Dual-Cache Latent Space Communication between Heterogeneous Language Models

Jiyao Liu[1*], **Qi Zhang**[2*†], **Yaoyi Jia**[1], **Ziwen Kan**[3], **Song Wang**[3]
[1]Independent Researcher
[2]Amazon Web Services (AWS)
[3]University of Central Florida (UCF)
jiyao.liu@outlook.com, qizmax@amazon.com, yaoyi.jia@colorado.edu, song.wang@ucf.edu

Abstract

Multi-agent LLM systems split work across models, so answering often requires knowledge that sits in another agent's context: a Sharer has encoded information that a Receiver needs to complete its task. Such agents usually communicate by exchanging text, which places autoregressive decoding on the critical path and reduces the exchange to a discrete message written without sight of the receiver's state. Recent latent protocols instead translate the sharer's key–value (KV) cache into the receiver's: C2C supports heterogeneous models but requires both to read the same input, while LCF-X removes this shared-context requirement through position-free sharer-cache pooling. Three restrictions remain: LCF-X compresses the sharer alone, supplies the same layer-local summary to every receiver position with no joint cross-layer memory to retrieve from, and assumes matched layer count and KV geometry. We introduce XKV, which lifts all three with three components: learned-query attention pools *both* caches; self-attention over receiver-aligned layer tokens, with a learned layer map reconciling different depths, mixes the pooled summaries into a compact joint memory; and a shared position decoder lets every raw receiver cache position retrieve its own per-head-gated residual in the receiver's native KV geometry. Both models stay frozen and may differ in family, depth, KV-head count, head dimension, and tokenizer; only the translator is trained. Across 45 dataset–model-pair settings (six heterogeneous and three same-model ordered pairings, five datasets), XKV attains the highest macro score and best average rank of the three protocols, improving on LCF-X on every dataset (by 4.6 exact-match and 4.2 F1 points on ROPES) and surpassing text communication on four of the five, while training 76% fewer parameters and translating a cache pair $10.3\times$ faster (5.8 against 59.9 ms); end to end, XKV is 26% faster than LCF-X and $6.8\times$ faster than text communication.

Introduction

Language models increasingly act as components of larger systems, retrieving evidence, invoking tools, and verifying one another (Yao et al. 2023; Du et al. 2024; Wu et al. 2024; Chen et al. 2024). In such systems, agents routinely observe different things: retrieved documents are partitioned across agents, tools return results only to their caller, and each interaction leaves private state behind. Whenever one agent needs evidence another has already encoded, the two must communicate, and collective performance depends on that channel. We study its two-model core, the *cross-context* setting: a Sharer has encoded part of the evidence for a question, a Receiver has encoded the remainder and must produce the answer; neither context suffices alone, so the receiver must integrate the evidence it lacks with what it already holds. Deployed systems address this almost exclusively by exchanging natural language. Text is portable and interpretable, but between neural models it is indirect: the sender autoregressively decodes its continuous state into tokens on the critical path, and the receiver must tokenize and re-encode them. Cache-to-Cache (C2C) eliminates this decode–encode cycle by translating one model's key–value (KV) cache into another's (Fu et al. 2026; Dery et al. 2026). It supports heterogeneous families and tokenizers, but aligns the two tokenizations of the same underlying input and fuses semantically corresponding positions, so it remains a shared-context method; cache-reuse methods that realign positions make a related correspondence assumption (Ye et al. 2025). The cross-context extension of Latent Cache Flow, LCF-X, removes this presupposition with a pooled, position-free cache summary (Rossi, Raghunath, and Wu 2026), showing that evidence can move at the cache level even when the contexts share no tokens.

Position-independent transfer, however, leaves three restrictions in place, contrasted in Figure 1. **First, what to communicate is decided without reference to the receiver**: only the sharer cache is pooled (LCF-X's projector reads the receiver cache only after the summary is formed), so one vector must carry every fact the receiver might need, selected without consulting what it has already encoded. **Second, the external signal is a layer-local, single-summary bottleneck**: every position of a receiver layer gets the same pooled sharer summary; the projector's view of the local cache row lets residuals vary across positions, but no position can select among multiple sharer summaries or retrieve information from other layers. **Third, flexibility is limited**: C2C accepts heterogeneous models but requires shared, token-alignable content, and original LCF-X accepts different content but assumes matched layer count and KV-head geometry; a prac-

*These authors contributed equally.
†Work does not relate to the author's position at Amazon.

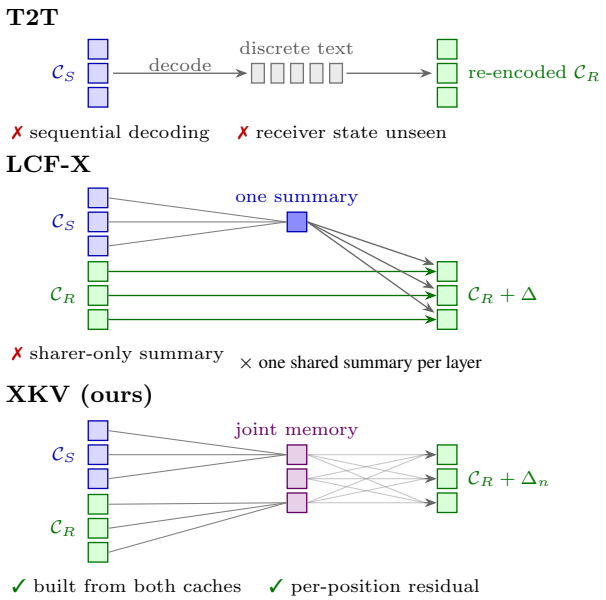

Figure 1: Communication protocols for two models holding complementary contexts, annotated with the challenges each raises (×) or resolves (✓). T2T decodes a message on the critical path without access to the receiver's state. LCF-X pools a summary from the sharer cache alone and supplies the same summary to every position, although its downstream projector also conditions on the local receiver cache row. XKV (ours) builds the communication memory jointly from both caches, and each receiver position retrieves its own receiver-native residual.

tical channel should accept a frozen pair that differs along these axes. All three must be addressed under an efficiency constraint: a receiver-aware channel that reinstates the cost latent communication was introduced to remove yields no net benefit.

We resolve these with XKV, a dual-cache translator that separates *what to communicate* from *where to write it*. Learned-query attention pools *both* caches into k candidate summaries per layer and head. Since depth carries no reliable semantic correspondence across architectures (Kornblith et al. 2019; Bansal, Nakkiran, and Barak 2021), a learned layer map reconciles the two depth axes, and self-attention mixes the summaries into a compact joint memory that represents the sharer's evidence relative to what the receiver has encoded, not as a fixed sender-authored message. Each raw receiver cache position then queries this memory through a shared decoder, and zero-initialized output heads emit per-head-gated residuals in the receiver's native cache geometry, giving every position a distinct update. Efficiency follows from the same choices rather than trading against them: pooling reduces $N \to k$ in one vectorized step rather than two hard-to-batch stages over variable-length spans; the memory has length L_R regardless of prompt length; and the decoder, gate, and output heads are shared across receiver layers. Both models remain frozen and may differ in depth, KV-head count, head dimension, and tokenizer; only the translator is trained.

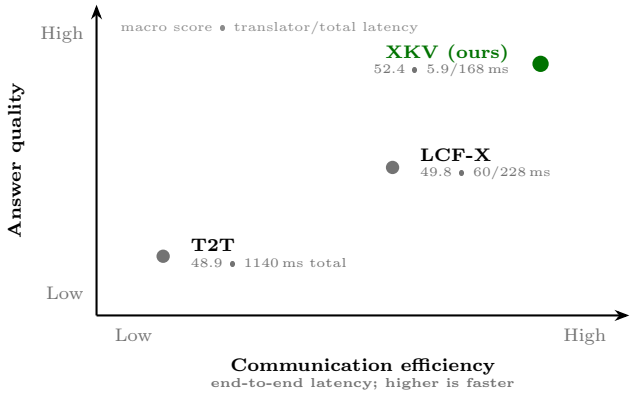

Figure 2: Answer quality versus communication cost, aggregated over all 45 dataset–model-pair settings (details in the technical appendix); the ordering on both axes is faithful, the spacing is not to scale. XKV dominates both baselines: it improves over LCF-X on every dataset with a $10.3\times$ faster translator and 76% fewer trainable parameters, and is $6.8\times$ faster end to end than T2T, which leads only on QASC. C2C is omitted: it requires token-aligned shared context.

