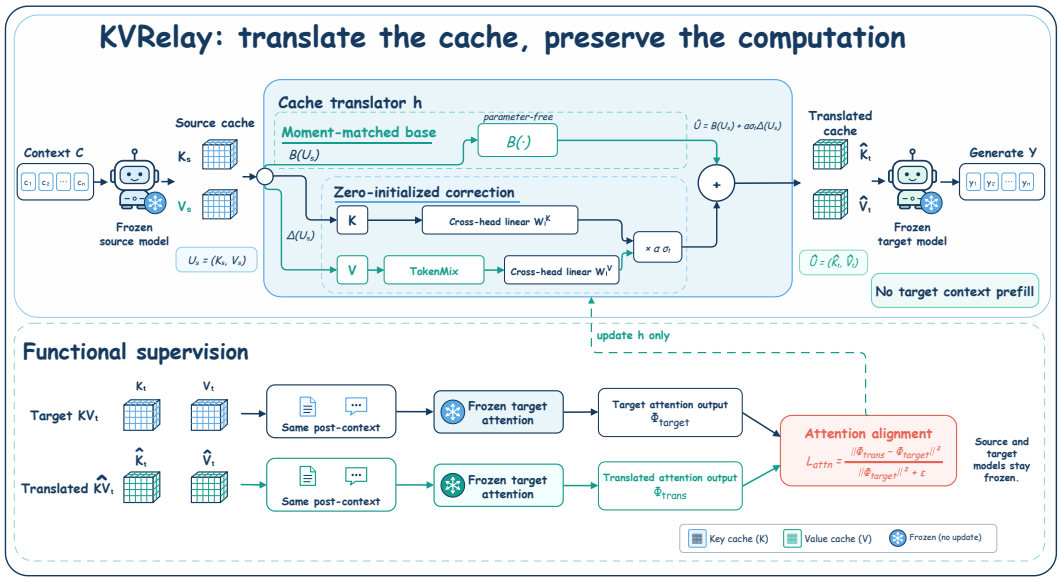

Figure 2: **Architecture and training pipeline of KVRelay.** Given a source KV cache, KVRelay first performs canonical positional alignment by de-rotating source Keys and re-applying the target model's native RoPE after translation. At each layer, the translator combines a parameter-free, moment-matched base mapping with a zero-initialized cross-head correction, while the Value branch additionally applies causal TokenMix. During training, only the translator is updated by aligning the attention outputs induced by translated and native target caches under the frozen target model; at inference time, the translated cache is injected directly, eliminating target-side context prefill.

so that $\mathbb{E}[\mathcal{B}_l^u(U_s)] = \mu_{t,l}^u$ and, whenever $\sigma_{s,\pi(l)}^u \geq \varepsilon$ channel-wise, $\mathrm{Var}[\mathcal{B}_l^u(U_s)] = (\sigma_{t,l}^u)^2$. When the floor is active, the mean is still matched exactly while the output variance is reduced by the corresponding stabilization factor. Statistics have shape $[L, 2, D_h]$: estimated per layer, per branch and per within-head channel, but pooled over samples, KV heads and valid tokens, so the base map never presumes a head-to-head correspondence. They are collected from 32 mini-batches and stored as non-trainable buffers inside the checkpoint, so training and evaluation always share identical statistics. Under a diagonal-affine restriction and away from floored channels, this is the unique positive map matching per-channel mean and variance; it is a *scale prior*, not a claim that whitening recovers the target representation.

Cross-head correction. $W_l^u : \mathbb{R}^{H_s D_s} \to \mathbb{R}^{H_t D_t}$ is a single affine map per layer and branch — no hidden layer, no activation — mixing all source heads jointly. Independently trained models have no reason to place the same semantics in the same head index, so we *permit* cross-head interaction instead of *enforcing* a diagonal correspondence. Weights and bias start at zero and the per-branch gate α_l^u at 1, hence $\Delta_l^u = 0$ at step 0 and the translator output is exactly the calibrated base: a deterministic source-conditioned initialization, plus whatever correction the downstream objective demands.

3.5 Context-capable Value translation

A pointwise translator sees only $V_{l,i}^s$, whereas $V_{l,i}^t = W_l^{V,t} h_{l,i}^t$ with $h_{l,i}^t = F_l^t(x_{\leq i})$ the target's own aggregate of the entire prefix. Two models may summarize the same tokens differently, which bounds what any per-position map can achieve.

Proposition 2 (Context cannot hurt the Bayes-optimal translator) *Assume finite second moments and define the conditional total variance of a vector as* $\mathrm{Var}(X \mid \mathcal{G}) = \mathbb{E}[\|X - \mathbb{E}[X \mid \mathcal{G}]\|_2^2 \mid \mathcal{G}]$. *Under squared loss the optimal pointwise and history-aware predictors incur* $\mathcal{E}_{\mathrm{point}} = \mathbb{E}\big[\mathrm{Var}(V_{l,i}^t \mid V_{l,i}^s)\big]$ *and* $\mathcal{E}_{\mathrm{ctx}} = \mathbb{E}\big[\mathrm{Var}(V_{l,i}^t \mid V_{l,\leq i}^s)\big]$, *and* $\mathcal{E}_{\mathrm{ctx}} \leq \mathcal{E}_{\mathrm{point}}$. *The inequality is strict exactly when* $\mathbb{E}[V_{l,i}^t \mid V_{l,\leq i}^s]$ *differs from* $\mathbb{E}[V_{l,i}^t \mid V_{l,i}^s]$ *on a set of positive probability.*

Table 1: Model pairs. "Params" is the translator size and the last column its ratio to the target model. Pair 4 is same-lineage and acts as a reference point for how far naive reuse can go when two models share training history.