We evaluate XKV on five split-evidence reasoning tasks (Yang et al. 2018; Trivedi et al. 2022; Khot et al. 2020; Lin et al. 2019; Geva et al. 2021) over the complete 3×3 grid of Qwen, Gemma, and Llama sharer–receiver pairs (Yang et al. 2025; Gemma Team 2025; Grattafiori et al. 2024), whose six off-diagonal pairs collectively span family, depth, attention-geometry, and tokenizer differences; since original LCF-X does not support such mismatches, we extend it into a stronger heterogeneous baseline. Across all 45 dataset–model-pair settings, XKV attains the highest macro score (52.4 against 49.8 for LCF-X and 48.9 for T2T), improving on LCF-X on every dataset, while its translator runs $10.3\times$ faster than the LCF-X fusor with 76% fewer trainable parameters; end to end it is 26% faster than LCF-X and $6.8\times$ faster than T2T (Figure 2). The contribution is therefore the removal of a trade-off: communication that is receiver-aware and retrieves position-specific updates from a joint cross-layer memory, yet is strictly cheaper than the sender-centric summary it replaces.

We make three contributions:

- We identify receiver-independent compression, layer-local single-summary conditioning, and matched-geometry assumptions as bottlenecks of cross-context cache communication, and reformulate the task as efficient joint translation of both models' states.
- We introduce XKV, combining symmetric direct cache pooling, cross-layer translation, and receiver-position queries into a joint memory for receiver-native reconstruction, with both frozen models free to differ in architecture and tokenizer.
- We extend LCF-X to mismatched layer counts and KV dimensions and conduct, to our knowledge, the first full cross-family, split-evidence evaluation (five tasks, nine ordered pairs), where XKV improves on both the strengthened LCF-X baseline and text communication at a fraction of their cost.

Related Work

Communication between language-model agents. Deployed multi-agent systems coordinate exclusively through natural language: agents exchange textual arguments in debate (Du et al. 2024), converse in AutoGen (Wu et al. 2024), and collaborate in role-specialized teams (Chen et al. 2024), paying autoregressive generation and re-encoding at every handoff; our T2T baseline follows this protocol. A second line avoids committing intermediate computation to language: Coconut feeds hidden states back as continuous reasoning (Hao et al. 2025), Interlat learns compressed hidden-state communication between agents (Du et al. 2026), and LatentMAS maintains a shared latent working memory (Zou et al. 2025). These methods operate on hidden-state sequences or a shared latent interface; XKV instead targets frozen, independently tokenized models and writes into the per-layer KV state that the receiver's attention already consumes.

KV caches as a communication medium. The KV cache is the persistent prompt state consumed during decoding (Pope et al. 2023). Within one model, eviction policies (Zhang et al. 2023; Li et al. 2024; Xiao et al. 2024) and latent projections (DeepSeek-AI 2024) show that state is highly redundant, which is what makes a compact cross-model channel plausible. Between models, C2C learns adapters that support heterogeneous models while fusing position-aligned caches of the same underlying input (Fu et al. 2026; Dery et al. 2026), and KV entries can be selectively shared (Shi et al. 2026) or reused across agents after positional realignment (Ye et al. 2025); all assume overlapping content, but our two models hold complementary evidence. Most closely, LCF-X pools the sharer cache into a position-free summary before layer-local residual fusion (Rossi, Raghunath, and Wu 2026). XKV advances this line by building a cross-layer memory from both caches and letting each receiver position retrieve a distinct update through shared parameters.

Bridging frozen representation spaces. Translating between frozen models presumes their states are relatable: similarity analyses find broadly comparable features that resist naive layer-to-layer identification (Kornblith et al. 2019), stitching shows a light learned map often suffices (Bansal, Nakkiran, and Barak 2021), and relative representations communicate zero-shot through shared anchors (Moschella et al. 2023); together these motivate our learned layer map and learned translation. Prefix tuning (Li and Liang 2021) and gist tokens (Mu, Li, and Goodman 2023) likewise inject continuous vectors into a frozen model, but as a static prefix or single-model compression rather than an example-dependent translation of two models' states. Finally, pooling with learned queries is a standard primitive (Lee et al. 2019; Jaegle et al. 2021); XKV applies it per layer and KV head to both caches at once, so the resulting slots stay cache-shaped and can be reassembled into receiver-native keys and values.

Method

XKV is a lightweight translator between two frozen language models, built on a single principle: *what to communicate* should be decided jointly from both models' encoded states,

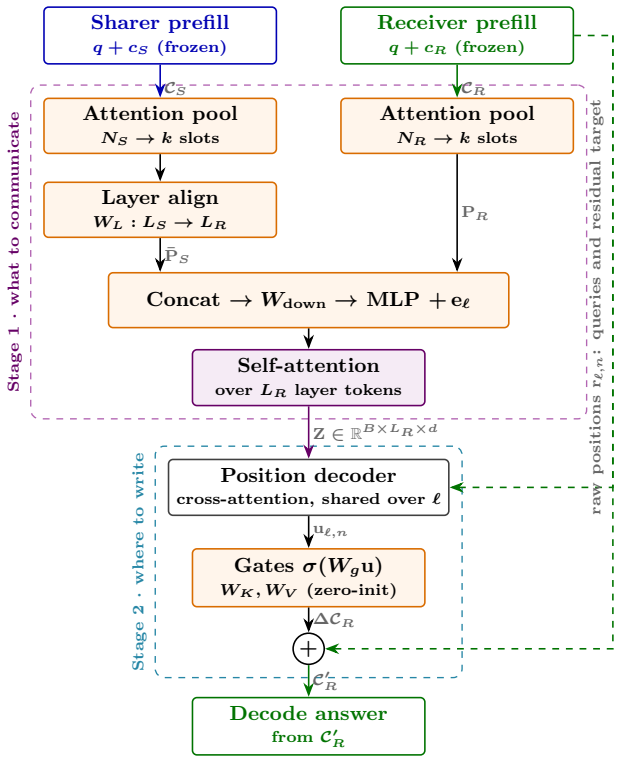

Figure 3: Overview of XKV. Both frozen models prefill their private contexts. Stage 1 (*what to communicate*): learned-query attention pools each cache from N_a positions to k slots per layer and head, a learned map aligns the sharer layer axis, and the concatenated summaries are bottlenecked and mixed by self-attention into one joint memory token per receiver layer. Stage 2 (*where to write*): every raw receiver cache position queries the memory through a shared cross-attention decoder, and shared, zero-initialized output heads emit per-head-gated KV residuals in the receiver's native cache geometry. Only the XKV translator is trained.

whereas *where to write it* should be dictated by the receiver's own cache geometry. Accordingly, XKV first compresses both KV caches into a compact joint layer memory through symmetric pooling, layer alignment, and cross-layer mixing, then lets every raw receiver cache position retrieve its own update from that memory, written back as receiver-native KV residuals (Figure 3).

Problem Setup

Let a frozen Sharer model S observe a question q and private context c_S, and let a frozen Receiver model R observe q and complementary context c_R. The receiver must predict answer tokens $y = (y_1, \ldots, y_T)$. Prefilling model $a \in \{S, R\}$ produces a KV cache

$$\mathcal{C}_a = \left\{(\mathbf{K}_a^\ell, \mathbf{V}_a^\ell)\right\}_{\ell=1}^{L_a}, \quad \mathbf{K}_a^\ell, \mathbf{V}_a^\ell \in \mathbb{R}^{B \times H_a \times N_a \times D_a}, \quad (1)$$

where L_a, H_a, N_a, and D_a are the number of layers, KV heads, cached positions, and the head dimension. These quantities and the tokenizer may differ between S and R; architectures that interleave full and sliding-window atten-

tion are handled by operating on the common valid suffix of the per-layer caches. XKV is a learned translator $F_\theta(\mathcal{C}_S, \mathcal{C}_R) = \Delta\mathcal{C}_R$ that produces receiver-shaped residuals and updates the prompt cache once,

$$\mathcal{C}'_R = \mathcal{C}_R + \Delta\mathcal{C}_R, \tag{2}$$

after which the receiver generates from $\mathcal{C}'_R$ with its standard causal decoder. Both language models stay frozen; only θ is trained.

Symmetric Direct Cache Pooling

Motivation. A single sharer-only summary forces several useful source facts to compete for one bottleneck and cannot express their relevance to what the receiver has already encoded; we therefore keep k candidate summaries per layer and head, and pool *both* caches with the same mechanism.

For side $a \in \{S, R\}$, layer ℓ, KV head h, and slot $j \in \{1, \dots, k\}$, XKV learns a query $\mathbf{q}_{a,\ell,h,j} \in \mathbb{R}^{D_a}$. Keys determine attention weights over the raw cache positions:

$$A_{a,\ell,h,j,n} = \mathrm{softmax}_n\left(\frac{\mathbf{q}^\top_{a,\ell,h,j}\mathbf{K}_{a,\ell,h,n}}{\sqrt{D_a}}\right), \tag{3}$$

and the same weights pool keys and values,

$$\mathbf{p}^X_{a,\ell,h,j} = \sum_{n=1}^{N_a} A_{a,\ell,h,j,n}\mathbf{X}_{a,\ell,h,n}, \quad \mathbf{X} \in \{\mathbf{K}, \mathbf{V}\}, \tag{4}$$

yielding $\mathbf{P}^K_a, \mathbf{P}^V_a \in \mathbb{R}^{B\times L_a\times H_a\times k\times D_a}$. Padding positions are masked out of the softmax. Unlike LCF-X's hierarchical within-span and then across-span pooling (Rossi, Raghunath, and Wu 2026), this operation reduces $N_a \to k$ in one step and is fully vectorized across layers, heads, and slots. In the default configuration, $k \in \{1, 2, 3\}$ grows with receiver model size.

Layer Alignment and Joint Memory

Motivation. Depth provides no reliable one-to-one semantic correspondence across architectures (Kornblith et al. 2019), so the memory should be receiver-aligned yet allow both depth conversion and communication among layers.

When $L_S \neq L_R$, an affine map along the layer axis, $\bar{\mathbf{P}}_S = W_L\mathbf{P}_S + \mathbf{b}_L$ with $W_L \in \mathbb{R}^{L_R\times L_S}$ applied independently to every pooled feature, converts the sharer summaries to L_R layers; the identity is used when depths match. For receiver layer ℓ, all k pooled slots of both sides are concatenated into one feature vector,

$$\mathbf{x}_\ell = \left[\mathrm{vec}(\bar{\mathbf{P}}^K_{S,\ell}); \mathrm{vec}(\bar{\mathbf{P}}^V_{S,\ell}); \mathrm{vec}(\mathbf{P}^K_{R,\ell}); \mathrm{vec}(\mathbf{P}^V_{R,\ell})\right]. \tag{5}$$

A shared down-projection and bottleneck MLP produce one memory token per receiver layer,

$$\mathbf{z}^0_\ell = \mathrm{LN}(\mathrm{MLP}(W_{\mathrm{down}}\mathbf{x}_\ell)) + \mathbf{e}_\ell, \quad \mathbf{Z}^0 \in \mathbb{R}^{B\times L_R\times d}, \tag{6}$$

where $\mathbf{e}_\ell$ is a learned receiver-layer embedding and the latent width d defaults to one eighth of the receiver hidden width. A pre-normalized Transformer block then mixes information across the receiver-aligned layer tokens:

$$\tilde{\mathbf{Z}} = \mathbf{Z}^0 + \mathrm{MHA}(\mathrm{LN}(\mathbf{Z}^0)), \tag{7}$$

$$\mathbf{Z} = \tilde{\mathbf{Z}} + \mathrm{FFN}(\mathrm{LN}(\tilde{\mathbf{Z}})). \tag{8}$$

The resulting $\mathbf{Z}$ is a joint, cross-layer communication memory. Its sequence length is L_R, independent of either prompt length, and it represents the sharer's evidence *in relation to* what the receiver has already encoded rather than as a fixed sender-authored message.

Receiver-Position Retrieval and Update

Motivation. Global pooling removes token locality. In LCF-X, every position receives the same external sharer summary, although a downstream projector also reads the local receiver cache and may therefore emit position-varying residuals. XKV instead uses the original receiver cache entries as explicit queries into a multi-slot, cross-layer memory, so each position can retrieve a different combination of sharer and receiver information.

For layer ℓ and position n, the raw receiver key and value are flattened across heads and concatenated,

$$\mathbf{r}_{\ell,n} = \left[\mathrm{vec}(\mathbf{K}^\ell_{R,:,n,:}); \mathrm{vec}(\mathbf{V}^\ell_{R,:,n,:})\right], \tag{9}$$

and a decoder shared across all receiver layers forms queries $\mathbf{q}_{\ell,n} = W_q\mathbf{r}_{\ell,n} + \mathbf{e}_\ell$, which cross-attend to all L_R memory tokens:

$$\tilde{\mathbf{u}}_\ell = \mathbf{q}_\ell + \mathrm{MHA}(\mathrm{LN}(\mathbf{q}_\ell), \mathbf{Z}, \mathbf{Z}), \tag{10}$$

$$\mathbf{u}_\ell = \tilde{\mathbf{u}}_\ell + \mathrm{FFN}(\mathrm{LN}(\tilde{\mathbf{u}}_\ell)). \tag{11}$$

The layer embedding $\mathbf{e}_\ell$ preserves layer identity under weight sharing. Shared output heads then predict receiver-shaped updates, with a deterministic sigmoid gate providing separate scales for every position, KV head, and cache type:

$$[\boldsymbol{\alpha}^K_{\ell,n}; \boldsymbol{\alpha}^V_{\ell,n}] = \sigma(W_g\mathbf{u}_{\ell,n}), \tag{12}$$

$$\Delta\mathbf{K}^\ell_{R,:,n,:} = \boldsymbol{\alpha}^K_{\ell,n} \odot \mathrm{reshape}(W_K\mathbf{u}_{\ell,n}), \tag{13}$$

$$\Delta\mathbf{V}^\ell_{R,:,n,:} = \boldsymbol{\alpha}^V_{\ell,n} \odot \mathrm{reshape}(W_V\mathbf{u}_{\ell,n}). \tag{14}$$

Residuals at padding positions are zeroed, and the remaining residuals are added to the receiver cache as in Eq. (2).

Identity at initialization. W_K and W_V are shared across receiver layers and initialized to zero, so an untrained translator reproduces receiver-only decoding exactly; training therefore learns a residual *correction* of the receiver cache rather than a replacement, which stabilizes early optimization.

Training and Inference

Both base models are frozen; only θ is trained with teacher-forced cross-entropy on the answer tokens:

$$\mathcal{L}(\theta) = -\frac{1}{T}\sum_{t=1}^{T}\log p_R(y_t \mid y_{<t}, q, c_R, F_\theta(\mathcal{C}_S, \mathcal{C}_R)). \tag{15}$$

Although the receiver weights are fixed, gradients flow through its attention computation into the XKV residuals. At inference, the receiver prefills all but the final prompt token, XKV modifies that prefix cache once, and the held-out token starts ordinary greedy decoding. Cache entries created for generated tokens are appended normally and never retroactively rewritten, so the one-shot update adds no per-token decoding cost.

Table 1: Full results for all nine ordered sharer–receiver settings. Each setting occupies three rows, one per communication method. We report EM/F1 for generative QA and accuracy (Acc.) for QASC and StrategyQA; higher is better. Cross-dataset Avg. averages the primary metric (F1 for generative QA and accuracy for classification) across all five datasets. The technical appendix provides the same results as per-dataset tables with cell-wise highlighting.

Sharer → Receiver	Method	ROPES		MuSiQue		QASC	StrategyQA	HotpotQA-bridge		Cross-dataset Avg.
		EM	F1	EM	F1	Acc.	Acc.	EM	F1	
Qwen → Qwen	T2T	35.8	45.2	4.6	12.2	78.6	**64.1**	16.6	27.1	45.44
	LCF-X	45.2	52.4	7.7	15.4	**82.4**	58.8	29.1	40.6	49.92
	XKV	**47.9**	**55.6**	**8.1**	**16.0**	82.2	61.4	**29.5**	**40.7**	**51.18**
Qwen → Gemma	T2T	42.4	48.6	**6.8**	**14.5**	**65.8**	51.9	22.4	34.0	**42.96**
	LCF-X	43.9	51.3	6.0	13.3	49.2	**60.8**	23.3	34.0	41.72
	XKV	**46.2**	**54.3**	5.4	12.5	55.1	58.2	**23.8**	**34.7**	**42.96**
Qwen → Llama	T2T	50.6	58.0	9.3	19.5	86.7	73.4	30.4	44.0	56.32
	LCF-X	51.4	58.5	16.0	26.3	92.0	72.9	41.1	54.5	60.84
	XKV	**57.3**	**61.8**	**19.2**	**29.9**	**93.3**	**73.6**	**41.5**	**55.0**	**62.72**
Gemma → Qwen	T2T	31.9	42.5	3.4	10.8	**84.6**	**65.9**	15.5	25.7	45.90
	LCF-X	47.2	53.7	**8.4**	**16.6**	81.3	59.3	29.4	40.9	50.36
	XKV	**49.1**	**55.4**	8.1	15.8	82.4	62.0	**29.5**	**41.0**	**51.32**
Gemma → Gemma	T2T	31.8	42.1	**6.2**	**13.4**	**55.7**	57.6	21.7	33.1	40.38
	LCF-X	45.5	51.4	4.6	11.9	44.8	58.2	23.5	**34.4**	40.14
	XKV	**48.1**	**55.6**	5.8	13.1	53.9	**58.8**	**23.9**	**34.4**	**43.16**
Gemma → Llama	T2T	46.9	55.7	9.4	18.3	89.4	70.9	30.4	44.5	55.76
	LCF-X	42.1	49.8	17.8	28.3	92.9	68.8	40.2	53.5	58.66
	XKV	**54.0**	**58.7**	**19.4**	**30.0**	**93.0**	**74.1**	**41.5**	**55.1**	**62.18**
Llama → Qwen	T2T	38.6	47.7	5.9	15.4	**85.0**	65.6	21.1	32.9	49.32
	LCF-X	46.7	54.8	7.6	15.5	81.1	53.1	28.6	40.3	48.96
	XKV	**53.0**	**59.0**	**8.5**	**16.5**	82.1	**65.8**	**29.3**	**40.9**	**52.86**
Llama → Gemma	T2T	40.3	48.3	**12.1**	**21.9**	**55.9**	53.7	**28.3**	**41.5**	**44.26**
	LCF-X	45.1	50.4	5.4	12.4	44.9	52.2	23.5	34.4	38.86
	XKV	**48.3**	**55.7**	5.2	12.2	54.9	**57.3**	24.4	35.0	43.02
Llama → Llama	T2T	48.4	55.7	14.2	26.3	90.7	72.1	36.1	52.0	59.36
	LCF-X	45.6	52.8	16.5	26.5	92.4	66.4	40.8	54.0	58.42
	XKV	**50.4**	**56.8**	**19.9**	**30.3**	**93.1**	**73.9**	**42.1**	**55.7**	**61.96**