#	Source	Target	Transfer setting	$L/H_{kv}/D_h$	Params	/target
0	Qwen3-0.6B	Qwen3-1.7B	cross scale	28/8/128	88.4M	5.1%
1	Qwen2.5-Coder-7B	Qwen2.5-7B-Instruct	code → instruct	28/4/128	22.2M	0.3%
2	Qwen2.5-Math-7B	Qwen2.5-7B-Instruct	math → instruct	28/4/128	22.2M	0.3%
3	Qwen2.5-Math-7B	Qwen2.5-Coder-7B	math → code	28/4/128	22.2M	0.3%
4	Llama-3.1-8B	ToolACE-8B	same lineage	32/8/128	101.0M	1.3%

This squared-loss result diagnoses whether source history contains additional information about the native target Value; it is not the training objective, which remains the functional loss in Eq. equation 3. It motivates — but does not guarantee — a history-aware Value branch: the gap $\mathcal{E}_{\text{point}} - \mathcal{E}_{\text{ctx}}$ is realized only if history changes the conditional mean, a finite receptive field covers the relevant history, optimization exploits it, and the downstream metric is sensitive to what remains. We therefore parameterize the capability and then test it. KVRelay keeps $\mathcal{T}^K(x) = x$ and inserts a length-preserving mixer on the Value branch only:

$$\mathcal{T}^V(x) = x + \text{Proj}\Big(\text{DWConv}_{k=4}^{\text{causal}}(\text{LN}(x))\Big). \tag{7}$$

The depthwise convolution acts per channel over $\{i-3, \ldots, i\}$ with left padding, and the pointwise projection restores channel mixing; padding is masked before mixing so that it cannot leak into valid tokens. The kernel is initialized to the identity filter and the outer residual stays zero-initialized, so adding the mixer does not perturb the calibrated start. Causality is a deployment choice: anti-causal and bidirectional kernels are legitimate offline references, but they require the whole context to be available before any position can be translated, whereas a causal kernel keeps translation incremental. Including its pointwise projection, the mixer costs $O(Nd_sk + Nd_s^2)$ for flattened source width $d_s = H_sD_s$, which is linear in sequence length N. Section 5.5 sweeps kernel size and direction.

4 Experimental setup

Model pairs. Table 1 lists five pairs spanning parameter-scale transfer, cross-domain transfer and same-lineage transfer. All pairs share a tokenizer and have equal layer counts and compatible KV-head geometry, so we use the identity layer mapping $\pi(l) = l$.

Training data. We train on 20,000 context-grounded multiple-choice examples drawn from BoolQ, PubMedQA, MultiRC, RACE-high, QASC, ReClor and OpenBookQA. The reusable context is stored separately from the question, options and answer cue. Exact text fingerprints remove both within-source duplicates and examples duplicated in the corresponding evaluation split. We use the same data and optimization protocol for all controlled comparisons. Full construction, training and model-selection details are given in Appendix F.

Evaluation and metrics. We evaluate on the complete evaluation splits of five context-dependent multiple-choice benchmarks: BoolQ validation, LogiQA test, PubMedQA labeled, MultiRC validation and ReClor validation. They contain 3270, 651, 1000, 1725 and 500 examples, respectively. We report accuracy for each benchmark and the unweighted mean across all five. LogiQA is excluded from the training mixture and thus also measures cross-task generalization.

Since target models differ in how much they benefit from context, we use *context recovery* as the primary measure of cache translation quality:

$$\text{rec} = \frac{\text{Acc}_{\text{trans}} - \text{Acc}_{\text{no-ctx}}}{\text{Acc}_{\text{target}} - \text{Acc}_{\text{no-ctx}}}. \tag{8}$$

A recovery of zero means that the translated cache provides no improvement over removing the context, whereas a recovery of one means that it preserves the complete gain provided by the target's

Table 2: Five-task mean accuracy (%) and context recovery. Point estimates use the common seed retained for paired comparisons with all ablations; five-seed statistics appear in Appendix D. **naive** is the raw source cache. On the four independently trained pairs an untranslated cache is *worse than no context at all*.

Pair	no-ctx	naive	source	Ridge	**KVRelay**	target	**rec**
Qwen3-0.6B → 1.7B	52.07	39.97	53.56	58.39	**64.16**	66.82	**0.820**
Coder-7B → 7B-Instruct	57.06	34.02	71.40	65.97	**75.97**	77.29	**0.935**
Math-7B → 7B-Instruct	57.06	33.85	67.33	67.16	**74.60**	77.29	**0.867**
Math-7B → Coder-7B	56.69	47.34	67.33	67.56	**69.75**	71.40	**0.888**
Llama-3.1-8B → ToolACE-8B	55.28	70.56	68.74	70.88	**71.46**	71.08	**1.024**
Mean	55.63	45.15	65.67	65.99	**71.19**	72.78	**0.907**

native cache. Across our five model pairs, the denominator ranges from 14.71 to 20.23 percentage points and is therefore well separated from zero. We additionally report absolute accuracy to ensure that the normalized score does not obscure task performance.

We compare KVRelay against five reference settings. **Target** uses the target model's native cache and serves as the upper reference. **No-ctx** removes the context from the target and provides the lower reference. **Naive** injects the unmodified source cache, while **Source** evaluates the source model with its native cache. **Ridge** is a closed-form linear mapping baseline described in Appendix F.

5 Results

5.1 Main results

Figure 3 visualizes the headline comparison; Table 2 reports the exact values.