Parameter sharing and complexity. Direct pooling is linear in prompt length, $O(\sum_a L_a H_a k N_a D_a)$, and the cross-layer self-attention operates on only L_R memory tokens. The position decoder attends from each receiver position to those L_R tokens rather than performing self-attention over all $L_R N_R$ raw cache rows. Moreover, the bottleneck, translation blocks, position decoder, gate, and output heads are all shared across receiver layers, in contrast to LCF-X's independent per-layer projectors, so the only depth-dependent parameters are the pooling queries, the layer embeddings, and the layer map. Together these choices keep the translator compact while retaining position-specific updates.

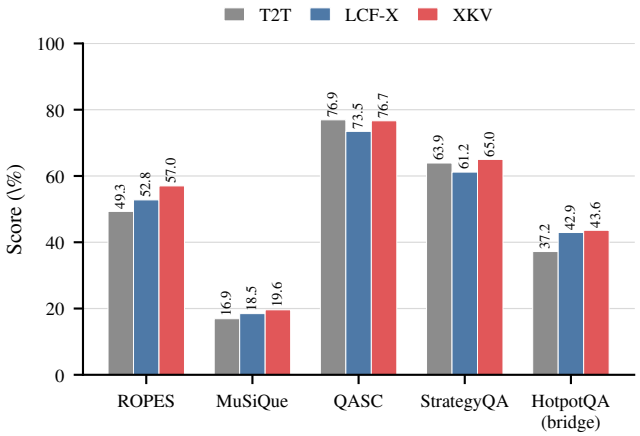

Figure 4: Dataset-level mean primary scores over the full 3×3 sharer–receiver grid. Each dataset forms one group and each bar is a communication method; we use F1 for generative QA and accuracy for classification. XKV is best on four of the five datasets.

Experimental Setup

Task setting and models

We evaluate communication in the *cross-context* setting: the Sharer and the Receiver each observe only part of the evidence, and the receiver must combine its own context with information received from the sharer. The question and, for classification tasks, the answer choices are visible to both models; only the evidence is partitioned. We use a deterministic per-example random assignment so that the two contexts are disjoint without systematically assigning a particular evidence type to one role. Specifically, ROPES assigns the background and situation passages one per side; QASC assigns its two supporting facts one per side; 2-hop MuSiQue and HotpotQA-bridge place one supporting paragraph on each side and divide the distractors between them (four per side for HotpotQA); and StrategyQA shuffles its supporting facts and splits them into two nonempty halves. Thus, neither model

receives all annotated supporting evidence. We use three frozen decoder-only language models spanning three families, Qwen3-0.6B (Yang et al. 2025), Gemma-3-1B (Gemma Team 2025), and Llama-3.2-3B (Grattafiori et al. 2024; Meta AI 2024), and evaluate the complete 3×3 ordered grid in which any model may act as sharer or receiver, yielding six heterogeneous off-diagonal pairings and three homogeneous diagonal pairings. Collectively, the off-diagonal pairs span differences in depth, attention geometry, and tokenizer; we assume neither token-level nor layer-level correspondence. All reported numbers are means over these nine pairings unless stated otherwise.

Datasets

We evaluate on five split-evidence benchmarks: **ROPES** (Lin et al. 2019), **MuSiQue** (Trivedi et al. 2022), and **HotpotQA-bridge** (Yang et al. 2018) for generative multi-hop question answering, and **QASC** (Khot et al. 2020) and **StrategyQA** (Geva et al. 2021) for classification-style reasoning. For each example, we partition the evidence so that the sharer and receiver hold complementary contexts and the receiver cannot solve the task reliably from its own context alone. For the generative QA datasets we report exact match (EM) and F1; for the classification datasets we report accuracy. In the main text, we use F1 for generative QA and accuracy for classification as the primary score.

Baselines

We compare XKV with two communication baselines. **T2T** is a text-to-text pipeline: the sharer generates a natural-language message from its private context and the question, and the receiver answers from its own context plus that message. This is the standard communication pattern in deployed agent systems (Du et al. 2024; Wu et al. 2024; Chen et al. 2024), but it incurs autoregressive message generation and re-encoding. **LCF-X**, the cross-context extension of Latent Cache Flow (Rossi, Raghunath, and Wu 2026), compresses the sharer cache into a position-free summary injected into the receiver as gated KV residuals; it is the most relevant latent baseline for our setting because it removes the shared-token assumption of earlier cache-to-cache methods (Fu et al. 2026; Dery et al. 2026). Since the original cross-context design assumes matched layer count and KV dimensions and was evaluated only on Qwen3-0.6B $\rightarrow$ Qwen3-0.6B, we strengthen it for the six heterogeneous cells with our learned $L_S \rightarrow L_R$ layer-axis map, a per-layer down-projection accepting concatenated sharer and receiver features of different widths, and an up-projection emitting receiver-native KV residuals; LCF-X in the experiments refers to this project-authored heterogeneous extension. **XKV** is our receiver-aware latent protocol, which builds a compact joint memory from both caches and lets each receiver position retrieve its own receiver-native residual update.

Training and evaluation details

For XKV and LCF-X, only the lightweight communication module is trained; all sharer and receiver weights remain frozen. Where applicable, we use the same default recipe across datasets, with the optimizer, learning-rate schedule, batch size, and decoding settings summarized in the technical appendix. We measure two kinds of efficiency: *communication latency* covers only the learned latent module (the fusor for LCF-X, the translator for XKV), while *end-to-end latency* covers the full per-example pipeline of sharer forward pass, communication step, and receiver-side answering. Since T2T has neither module, we report only its end-to-end latency, which includes autoregressive message generation and receiver re-encoding. Every experiment is run with Nvidia A100 80GB, Ubuntu 22.04 LTS, and CUDA 12.2 once.

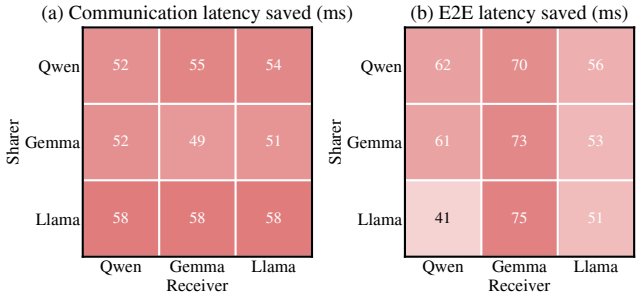

Figure 5: Pair-level communication and end-to-end latency savings relative to the faster baseline, averaged over five datasets. Rows denote sharers and columns receivers. (a) Communication latency saved by XKV (ms). (b) End-to-end latency saved by XKV. Positive values mean XKV is faster.

Results

Overall comparison

Table 1 reports the complete pair-level results for all nine ordered sharer–receiver settings, and Figure 4 aggregates them into dataset-level means (F1 for generative QA, accuracy for classification). Per-dataset score matrices and EM breakdowns are provided in the technical appendix.