Untranslated reuse is worse than useless. On the four independently trained pairs, **naive** injection scores 33.85–47.34, i.e. 9–23 pp *below* the same target answering with no context at all. The exception is instructive: Llama-3.1-8B→ToolACE-8B is a base model and its own fine-tune, and there naive reuse already reaches 70.56 out of 71.08. Functional compatibility tracks shared training history, not shared tensor shapes.

KVRelay recovers most of the target's context gain. Across five seeds, mean accuracy is 71.14 ± 0.17 and mean recovery is 0.903 ± 0.010 (Appendix D). The paired comparison in Table 2 yields 0.820–1.024 recovery (mean 0.907) and 71.19 mean accuracy against a 72.78 native-cache reference. It also outperforms the *source* model answering with its own cache (65.67) on every pair—the practically relevant comparison for escalation, since after handoff the stronger target answers using the weaker model's context.

Closed-form linear mapping is a real but limited baseline. Ridge averages 65.99, while KVRelay improves the mean by $5.20\,pp$ and performs better on all five model pairs. The gain is smallest on the nearly saturated same-lineage pair, where the two methods differ by $0.58\,pp$. Scaling Ridge calibration from 0.128M to 1.946M tokens did not improve it, so the overall gap is not explained by calibration data.

5.2 Supervision and initialization

Replacing functional alignment with direct KV MSE loses $4.88\,pp$ on average and is worse in all 25 pair×task cells (per-task results in Appendix C), consistent with Section 3.2: tensor reconstruction penalizes Key differences that need not change attention. Removing the calibrated base loses 3.24–$9.40\,pp$ on every pair ($6.10\,pp$ mean). The exact base family remains unresolved—identity and moment calibration differ by at most $0.58\,pp$ in Table 4—but removing zero-initialization costs $5.55\,pp$ on the cross-scale pair.

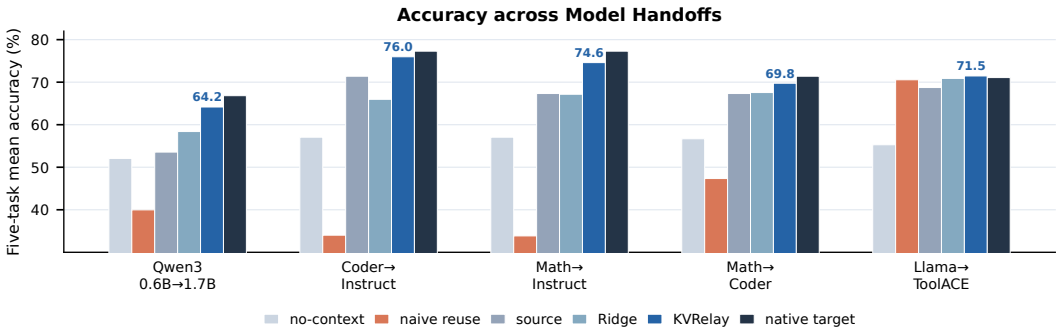

Figure 3: **Cross-model cache handoff results.** Five-task mean accuracy for every model pair and reference setting, using the same paired-comparison seed as Table 2. KVRelay approaches the native target while consistently outperforming both the source model and the closed-form Ridge translator.

Table 3: Controlled supervision and initialization ablations (five-task mean accuracy, %). **feat** uses canonical-space KV MSE; **legacy** uses attention+KL+context contrast; **zero** removes the calibrated base. Architecture, data, budget, and seed are matched within each comparison.

Pair	target	**KVRelay**	feat	legacy	zero
Qwen3-0.6B → 1.7B	66.82	**64.16**	56.47	56.40	60.92
Coder-7B → 7B-Instruct	77.29	**75.97**	68.16	74.43	68.45
Math-7B → 7B-Instruct	77.29	**74.60**	70.21	74.80	65.20
Math-7B → Coder-7B	71.40	**69.75**	65.92	69.60	62.81
Llama-3.1-8B → ToolACE-8B	71.08	**71.46**	70.77	69.70	68.05
Mean	72.78	**71.19**	66.31	68.99	65.09
Δ vs. KVRelay		—	−4.88	−2.20	−6.10

Table 4: Single-variable ablations on Qwen3-0.6B→1.7B (five-task mean accuracy, %). Data, loss, and seed are fixed; Value-mixing variants are discussed in Section 5.5.

Group	Variant	Params	Mean	Δ	Value-mixing variant	Params	Mean	Δ
KVRelay: calibrated base, cross-head linear, causal k=4		88.4M	**64.16**	—				
Base	identity V	88.4M	63.93	−0.23	none (affine)	58.8M	63.22	−0.94
	identity K	88.4M	63.58	−0.58	causal k=2	88.3M	64.46	+0.30
	identity K,V	88.4M	63.88	−0.28	causal k=8	88.5M	64.29	+0.14
Init.	random DW kernel	88.4M	64.07	−0.09	causal k=16	88.7M	63.68	−0.47
	no residual zero-init	88.4M	58.61	**−5.55**	causal k=64	90.1M	64.11	−0.05
	zero	88.4M	60.92	**−3.24**	anti-causal k=4	88.4M	63.49	−0.66
Capacity	3 linear layers	206.0M	63.73	−0.43	bidirectional k=4	88.4M	64.24	+0.09
	3-layer GELU MLP	206.0M	62.00	**−2.16**	causal cumulative mean	88.3M	64.12	−0.04
Heads	per-head (block diagonal)	44.4M	60.97	**−3.19**	global mean	88.3M	64.20	+0.04
	one map shared across heads	31.5M	56.62	**−7.53**				

5.3 Heads should be allowed to mix; capacity should not be added

Constraining the residual to be block diagonal across heads costs 3.19 *pp*, and sharing one map across all heads costs 7.53 *pp* (Table 4): independently trained models do not agree on head semantics, and a translator required to respect head indices pays for that assumption. The converse experiment is equally clear. Three stacked linear layers change nothing (−0.43 *pp*), while inserting GELU costs 2.16 *pp* at 2.33× the parameters. In this regime, cache translation is a well-conditioned near-linear problem that mainly needs the right origin and the right supervision.