XKV is the strongest protocol overall: it attains the best or tied-best cross-dataset average in eight of the nine settings and the best result in 31 of the 45 dataset–pair cells. It improves the LCF-X mean on every dataset ($+2.61$ macro points, up to $+4.2$ F1 on ROPES) and leads T2T on four of the five datasets ($+3.52$ macro points, including $+7.7$ F1 on ROPES and $+6.4$ F1 on HotpotQA-bridge), trailing only on QASC by 0.27 points. Figure 4 also shows that the baselines are complementary: LCF-X is stronger on the three generative QA datasets and T2T on the two classification datasets. XKV is the only protocol competitive in both regimes, which is why its advantage survives aggregation. These quality gains cost nothing in latency: in every ordered pair, Figure 5 shows XKV saving 49–58 ms of communication latency and 41–75 ms end to end relative to the faster baseline, averaged over the five datasets.

Efficiency

Table 2 breaks down communication (Comm.) and end-to-end (E2E) latency per dataset for the three same-model settings; T2T has no latent module, so only its E2E latency is defined. Latency grids for all nine pairings appear in the technical appendix.

Table 2: Latency results for same-model sharer–receiver settings. Each dataset reports latency in milliseconds per example; lower is better. Communication (Comm.) is fusor-only time for LCF-X and translator-only time for XKV. Since T2T has neither module, only its end-to-end (E2E) latency is reported. Cross-dataset Avg. averages the available measurement over five datasets.

Sharer → Receiver	Method	ROPES		MuSiQue		QASC		StrategyQA		HotpotQA-bridge		Cross-dataset Avg.	
		Comm.	E2E	Comm.	E2E	Comm.	E2E	Comm.	E2E	Comm.	E2E	Comm.	E2E
Qwen → Qwen	T2T	–	955	–	974	–	328	–	848	–	995	–	820.0
	LCF-X	31	165	115	376	42	156	42	154	64	259	58.8	222.0
	XKV	**6**	**139**	**6**	**218**	**6**	**119**	**10**	**122**	**6**	**201**	**6.8**	**159.8**
Gemma → Gemma	T2T	–	1549	–	1225	–	790	–	1884	–	1372	–	1364.0
	LCF-X	39	213	84	427	45	188	44	215	58	294	54.0	267.4
	XKV	**5**	**171**	**5**	**264**	**5**	**145**	**5**	**146**	**5**	**248**	**5.0**	**194.8**
Llama → Llama	T2T	–	1282	–	1421	–	842	–	1296	–	1377	–	1243.6
	LCF-X	41	147	123	325	46	135	46	139	64	215	64.0	192.2
	XKV	**6**	**111**	**6**	**223**	**6**	**94**	**6**	**105**	**6**	**174**	**6.0**	**141.4**

Table 3: Ablation results over QASC, ROPES, and StrategyQA. Score changes are relative to full XKV and use accuracy for QASC and StrategyQA and F1 for ROPES. "Comm." is translator-only latency.

Variant	QASC Δ	ROPES F1 Δ	StrategyQA Δ	Mean Δ	Comm. (ms)	E2E (ms)	Params
Full XKV	0.00	0.00	0.00	0.00	5.8	131.7	4.55M
−RP	-0.41	-0.82	-0.16	-0.46	4.7	128.7	3.58M
−RX	-1.98	-0.44	-0.08	-0.83	3.7	127.1	3.43M
LCFP	-0.82	-2.00	+0.01	-0.94	39.1	183.7	3.53M

Two patterns stand out. First, the XKV translator is nearly constant at 5–10 ms in every cell, whereas the LCF-X fusor ranges from 31 to 123 ms and grows with context length, costing two to three times more on MuSiQue than on QASC or StrategyQA. The reason is structural: direct pooling is a single batched operation, while span pooling must process a number of variable-length spans that scales with the prompt. Second, although both latent methods pay the same two frozen prefills, XKV is fastest end to end in all fifteen cells, cutting E2E time by 26–28% relative to LCF-X; T2T is $5.1\times$ to $8.8\times$ slower than XKV because autoregressive message generation dominates its budget. Aggregated over all 45 dataset–pair cells, the XKV translator averages 5.8 ms against 59.9 ms for the LCF-X fusor ($10.3\times$ faster), lowers mean E2E latency from 227.9 to 167.6 ms (26.4%), runs $6.8\times$ faster end to end than T2T, and trains 76.1% fewer parameters than LCF-X (4.55M against 19.03M). XKV is thus simultaneously the most accurate and the cheapest channel in our comparison.

Ablation study

We ablate XKV on QASC, ROPES, and StrategyQA over all 27 matched dataset–model-pair settings, removing one design decision at a time. −RP builds the memory from a sharer-only summary as in LCF-X, testing whether *what to communicate* should depend on the receiver; −RX replaces position-specific retrieval from the raw receiver cache with broadcast layer memory, testing whether receiver positions should explicitly query the joint memory;**LCFP** swaps our vectorized direct pooling for LCF-X's two-stage span pooling. Table 3 reports score changes relative to Full XKV together with average latency and parameter counts; per-pair matrices are in the technical appendix. Both receiver-aware components earn their small cost. Removing receiver pooling lowers the score on all three datasets (-0.46 on average) while saving only 1.1 ms and 0.97M parameters. Removing receiver cross-attention costs more (-0.83 on average, -1.98 on QASC, where evidence must be integrated selectively), showing that broadcast layer memory cannot substitute for receiver-position queries into the joint memory. Since neither removal changes E2E latency by more than 3.5%, receiver-awareness is essentially free at inference time. LCFP shows that XKV's efficiency and quality are not in tension: it raises translator latency from 5.8 to 39.1 ms ($6.7\times$) and E2E latency by 39.4%, yet also lowers the mean score by 0.94 points, including 2.00 F1 on ROPES. Direct pooling is thus not merely an implementation optimization; replacing it makes the translator both slower and less accurate.

Conclusion

We presented XKV, a latent communication protocol for frozen, heterogeneous language models holding complementary evidence. XKV decides *what to communicate* jointly, pooling both KV caches into a compact cross-layer memory, and lets every receiver cache position query that memory for its own receiver-native residual. Across 45 settings (five split-evidence datasets, six heterogeneous and three same-model ordered pairs), XKV attains the best overall score and improves on LCF-X on every dataset, with a translator $10.3\times$ faster than the LCF-X fusor, 76.1% fewer trainable parameters, and end-to-end communication $6.8\times$ faster than text exchange; ablations confirm that both receiver-aware components add quality at negligible cost. Receiver-aware, position-query cache communication thus removes the trade-off between quality and efficiency. Future work includes scaling to larger models, multi-turn and many-agent communication, and interpreting what the latent messages transmit.

References

Bansal, Y.; Nakkiran, P.; and Barak, B. 2021. Revisiting Model Stitching to Compare Neural Representations. In *Advances in Neural Information Processing Systems*.

Chen, W.; Su, Y.; Zuo, J.; Yang, C.; Yuan, C.; Chan, C.-M.; Yu, H.; Lu, Y.; Hung, Y.-H.; Qian, C.; Qin, Y.; Cong, X.; Xie, R.; Liu, Z.; Sun, M.; and Zhou, J. 2024. AgentVerse: Facilitating Multi-Agent Collaboration and Exploring Emergent Behaviors. In *International Conference on Learning Representations*.

DeepSeek-AI. 2024. DeepSeek-V2: A Strong, Economical, and Efficient Mixture-of-Experts Language Model. *arXiv preprint arXiv:2405.04434*.

Dery, L. M.; Yahav, Z.; Prior, H.; Feng, Q.; Shen, J.; and Szlam, A. 2026. Latent Space Communication via K-V Cache Alignment. *arXiv preprint arXiv:2601.06123*.

Du, Y.; Li, S.; Torralba, A.; Tenenbaum, J. B.; and Mordatch, I. 2024. Improving Factuality and Reasoning in Language Models through Multiagent Debate. In *International Conference on Machine Learning*.

Du, Z.; Wang, R.; Bai, H.; Cao, Z.; Zhu, X.; Zheng, B.; Chen, W.; and Ying, H. 2026. Enabling Agents to Communicate Entirely in Latent Space. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics*.

Fu, T.; Min, Z.; Zhang, H.; Yan, J.; Dai, G.; Ouyang, W.; and Wang, Y. 2026. Cache-to-Cache: Direct Semantic Communication Between Large Language Models. In *International Conference on Learning Representations*. ArXiv:2510.03215.

Gemma Team. 2025. Gemma 3 Technical Report. *arXiv preprint arXiv:2503.19786*.

Geva, M.; Khashabi, D.; Segal, E.; Khot, T.; Roth, D.; and Berant, J. 2021. Did Aristotle Use a Laptop? A Question Answering Benchmark with Implicit Reasoning Strategies. *Transactions of the Association for Computational Linguistics*, 9: 346–361.