5.4 Measured handoff cost

Table 5 measures the deployment regime that motivates cache handoff: the source cache has already been materialized, so native serving pays target prefill plus the post-context forward, whereas reuse

Table 5: Measured handoff cost on all five evaluation tasks. Each cell reports complete translation time as a percentage of target prefill / end-to-end TTFT speedup with an existing source cache. Complete translation includes source de-RoPE, translator forward, and target re-RoPE. Measurements use bfloat16, batch size 8, five real batches per task, and ten timed repetitions after warmup.

Pair	BoolQ	LogiQA	PubMedQA	MultiRC	ReClor
Qwen3-0.6B → 1.7B	32.2% / 1.97×	33.6% / 1.48×	27.8% / 2.42×	27.5% / 2.32×	43.9% / 1.27×
Coder-7B → 7B-Instruct	7.5% / 5.26×	8.7% / 1.89×	5.3% / 6.05×	4.7% / 4.95×	13.5% / 1.50×
Math-7B → 7B-Instruct	7.6% / 5.26×	8.7% / 1.89×	5.3% / 6.04×	4.9% / 4.93×	13.6% / 1.50×
Math-7B → Coder-7B	7.6% / 5.23×	8.7% / 1.88×	5.3% / 6.01×	4.8% / 4.94×	13.8% / 1.49×
Llama-3.1-8B → ToolACE-8B	10.3% / 4.57×	11.3% / 1.80×	8.7% / 5.00×	8.7% / 4.13×	15.6% / 1.48×

pays the complete positional conversion and translation path (source de-RoPE, translator, and target re-RoPE) plus the same post-context forward. Across all 25 pair–task cells, complete translation costs 4.7–43.9% of target prefill (13.6% on average) and reduces end-to-end TTFT by 1.27–$6.05\times$ (mean $3.41\times$). The three Qwen2.5 transfers require only 4.7–13.8% of target prefill and reach up to $6.05\times$ speedup. The cross-scale Qwen3 pair is the least favorable because its translator width follows the cache geometry while its 1.7B target is comparatively small; nevertheless every task still speeds up, by 1.27–$2.42\times$. The intended regime is repeated reuse, routing, or migration of caches that already exist; these measurements do not treat a newly computed source cache as free work.

5.5 How much does the Value receptive field matter?

Proposition 2 says that conditioning on source history cannot increase the Bayes-optimal error, and Section 3.5 argues that Values are the branch where target-side context aggregation is missing from a pointwise map. We therefore sweep the receptive field and its direction: kernel sizes 2, 4, 8, 16 and 64, anti-causal, bidirectional, causal cumulative mean, global mean, and no mixing at all. Relative to the pointwise translator, the best causal mixer (k=2) improves accuracy by $1.24\,pp$ (64.46 versus 63.22), while the default causal k=4 improves by $0.94\,pp$. The remaining context-capable variants form a narrow band, indicating that access to a short source history matters more than a particular receptive-field length or direction. We retain causal k=4 as a conservative incremental default: unlike anti-causal, bidirectional, or global mixing, it does not require future context, and its result is within $0.30\,pp$ of the best setting.

The mixer adds 29.6M parameters to the 58.8M pointwise Qwen3 translator. Thus the result is a quality–cost tradeoff rather than a free gain: the mixer gives a measurable $1.24\,pp$ best-case improvement on the cross-scale pair, whereas the pointwise translator remains an attractive lightweight alternative when parameter budget dominates.

6 Limitations and Future Work

Our evaluation covers differently post-trained models within a pretrained family and one cross-scale Qwen3 pair, but not independently pretrained families or unequal-depth architectures. Future work should test more distant pairs and real-world model upgrades and agent handoffs; thus, our results are initial evidence rather than a general solution to cross-model cache reuse.

7 Conclusion

We formulate cross-model cache reuse as interface adaptation and introduce KVRelay, which aligns the computation induced in frozen target attention from a calibrated, source-conditioned start. Across five pairs and five seeds it recovers $90.3 \pm 1.0\%$ of native context gain, while translation costs 13.6% of target prefill and yields a mean $3.41\times$ TTFT speedup. These results suggest that model upgrades and agent handoffs can reuse existing context without target re-prefilling; cross-family transfer and realistic long-horizon agent workflows remain open.

AI USE STATEMENT

Generative AI tools were used as assistive tools for manuscript drafting and editing, presentation and table formatting, and limited feedback on methodological exposition and mathematical arguments. The authors independently developed the core method, conducted the experiments, and verified all mathematical statements and AI-assisted material. The authors take full responsibility for the final content, claims, and artifacts in this work.

ETHICS STATEMENT

This work studies the reuse of internal model states to improve inference efficiency. It uses publicly available language models and established benchmarks and does not involve human subjects or personally identifiable information. Cross-model cache transfer may inherit behavioral biases or unsafe tendencies from the participating models; deployment should therefore retain the target model's standard safety evaluation, access control, and monitoring procedures. The authors have reviewed the work in accordance with the ICLR Code of Ethics.

REPRODUCIBILITY STATEMENT

The method, objective, cache-injection protocol, model pairs, datasets, evaluation metrics, and principal hyperparameters are described in the main paper. Appendix A provides proofs, Appendix B reports per-task results, and Appendices C–E provide detailed ablations and robustness results. Data construction, optimization, checkpoint selection, positional handling, and baseline calibration are specified in Appendix F. These details are intended to support independent reproduction of the reported results.