Grattafiori, A.; Dubey, A.; Jauhri, A.; Pandey, A.; Kadian, A.; et al. 2024. The Llama 3 Herd of Models. *arXiv preprint arXiv:2407.21783*.

Hao, S.; Sukhbaatar, S.; Su, D.; Li, X.; Hu, Z.; Weston, J.; and Tian, Y. 2025. Training Large Language Models to Reason in a Continuous Latent Space. In *International Conference on Learning Representations*.

Jaegle, A.; Gimeno, F.; Brock, A.; Vinyals, O.; Zisserman, A.; and Carreira, J. 2021. Perceiver: General Perception with Iterative Attention. In *Proceedings of the 38th International Conference on Machine Learning*, 4651–4664.

Khot, T.; Clark, P.; Guerquin, M.; Jansen, P.; and Sabharwal, A. 2020. QASC: A Dataset for Question Answering via Sentence Composition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 8082–8090.

Kornblith, S.; Norouzi, M.; Lee, H.; and Hinton, G. 2019. Similarity of Neural Network Representations Revisited. In *International Conference on Machine Learning*.

Lee, J.; Lee, Y.; Kim, J.; Kosiorek, A. R.; Choi, S.; and Teh, Y. W. 2019. Set Transformer: A Framework for Attention-based Permutation-Invariant Neural Networks. In *Proceedings of the 36th International Conference on Machine Learning*, 3744–3753.

Li, X. L.; and Liang, P. 2021. Prefix-Tuning: Optimizing Continuous Prompts for Generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics*.

Li, Y.; Huang, Y.; Yang, B.; Venkitesh, B.; Locatelli, A.; Ye, H.; Cai, T.; Lewis, P.; and Chen, D. 2024. SnapKV: LLM Knows What You are Looking for Before Generation. In *Advances in Neural Information Processing Systems*.

Lin, K.; Tafjord, O.; Clark, P.; and Gardner, M. 2019. Reasoning Over Paragraph Effects in Situations. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, 58–62.

Meta AI. 2024. Llama 3.2 Model Card. https://github.com/meta-llama/llama-models/blob/main/models/llama3_2/MODEL_CARD.md.

Moschella, L.; Maiorca, V.; Fumero, M.; Norelli, A.; Locatello, F.; and Rodolà, E. 2023. Relative Representations Enable Zero-shot Latent Space Communication. In *International Conference on Learning Representations*.

Mu, J.; Li, X. L.; and Goodman, N. 2023. Learning to Compress Prompts with Gist Tokens. In *Advances in Neural Information Processing Systems*.

Pope, R.; Douglas, S.; Chowdhery, A.; Devlin, J.; Bradbury, J.; Levskaya, A.; Heek, J.; Xiao, K.; Agrawal, S.; and Dean, J. 2023. Efficiently Scaling Transformer Inference. In *Proceedings of Machine Learning and Systems*.

Rossi, M.; Raghunath, P.; and Wu, E. 2026. Latent Cache Flow: Model-to-Model Communication Without Text. *arXiv preprint arXiv:2605.22863*.

Shi, X.; Chiesa, M.; Maguire Jr., G. Q.; and Kostic, D. 2026. KVComm: Enabling Efficient LLM Communication through Selective KV Sharing. In *International Conference on Learning Representations*.

Trivedi, H.; Balasubramanian, N.; Khot, T.; and Sabharwal, A. 2022. MuSiQue: Multihop Questions via Single-hop Question Composition. *Transactions of the Association for Computational Linguistics*, 10: 539–554.

Wu, Q.; Bansal, G.; Zhang, J.; Wu, Y.; Li, B.; Zhu, E.; Jiang, L.; Zhang, X.; Zhang, S.; Liu, J.; Awadallah, A. H.; White, R. W.; Burger, D.; and Wang, C. 2024. AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation Framework. In *First Conference on Language Modeling*.

Xiao, G.; Tian, Y.; Chen, B.; Han, S.; and Lewis, M. 2024. Efficient Streaming Language Models with Attention Sinks. In *International Conference on Learning Representations*.

Yang, A.; Li, A.; Yang, B.; Zhang, B.; Hui, B.; Zheng, B.; Yu, B.; Gao, C.; Huang, C.; Lv, C.; et al. 2025. Qwen3 Technical Report. *arXiv preprint arXiv:2505.09388*.

Yang, Z.; Qi, P.; Zhang, S.; Bengio, Y.; Cohen, W. W.; Salakhutdinov, R.; and Manning, C. D. 2018. HotpotQA:

A Dataset for Diverse, Explainable Multi-hop Question Answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2369–2380.

Yao, S.; Zhao, J.; Yu, D.; Du, N.; Shafran, I.; Narasimhan, K.; and Cao, Y. 2023. ReAct: Synergizing Reasoning and Acting in Language Models. In *International Conference on Learning Representations*.

Ye, H.; Gao, Z.; Ma, M.; Wang, Q.; Fu, Y.; Chung, M.-Y.; Lin, Y.; Liu, Z.; Zhang, J.; Zhuo, D.; and Chen, Y. 2025. KVCOMM: Online Cross-context KV-cache Communication for Efficient LLM-based Multi-agent Systems. In *Advances in Neural Information Processing Systems*.

Zhang, Z.; Sheng, Y.; Zhou, T.; Chen, T.; Zheng, L.; Cai, R.; Song, Z.; Tian, Y.; Ré, C.; Barrett, C.; Wang, Z.; and Chen, B. 2023. H2O: Heavy-Hitter Oracle for Efficient Generative Inference of Large Language Models. In *Advances in Neural Information Processing Systems*.

Zou, J.; Yang, X.; Qiu, R.; Li, G.; Tieu, K.; Lu, P.; Shen, K.; Tong, H.; Choi, Y.; He, J.; Zou, J.; Wang, M.; and Yang, L. 2025. Latent Collaboration in Multi-Agent Systems. *arXiv preprint arXiv:2511.20639*.

+

Table 4: Appendix backup: full latency results for all nine ordered sharer–receiver settings. Each dataset reports latency in milliseconds per example; lower is better. Communication (Comm.) is fusor-only time for LCF-X and translator-only time for XKV. Because T2T has neither module, only its end-to-end (E2E) latency is reported. Cross-dataset Avg. averages the available measurement over five datasets.

Sharer → Receiver	Method	ROPES		MuSiQue		QASC		StrategyQA		HotpotQA-bridge		Cross-dataset Avg.	
		Comm.	E2E	Comm.	E2E	Comm.	E2E	Comm.	E2E	Comm.	E2E	Comm.	E2E
Qwen → Qwen	T2T	–	955	–	974	–	328	–	848	–	995	–	820.0
	LCF-X	31	165	115	376	42	156	42	154	64	259	58.8	222.0
	XKV	**6**	**139**	**6**	**218**	**6**	**119**	**10**	**122**	**6**	**201**	**6.8**	**159.8**
Qwen → Gemma	T2T	–	914	–	971	–	291	–	855	–	1001	–	806.4
	LCF-X	35	193	118	439	42	177	41	177	65	295	60.2	256.2
	XKV	**5**	**160**	**5**	**253**	**5**	**136**	**5**	**140**	**5**	**240**	**5.0**	**185.8**
Qwen → Llama	T2T	–	927	–	1006	–	247	–	807	–	999	–	797.2
	LCF-X	36	148	116	331	42	139	42	139	65	224	60.2	196.2
	XKV	**6**	**119**	**6**	**201**	**6**	**102**	**6**	**109**	**6**	**169**	**6.0**	**140.0**
Gemma → Qwen	T2T	–	1605	–	1236	–	810	–	1882	–	1332	–	1373.0
	LCF-X	42	182	91	370	48	171	48	170	62	266	58.2	231.8
	XKV	**6**	**159**	**6**	**229**	**6**	**127**	**6**	**126**	**7**	**213**	**6.2**	**170.8**
Gemma → Gemma	T2T	–	1549	–	1225	–	790	–	1884	–	1372	–	1364.0
	LCF-X	39	213	84	427	45	188	44	215	58	294	54.0	267.4
	XKV	**5**	**171**	**5**	**264**	**5**	**145**	**5**	**146**	**5**	**248**	**5.0**	**194.8**
Gemma → Llama	T2T	–	1566	–	1274	–	742	–	1823	–	1365	–	1354.0
	LCF-X	42	163	85	300	48	154	48	157	60	225	56.6	199.8
	XKV	**6**	**126**	**6**	**212**	**6**	**108**	**6**	**112**	**6**	**178**	**6.0**	**147.2**
Llama → Qwen	T2T	–	1301	–	1330	–	925	–	1344	–	1387	–	1257.4
	LCF-X	42	171	120	372	47	155	48	**157**	64	257	64.2	222.4
	XKV	**6**	**134**	**6**	**238**	**6**	**110**	**6**	217	**6**	**207**	**6.0**	**181.2**
Llama → Gemma	T2T	–	1219	–	1333	–	880	–	1361	–	1407	–	1240.0
	LCF-X	40	197	118	435	46	176	47	213	64	293	63.0	262.8
	XKV	**5**	**153**	**5**	**278**	**5**	**133**	**5**	**134**	**5**	**239**	**5.0**	**187.4**
Llama → Llama	T2T	–	1282	–	1421	–	842	–	1296	–	1377	–	1243.6
	LCF-X	41	147	123	325	46	135	46	139	64	215	64.0	192.2
	XKV	**6**	**111**	**6**	**223**	**6**	**94**	**6**	**105**	**6**	**174**	**6.0**	**141.4**

Table 5: Default training / inference hyperparameters.