REFERENCES

Femi Bello, Anubrata Das, Fanzhi Zeng, Fangcong Yin, and Liu Leqi. Linear representation transferability hypothesis: Leveraging small models to steer large models. *CoRR*, abs/2506.00653, 2025. doi: 10.48550/ARXIV.2506.00653. URL https://doi.org/10.48550/arXiv.2506.00653.

Alan Chen, Jack Merullo, Alessandro Stolfo, and Ellie Pavlick. Transferring linear features across language models with model stitching. In Danielle Belgrave, Cheng Zhang, Laura N. Montoya, Hsuan-Tien Lin, Razvan Pascanu, Piotr Koniusz, Marzyeh Ghassemi, Nancy Chen, Iván Vladimir Meza Ruíz, and Arturo Loaiza-Bonilla (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025*, 2025. URL http://papers.nips.cc/paper_files/paper/2025/hash/4569a868e7aa891248832ec08445d071-Abstract-Conference.html.

Siyi Chen, Xiaoyan Zhang, Meng Wu, Jonathan Tremblay, Valts Blukis, Stan Birchfield, René Vidal, Alvaro Velasquez, Sijia Liu, and Qing Qu. See what I see, know what I think: Dense latent communication across heterogeneous agents. *CoRR*, abs/2606.13594, 2026. doi: 10.48550/ARXIV.2606.13594. URL https://doi.org/10.48550/arXiv.2606.13594.

Lucio M. Dery, Zohar Yahav, Henry Prior, Qixuan Feng, Jiajun Shen, and Arthur Szlam. Latent space communication via K-V cache alignment. *CoRR*, abs/2601.06123, 2026. doi: 10.48550/ARXIV.2601.06123. URL https://doi.org/10.48550/arXiv.2601.06123.

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. *CoRR*, abs/2510.03215, 2025. doi: 10.48550/ARXIV.2510.03215. URL https://doi.org/10.48550/arXiv.2510.03215.

In Gim, Guojun Chen, Seung-Seob Lee, Nikhil Sarda, Anurag Khandelwal, and Lin Zhong. Prompt cache: Modular attention reuse for low-latency inference. In Phillip B. Gibbons, Gennady Pekhimenko, and Christopher De Sa (eds.), *Proceedings of the Seventh Annual Conference on Machine Learning and Systems, MLSys 2024, Santa Clara, CA, USA, May 13-16, 2024*. mlsys.org,

2024. URL https://proceedings.mlsys.org/paper_files/paper/2024/hash/a66caa1703fe34705a4368c3014c1966-Abstract-Conference.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. *CoRR*, abs/2608.03893, 2026. doi: 10.48550/ARXIV.2608.03893. URL https://doi.org/10.48550/arXiv.2608.03893.

Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, Chenyu Ran, Lingfeng Xiao, Chenglin Wu, and Jürgen Schmidhuber. Metagpt: Meta programming for a multi-agent collaborative framework. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=VtmBAGCN7o.

Junhao Hu, Wenrui Huang, Weidong Wang, Haoyi Wang, Tiancheng Hu, Qin Zhang, Hao Feng, Xusheng Chen, Yizhou Shan, and Tao Xie. EPIC: efficient position-independent caching for serving large language models. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, volume 267 of *Proceedings of Machine Learning Research*. PMLR / OpenReview.net, 2025. URL https://proceedings.mlr.press/v267/hu25j.html.

Youcheng Huang, Chen Huang, Duanyu Feng, Wenqiang Lei, and Jiancheng Lv. Cross-model transferability among large language models on the platonic representations of concepts. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pp. 3686–3704. Association for Computational Linguistics, 2025. doi: 10.18653/V1/2025.ACL-LONG.185. URL https://doi.org/10.18653/v1/2025.acl-long.185.

Minyoung Huh, Brian Cheung, Tongzhou Wang, and Phillip Isola. Position: The platonic representation hypothesis. In Ruslan Salakhutdinov, Zico Kolter, Katherine A. Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, volume 235 of *Proceedings of Machine Learning Research*, pp. 20617–20642. PMLR / OpenReview.net, 2024. URL https://proceedings.mlr.press/v235/huh24a.html.

Jin-Woo Lee, Minkyung Song, Junghyun Oh, Seunghoon Han, Soyoung Park, Gwangseon Jang, and Sungsu Lim. Mixture-of-translators: Translating KV caches across heterogeneous large language models. *CoRR*, abs/2607.28979, 2026. doi: 10.48550/ARXIV.2607.28979. URL https://doi.org/10.48550/arXiv.2607.28979.

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 Srikanth Kandula and Hakim Weatherspoon (eds.), *23rd USENIX Symposium on Networked Systems Design and Implementation, NSDI 2026, Renton, WA, May 4-6, 2026*, pp. 319–338. USENIX Association, 2026. URL https://www.usenix.org/conference/nsdi26/presentation/liu-yuhan.

Valentino Maiorca, Luca Moschella, Antonio Norelli, Marco Fumero, Francesco Locatello, and Emanuele Rodolà. Latent space translation via semantic alignment. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/ad5fa03c906ca15905144ca3fbf2a768-Abstract-Conference.html.

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 from preference data. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore,*

April 24-28, 2025. OpenReview.net, 2025. URL https://openreview.net/forum?id=8sSqNntaMr.

Maximillian Rossi, Prajwal Raghunath, and Eugene Wu. Latent cache flow: Model-to-model communication without text. *CoRR*, abs/2605.22863, 2026. doi: 10.48550/ARXIV.2605.22863. URL https://doi.org/10.48550/arXiv.2605.22863.

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. *CoRR*, abs/2603.13281, 2026. doi: 10.48550/ARXIV.2603.13281. URL https://doi.org/10.48550/arXiv.2603.13281.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Shaokun Zhang, Erkang Zhu, Beibin Li, Li Jiang, Xiaoyun Zhang, and Chi Wang. Autogen: Enabling next-gen LLM applications via multi-agent conversation framework. *CoRR*, abs/2308.08155, 2023. doi: 10.48550/ARXIV.2308.08155. URL https://doi.org/10.48550/arXiv.2308.08155.

Jingbo Yang, Bairu Hou, Wei Wei, Yujia Bao, and Shiyu Chang. Kvlink: Accelerating large language models via efficient KV cache reuse. In Danielle Belgrave, Cheng Zhang, Laura N. Montoya, Hsuan-Tien Lin, Razvan Pascanu, Piotr Koniusz, Marzyeh Ghassemi, Nancy Chen, Iván Vladimir Meza Ruíz, and Arturo Loaiza-Bonilla (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025*, 2025. URL http://papers.nips.cc/paper_files/paper/2025/hash/c217d123885341853cecbdd2be809983-Abstract-Conference.html.

Jiayi Yao, Hanchen Li, Yuhan Liu, Siddhant Ray, Yihua Cheng, Qizheng Zhang, Kuntai Du, Shan Lu, and Junchen Jiang. Cacheblend: Fast large language model serving for RAG with cached knowledge fusion. In *Proceedings of the Twentieth European Conference on Computer Systems, EuroSys 2025, Rotterdam, The Netherlands, 30 March 2025 - 3 April 2025*, pp. 94–109. ACM, 2025. doi: 10.1145/3689031.3696098. URL https://doi.org/10.1145/3689031.3696098.

A PROOFS

Proposition 1 (routing equivalence). Let $K'_j = K_j + c$ for all j and fix a query q_i. Then $q_i^\top K'_j = q_i^\top K_j + q_i^\top c$, where the second term does not depend on j. Since $\mathrm{softmax}(z + \gamma\mathbf{1}) = \mathrm{softmax}(z)$ for any scalar γ, the attention row is unchanged, hence so is $A(K')V = A(K)V$. Consequently the set $\{\hat{K} : \mathrm{softmax}(Q\hat{K}^\top) = \mathrm{softmax}(QK_t^\top)\}$ is non-trivial and $\|\hat{K} - K_t\|^2$ is not a faithful measure of functional error. □

Proposition 2 (conditioning cannot hurt). Under squared loss the Bayes-optimal predictor given a σ-algebra $\mathcal{G}$ is the conditional expectation, with risk $\mathbb{E}\big[\mathrm{Var}(V_{l,i}^t \mid \mathcal{G})\big]$. Taking $\mathcal{G}_1 = \sigma(V_{l,i}^s) \subseteq \mathcal{G}_2 = \sigma(V_{l,\le i}^s)$ and applying the law of total variance, $\mathbb{E}[\mathrm{Var}(X \mid \mathcal{G}_1)] = \mathbb{E}[\mathrm{Var}(X \mid \mathcal{G}_2)] + \mathbb{E}[\mathrm{Var}(\mathbb{E}[X \mid \mathcal{G}_2] \mid \mathcal{G}_1)] \ge \mathbb{E}[\mathrm{Var}(X \mid \mathcal{G}_2)]$, which gives $\mathcal{E}_{\mathrm{ctx}} \le \mathcal{E}_{\mathrm{point}}$. The second term vanishes exactly when $\mathbb{E}[V_{l,i}^t \mid V_{l,\le i}^s]$ is $\sigma(V_{l,i}^s)$-measurable. Thus strict improvement requires history to change the conditional mean; conditional dependence that changes only higher moments is insufficient. □

Remark on the calibrated base map. For a diagonal affine map $x \mapsto a \odot x + b$ acting per channel with positive source standard deviation, matching $\mathbb{E}[a \odot X_s + b] = \mu_t$ and $\mathrm{Var}[a \odot X_s + b] = \sigma_t^2$ has the unique positive solution $a = \sigma_t/\sigma_s$, $b = \mu_t - a \odot \mu_s$. This is exactly $\mathcal{B}_l^u$ on channels where the numerical floor is inactive. If $\sigma_s < \varepsilon$, the implemented scale $a = \sigma_t/\varepsilon$ preserves exact mean matching but intentionally relaxes exact variance matching for stability. This is a statement about marginal moments only; it does not imply that the base map recovers the target representation, and our ablations confirm that its particular form is not what drives accuracy.

B PER-TASK RESULTS

Tables 6–10 give the per-task decomposition of Table 2.

Table 6: Qwen3-0.6B → Qwen3-1.7B (per task, %).

Setting	BoolQ	LogiQA	PubMedQA	MultiRC	ReClor	Mean
target	78.78	41.63	69.00	88.12	56.60	66.82
no-ctx	62.26	33.03	55.30	70.96	38.80	52.07
source	65.87	34.10	56.90	67.94	43.00	53.56
naive	52.94	23.66	49.50	50.55	23.20	39.97
KVRelay	**74.50**	**40.71**	**65.60**	**87.19**	**52.80**	**64.16**

Table 7: Qwen2.5-Coder-7B → Qwen2.5-7B-Instruct (per task, %).

Setting	BoolQ	LogiQA	PubMedQA	MultiRC	ReClor	Mean
target	86.15	55.76	76.20	94.96	73.40	77.29
no-ctx	67.43	39.02	54.70	75.94	48.20	57.06
source	80.18	46.70	75.20	93.74	61.20	71.40
naive	42.02	19.51	34.10	50.49	24.00	34.02
KVRelay	**85.50**	**51.92**	**75.60**	**95.01**	**71.80**	**75.97**

Table 8: Qwen2.5-Math-7B → Qwen2.5-7B-Instruct (per task, %).