Setting	Value
Optimizer	AdamW
Learning rate	1×10^{-4}
LR schedule	linear warmup (10% of total steps)
Effective batch size	256
Gradient clipping	1.0
Pooling spans P	4
Translator bottleneck dim	128
Validation holdout	2.8% of train (when no explicit eval split)
Receiver max new tokens	64
Seed	0

Table 6: ROPES EM/F1 scores (%). Higher is better; cell-wise best values are bold. Mean across the nine model pairs: LCF-X 45.86/52.79; XKV 50.48/56.99; T2T 40.74/49.31.

Sharer	Receiver	LCF-X	XKV	T2T
Qwen-0.6B	Qwen-0.6B	45.2/52.4	**47.9/55.6**	35.8/45.2
Qwen-0.6B	Gemma-1B	43.9/51.3	**46.2/54.3**	42.4/48.6
Qwen-0.6B	Llama-3B	51.4/58.5	**57.3/61.8**	50.6/58.0
Gemma-1B	Qwen-0.6B	47.2/53.7	**49.1/55.4**	31.9/42.5
Gemma-1B	Gemma-1B	45.5/51.4	**48.1/55.6**	31.8/42.1
Gemma-1B	Llama-3B	42.1/49.8	**54.0/58.7**	46.9/55.7
Llama-3B	Qwen-0.6B	46.7/54.8	**53.0/59.0**	38.6/47.7
Llama-3B	Gemma-1B	45.1/50.4	**48.3/55.7**	40.3/48.3
Llama-3B	Llama-3B	45.6/52.8	**50.4/56.8**	48.4/55.7

Table 7: ROPES latency in milliseconds per example. Communication is measured only for the latent modules (LCF-X fusor and XKV translator); T2T therefore reports end-to-end latency only. Lower is better; cell-wise minima at the displayed precision are bold. Means across the nine model pairs: LCF-X 38.7/175.4; XKV 5.7/141.3; T2T E2E 1257.6.

Sharer	Receiver	LCF-X		XKV		T2T
		Comm.	E2E	Comm.	E2E	E2E
Qwen-0.6B	Qwen-0.6B	31	165	**6**	**139**	955
Qwen-0.6B	Gemma-1B	35	193	**5**	**160**	914
Qwen-0.6B	Llama-3B	36	148	**6**	**119**	927
Gemma-1B	Qwen-0.6B	42	182	**6**	**159**	1605
Gemma-1B	Gemma-1B	39	213	**5**	**171**	1549
Gemma-1B	Llama-3B	42	163	**6**	**126**	1566
Llama-3B	Qwen-0.6B	42	171	**6**	**134**	1301
Llama-3B	Gemma-1B	40	197	**5**	**153**	1219
Llama-3B	Llama-3B	41	147	**6**	**111**	1282

Table 8: MuSiQue EM/F1 scores (%). Higher is better; cell-wise best values are bold. Mean across the nine model pairs: LCF-X 10.00/18.47; XKV 11.07/19.59; T2T 7.99/16.92.

Sharer	Receiver	LCF-X	XKV	T2T
Qwen-0.6B	Qwen-0.6B	7.7/15.4	**8.1/16.0**	4.6/12.2
Qwen-0.6B	Gemma-1B	6.0/13.3	5.4/12.5	**6.8/14.5**
Qwen-0.6B	Llama-3B	16.0/26.3	**19.2/29.9**	9.3/19.5
Gemma-1B	Qwen-0.6B	**8.4/16.6**	8.1/15.8	3.4/10.8
Gemma-1B	Gemma-1B	4.6/11.9	5.8/13.1	**6.2/13.4**
Gemma-1B	Llama-3B	17.8/28.3	**19.4/30.0**	9.4/18.3
Llama-3B	Qwen-0.6B	7.6/15.5	**8.5/16.5**	5.9/15.4
Llama-3B	Gemma-1B	5.4/12.4	5.2/12.2	**12.1/21.9**
Llama-3B	Llama-3B	16.5/26.5	**19.9/30.3**	14.2/26.3

Table 9: MuSiQue latency in milliseconds per example. Communication is measured only for the latent modules (LCF-X fusor and XKV translator); T2T therefore reports end-to-end latency only. Lower is better; cell-wise minima at the displayed precision are bold. Means across the nine model pairs: LCF-X 107.8/375.0; XKV 5.7/235.1; T2T E2E 1196.7.

Sharer	Receiver	LCF-X		XKV		T2T
		Comm.	E2E	Comm.	E2E	E2E
Qwen-0.6B	Qwen-0.6B	115	376	**6**	**218**	974
Qwen-0.6B	Gemma-1B	118	439	**5**	**253**	971
Qwen-0.6B	Llama-3B	116	331	**6**	**201**	1006
Gemma-1B	Qwen-0.6B	91	370	**6**	**229**	1236
Gemma-1B	Gemma-1B	84	427	**5**	**264**	1225
Gemma-1B	Llama-3B	85	300	**6**	**212**	1274
Llama-3B	Qwen-0.6B	120	372	**6**	**238**	1330
Llama-3B	Gemma-1B	118	435	**5**	**278**	1333
Llama-3B	Llama-3B	123	325	**6**	**223**	1421

Table 10: QASC accuracy scores (%). Higher is better; cell-wise best values are bold. Mean across the nine model pairs: LCF-X 73.44; XKV 76.67; T2T 76.93.

Sharer	Receiver	LCF-X	XKV	T2T
Qwen-0.6B	Qwen-0.6B	**82.4**	82.2	78.6
Qwen-0.6B	Gemma-1B	49.2	55.1	**65.8**
Qwen-0.6B	Llama-3B	92.0	**93.3**	86.7
Gemma-1B	Qwen-0.6B	81.3	82.4	**84.6**
Gemma-1B	Gemma-1B	44.8	53.9	**55.7**
Gemma-1B	Llama-3B	92.9	**93.0**	89.4
Llama-3B	Qwen-0.6B	81.1	82.1	**85.0**
Llama-3B	Gemma-1B	44.9	54.9	**55.9**
Llama-3B	Llama-3B	92.4	**93.1**	90.7

Table 11: QASC latency in milliseconds per example. Communication is measured only for the latent modules (LCF-X fusor and XKV translator); T2T therefore reports end-to-end latency only. Lower is better; cell-wise minima at the displayed precision are bold. Means across the nine model pairs: LCF-X 45.1/161.2; XKV 5.7/119.3; T2T E2E 650.6.

Sharer	Receiver	LCF-X		XKV		T2T
		Comm.	E2E	Comm.	E2E	E2E
Qwen-0.6B	Qwen-0.6B	42	156	**6**	**119**	328
Qwen-0.6B	Gemma-1B	42	177	**5**	**136**	291
Qwen-0.6B	Llama-3B	42	139	**6**	**102**	247
Gemma-1B	Qwen-0.6B	48	171	**6**	**127**	810
Gemma-1B	Gemma-1B	45	188	**5**	**145**	790
Gemma-1B	Llama-3B	48	154	**6**	**108**	742
Llama-3B	Qwen-0.6B	47	155	**6**	**110**	925
Llama-3B	Gemma-1B	46	176	**5**	**133**	880
Llama-3B	Llama-3B	46	135	**6**	**94**	842

Table 12: StrategyQA accuracy scores (%). Higher is better; cell-wise best values are bold. Mean across the nine model pairs: LCF-X 61.17; XKV 65.01; T2T 63.91.