Setting	BoolQ	LogiQA	PubMedQA	MultiRC	ReClor	Mean
target	86.15	55.76	76.20	94.96	73.40	77.29
no-ctx	67.43	39.02	54.70	75.94	48.20	57.06
source	76.21	42.86	73.60	90.20	53.80	67.33
naive	38.99	21.20	33.90	50.96	24.20	33.85
KVRelay	**84.50**	**51.00**	**74.50**	**93.80**	**69.20**	**74.60**

C PER-TASK SUPERVISION ABLATION

Table 11 expands Table 3 to the task level. Canonical-space KV MSE is numerically worse in all 25 cells. The legacy multi-term objective is less consistent: it is substantially worse on the cross-scale pair, but approximately tied with the main loss on two Qwen2.5 transfers.

D CANONICAL-ROPE SEED STABILITY

We train the canonical-RoPE main configuration with seeds 0–4 for every pair. For each seed, we first average accuracy over the five tasks, then compute recovery from that task-macro accuracy and the corresponding task-macro target and no-context references, exactly matching Eq. (8) and Table 2. Table 12 reports the mean and sample standard deviation over the resulting five seed-level values. The main tables retain seed 0 so that all ablations remain paired, while these runs quantify main-method variability.

The result is stable across training randomness: per-pair accuracy standard deviations range from 0.10 to 0.66 points, and the pair-macro average varies by only 0.17 points across seeds. The corresponding mean recovery remains 0.903 ± 0.010, close to the seed-0 value of 0.907 reported in Table 2.

Table 9: Qwen2.5-Math-7B → Qwen2.5-Coder-7B (per task, %).

Setting	BoolQ	LogiQA	PubMedQA	MultiRC	ReClor	Mean
target	80.18	46.70	75.20	93.74	61.20	71.40
no-ctx	64.62	39.94	57.90	78.61	42.40	56.69
source	76.21	42.86	73.60	90.20	53.80	67.33
naive	61.93	27.96	50.20	63.01	33.60	47.34
KVRelay	**78.10**	**47.47**	**72.20**	**92.58**	**58.40**	**69.75**

Table 10: Llama-3.1-8B → ToolACE-8B (per task, %). The target is a fine-tune of the source, and naive reuse is already close to the ceiling.

Setting	BoolQ	LogiQA	PubMedQA	MultiRC	ReClor	Mean
target	86.06	43.63	75.50	95.42	54.80	71.08
no-ctx	68.99	35.02	58.10	77.28	37.00	55.28
source	82.54	39.78	75.50	92.70	53.20	68.74
naive	85.50	42.86	73.70	95.13	55.60	70.56
KVRelay	**85.96**	**43.93**	**75.70**	**95.30**	**56.40**	**71.46**

E Repeated-context length stress test

The main experiments train and evaluate on contexts capped at 512 tokens. To separate positional and cache-length extrapolation from semantic long-context reasoning, we additionally construct repeated-context tasks from BoolQ, MultiRC and PubMedQA. Each original passage is repeated until the resulting context approaches caps of 512, 1024, 2048 or 4096 tokens; each pair–task–length cell contains 300 examples. This construction deliberately preserves the underlying question and evidence while moving the cache to positions not observed during translator training. It is therefore a stress test of canonical positional handling and error accumulation, not a substitute for natural long-document evaluation.

Four of five pairs remain within $2.11\,pp$ of the native-cache reference at every tested length, including both transfers from Qwen2.5-Math whose source and target use different RoPE bases. This supports the intended role of de-RoPE/re-RoPE: the learned Key map need not absorb a fixed source positional frame, and can be re-positioned using the target model's native RoPE. The result is not universal, however. On Qwen3-0.6B→1.7B the gap grows from $3.78\,pp$ at 512 tokens to $5.89\,pp$ at 2048 and $21.33\,pp$ at 4096. The 4096-token failure occurs on all three tasks (BoolQ $20.33\,pp$, MultiRC $29.33\,pp$, PubMedQA $14.33\,pp$), so it is not driven by one benchmark. Canonical positional alignment removes a coordinate mismatch; it does not guarantee that a translator learned on short contexts extrapolates across every model-scale gap.

F Additional implementation details

Data construction and separation. The corpus contains fixed quotas from BoolQ (4500), PubMedQA (4000), MultiRC (4000), RACE-high (3500), QASC (1500), ReClor (1500) and OpenBookQA (1000). Every record stores a standalone reusable context separately from the question, options and answer cue. Before sampling, we normalize whitespace and case in the tuple (context, query, choices), remove within-source duplicates, and remove exact matches against the corresponding evaluation split; we then deduplicate globally across sources. BoolQ, MultiRC, PubMedQA and ReClor are therefore same-dataset but disjoint-split evaluation, whereas LogiQA is absent from training. The text corpus is model-independent and is tokenized only when a target model is selected.

Training configuration. The shuffled 20,000-example corpus is divided deterministically into its first 512 examples for validation and the remaining 19,488 for training. Contexts are truncated to 512 tokens; their median and 95th-percentile lengths are 221 and 465 tokens, respectively. We train

Table 11: Canonical-RoPE supervision ablation per task (%).

Pair	Setting	BoolQ	LogiQA	PubMedQA	MultiRC	ReClor	Mean
Qwen3-0.6B → 1.7B	KVRelay	74.50	40.71	65.60	87.19	52.80	**64.16**
	feat	66.21	34.25	54.00	79.88	48.00	56.47
	legacy	69.51	35.33	63.10	74.26	39.80	56.40
Coder-7B → 7B-Instruct	KVRelay	85.50	51.92	75.60	95.01	71.80	**75.97**
	feat	77.31	42.09	66.20	92.00	63.20	68.16
	legacy	84.95	51.31	76.00	92.87	67.00	74.43
Math-7B → 7B-Instruct	KVRelay	84.50	51.00	74.50	93.80	69.20	**74.60**
	feat	76.18	49.77	65.90	92.41	66.80	70.21
	legacy	84.59	51.15	74.40	93.45	70.40	74.80
Math-7B → Coder-7B	KVRelay	78.10	47.47	72.20	92.58	58.40	**69.75**
	feat	71.96	44.39	67.40	90.26	55.60	65.92
	legacy	79.20	45.93	72.40	91.88	58.60	69.60
Llama-3.1-8B → ToolACE-8B	KVRelay	85.96	43.93	75.70	95.30	56.40	**71.46**
	feat	85.69	43.32	74.00	95.25	55.60	70.77
	legacy	85.11	40.71	75.90	94.20	52.60	69.70

Table 12: Five-seed stability of the canonical-RoPE main method. Accuracy is the five-task mean (%); recovery is the fraction of the target's native context gain. The final row first averages over model pairs within each seed and then reports mean ± sample standard deviation.

Pair	Accuracy (%)	Recovery
Qwen3-0.6B → 1.7B	64.04 ± 0.66	0.811 ± 0.045
Coder-7B → 7B-Instruct	75.91 ± 0.48	0.932 ± 0.024
Math-7B → 7B-Instruct	74.76 ± 0.23	0.875 ± 0.011
Math-7B → Coder-7B	69.86 ± 0.10	0.895 ± 0.007
Llama-3.1-8B → ToolACE-8B	71.13 ± 0.26	1.003 ± 0.016
Pair macro-average	71.14 ± 0.17	0.903 ± 0.010

each translator for three epochs using a learning rate of 3×10^{-4} with 27 warmup steps. Each of eight GPUs processes a local batch of two examples, and gradients are accumulated for four steps, giving an effective batch size of 64 and 912 optimization steps in total. Training uses bfloat16 precision. Unless otherwise noted, every controlled ablation changes only the factor under study while retaining the same data split, optimization schedule and random seed. The attention loss is evaluated at every valid position in the post-context sequence (the query followed by the gold answer); answer-logit KL and direct KV losses have zero weight in the main configuration.

Statistics collection. When no precomputed statistics file is supplied, the first 32 mini-batches of training are used to estimate $\mu^u_{s,l}, \sigma^u_{s,l}, \mu^u_{t,l}, \sigma^u_{t,l}$ over samples, KV heads and valid context tokens. They are written into non-trainable buffers and saved with the checkpoint, so evaluation always reuses the statistics the model was trained with.

Numerical precision. All training, accuracy evaluation, and timing measurements use bfloat16 precision.

Choice scoring and model selection. For each candidate, we append its token sequence to the shared query and compute the teacher-forced, token-average negative log likelihood over all candidate tokens; this handles multi-token labels without favoring shorter candidates. The prediction is the candidate with minimum average NLL. We evaluate the translator every 100 optimization steps on the fixed 512-example validation set and select the checkpoint with the highest multiple-choice accuracy. Training still runs for the complete epoch budget, and the final checkpoint is stored separately but is not used for the reported results.

Positions and cache injection. Contexts are left padded, with position IDs restarting at zero at the first valid token. The right-padded post-context sequence continues from each context's true length.

Table 13: Repeated-context stress test. Each entry is target accuracy / translated-cache accuracy (%), macro-averaged over BoolQ, MultiRC and PubMedQA. "Gap" is target minus translation. Translators are trained only with contexts capped at 512 tokens.

Pair	512	1024	2048	4096
Qwen3-0.6B → 1.7B	78.56/74.78 (3.78)	78.00/74.22 (3.78)	78.67/72.78 (5.89)	78.33/57.00 (21.33)
Coder-7B → 7B-Instruct	86.89/86.11 (0.78)	85.67/85.11 (0.56)	84.56/84.33 (0.22)	84.22/84.22 (0.00)
Math-7B → 7B-Instruct	86.89/84.89 (2.00)	85.67/84.56 (1.11)	84.56/83.56 (1.00)	84.22/82.67 (1.56)
Math-7B → Coder-7B	82.11/80.00 (2.11)	80.22/78.56 (1.67)	78.44/77.89 (0.56)	78.56/77.67 (0.89)
Llama-3.1-8B → ToolACE-8B	85.44/85.56 (-0.11)	85.11/85.22 (-0.11)	84.89/85.56 (-0.67)	84.89/84.89 (0.00)

We invert the source model's native rotary embedding on source Keys, compute Key statistics and translation in that canonical frame, and apply the target model's native rotary embedding immediately before installing the translated tensors as the target's dynamic cache. Values are not rotated. The implementation calls each model's own rotary module, including its configured scaling, and explicitly divides by the rotary norm in the inverse. Post-context cache slots are assigned separately from these RoPE position IDs. Three pairs have identical source/target RoPE bases; the two pairs sourced from Qwen2.5-Math use a different base and are therefore direct tests of positional-frame conversion rather than of a learned map implicitly absorbing the mismatch.

Baseline details. The Ridge baseline solves an independent regularized least-squares problem per target layer and head, using the top-k ($k = 8$) most correlated source layers as features. It is calibrated on 7600 sequences (1.946M tokens); an earlier 0.128M-token calibration gives the same five-pair average, so the gap to KVRelay is not explained by calibration size.