Sharer	Receiver	LCF-X	XKV	T2T
Qwen-0.6B	Qwen-0.6B	58.8	61.4	**64.1**
Qwen-0.6B	Gemma-1B	**60.8**	58.2	51.9
Qwen-0.6B	Llama-3B	72.9	**73.6**	73.4
Gemma-1B	Qwen-0.6B	59.3	62.0	**65.9**
Gemma-1B	Gemma-1B	58.2	**58.8**	57.6
Gemma-1B	Llama-3B	68.8	**74.1**	70.9
Llama-3B	Qwen-0.6B	53.1	**65.8**	65.6
Llama-3B	Gemma-1B	52.2	**57.3**	53.7
Llama-3B	Llama-3B	66.4	**73.9**	72.1

Table 13: StrategyQA latency in milliseconds per example. Communication is measured only for the latent modules (LCF-X fusor and XKV translator); T2T therefore reports end-to-end latency only. Lower is better; cell-wise minima at the displayed precision are bold. Means across the nine model pairs: LCF-X 45.1/169.0; XKV 6.1/134.6; T2T E2E 1344.4.

Sharer	Receiver	LCF-X		XKV		T2T
		Comm.	E2E	Comm.	E2E	E2E
Qwen-0.6B	Qwen-0.6B	42	154	**10**	**122**	848
Qwen-0.6B	Gemma-1B	41	177	**5**	**140**	855
Qwen-0.6B	Llama-3B	42	139	**6**	**109**	807
Gemma-1B	Qwen-0.6B	48	170	**6**	**126**	1882
Gemma-1B	Gemma-1B	44	215	**5**	**146**	1884
Gemma-1B	Llama-3B	48	157	**6**	**112**	1823
Llama-3B	Qwen-0.6B	48	**157**	**6**	217	1344
Llama-3B	Gemma-1B	47	213	**5**	**134**	1361
Llama-3B	Llama-3B	46	139	**6**	**105**	1296

Table 14: HotpotQA-bridge EM/F1 scores (%). Higher is better; cell-wise best values are bold. Mean across the nine model pairs: LCF-X 31.06/42.96; XKV 31.72/43.61; T2T 24.72/37.20.

Sharer	Receiver	LCF-X	XKV	T2T
Qwen-0.6B	Qwen-0.6B	29.1/40.6	**29.5/40.7**	16.6/27.1
Qwen-0.6B	Gemma-1B	23.3/34.0	**23.8/34.7**	22.4/34.0
Qwen-0.6B	Llama-3B	41.1/54.5	**41.5/55.0**	30.4/44.0
Gemma-1B	Qwen-0.6B	29.4/40.9	**29.5/41.0**	15.5/25.7
Gemma-1B	Gemma-1B	23.5/**34.4**	**23.9/34.4**	21.7/33.1
Gemma-1B	Llama-3B	40.2/53.5	**41.5/55.1**	30.4/44.5
Llama-3B	Qwen-0.6B	28.6/40.3	**29.3/40.9**	21.1/32.9
Llama-3B	Gemma-1B	23.5/34.4	24.4/35.0	**28.3/41.5**
Llama-3B	Llama-3B	40.8/54.0	**42.1/55.7**	36.1/52.0

Table 15: HotpotQA-bridge latency in milliseconds per example. Communication is measured only for the latent modules (LCF-X fusor and XKV translator); T2T therefore reports end-to-end latency only. Lower is better; cell-wise minima at the displayed precision are bold. Means across the nine model pairs: LCF-X 62.9/258.7; XKV 5.8/207.7; T2T E2E 1248.3.

Sharer	Receiver	LCF-X		XKV		T2T
		Comm.	E2E	Comm.	E2E	E2E
Qwen-0.6B	Qwen-0.6B	64	259	**6**	**201**	995
Qwen-0.6B	Gemma-1B	65	295	**5**	**240**	1001
Qwen-0.6B	Llama-3B	65	224	**6**	**169**	999
Gemma-1B	Qwen-0.6B	62	266	**7**	**213**	1332
Gemma-1B	Gemma-1B	58	294	**5**	**248**	1372
Gemma-1B	Llama-3B	60	225	**6**	**178**	1365
Llama-3B	Qwen-0.6B	64	257	**6**	**207**	1387
Llama-3B	Gemma-1B	64	293	**5**	**239**	1407
Llama-3B	Llama-3B	64	215	**6**	**174**	1377

Table 16: QASC accuracy scores (%). Higher is better; cell-wise best values are bold. Mean across the nine model pairs: Full 76.67; −RP 76.26; −RX 74.69; LCFP 75.84.

Sharer	Receiver	Full	−RP	−RX	LCFP
Qwen-0.6B	Qwen-0.6B	**82.2**	81.6	80.8	81.7
Qwen-0.6B	Gemma-1B	**55.1**	53.8	49.6	51.9
Qwen-0.6B	Llama-3B	**93.3**	93.1	92.8	93.1
Gemma-1B	Qwen-0.6B	**82.4**	81.9	81.3	81.3
Gemma-1B	Gemma-1B	53.9	52.3	49.6	**54.0**
Gemma-1B	Llama-3B	93.0	**93.6**	93.2	93.5
Llama-3B	Qwen-0.6B	**82.1**	81.9	81.1	80.9
Llama-3B	Gemma-1B	**54.9**	54.6	51.0	53.2
Llama-3B	Llama-3B	93.1	**93.5**	92.8	93.0

Table 17: QASC ablation latency in milliseconds per example. Entries are translator/total; lower is better and cell-wise minima are bold. Mean across the nine model pairs: Full 5.7/119.3; −RP 4.7/120.2; −RX 3.7/118.4; LCFP 37.4/171.3.

Sharer	Receiver	Full	−RP	−RX	LCFP
Qwen-0.6B	Qwen-0.6B	6/119	5/118	**4/115**	48/320
Qwen-0.6B	Gemma-1B	5/**136**	4/139	**3**/138	31/167
Qwen-0.6B	Llama-3B	6/102	5/102	**4/101**	33/130
Gemma-1B	Qwen-0.6B	6/127	5/128	**4/123**	39/161
Gemma-1B	Gemma-1B	5/**145**	4/146	**3**/147	36/178
Gemma-1B	Llama-3B	6/108	5/110	**4/106**	38/146
Llama-3B	Qwen-0.6B	6/110	5/110	**4/109**	38/146
Llama-3B	Gemma-1B	5/133	4/134	**3/132**	36/166
Llama-3B	Llama-3B	6/**94**	5/95	**4**/95	38/128

Table 18: ROPES EM/F1 scores (%). Higher is better; cell-wise best values are bold. Mean across the nine model pairs: Full 50.48/56.99; −RP 49.38/56.17; −RX 49.79/56.54; LCFP 47.77/54.99.

Sharer	Receiver	Full	−RP	−RX	LCFP
Qwen-0.6B	Qwen-0.6B	47.9/55.6	**49.5/56.6**	46.3/53.5	45.6/52.7
Qwen-0.6B	Gemma-1B	46.2/**54.3**	**47.0**/54.1	45.5/53.7	46.6/53.9
Qwen-0.6B	Llama-3B	**57.3/61.8**	53.3/59.2	55.6/61.2	53.7/60.3
Gemma-1B	Qwen-0.6B	**49.1/55.4**	45.6/52.1	44.9/51.7	44.7/53.6
Gemma-1B	Gemma-1B	**48.1/55.6**	46.7/54.2	47.0/53.8	44.0/51.2
Gemma-1B	Llama-3B	54.0/58.7	52.8/58.1	**57.5/62.9**	52.0/58.3
Llama-3B	Qwen-0.6B	**53.0/59.0**	48.9/57.3	45.7/53.9	45.3/53.2
Llama-3B	Gemma-1B	48.3/**55.7**	**48.5**/55.6	47.0/54.8	46.3/53.5
Llama-3B	Llama-3B	50.4/56.8	52.1/58.3	**58.6/63.4**	51.7/58.2

Table 19: ROPES ablation latency in milliseconds per example. Entries are translator/total; lower is better and cell-wise minima are bold. Mean across the nine model pairs: Full 5.7/141.3; −RP 4.8/144.6; −RX 3.7/141.1; LCFP 37.4/177.9.

Sharer	Receiver	Full	−RP	−RX	LCFP
Qwen-0.6B	Qwen-0.6B	6/**139**	5/**139**	**4**/140	34/171
Qwen-0.6B	Gemma-1B	5/**160**	4/163	**3**/167	33/194
Qwen-0.6B	Llama-3B	6/119	5/118	**4/116**	34/147
Gemma-1B	Qwen-0.6B	6/159	6/153	**4/147**	39/186
Gemma-1B	Gemma-1B	5/171	4/**169**	**3/169**	38/210
Gemma-1B	Llama-3B	6/**126**	5/127	**4**/127	40/167
Llama-3B	Qwen-0.6B	6/**134**	5/138	**4/134**	40/175
Llama-3B	Gemma-1B	5/**153**	4/174	**3**/158	39/200
Llama-3B	Llama-3B	6/**111**	5/120	**4**/112	40/151

Table 20: StrategyQA accuracy scores (%). Higher is better; cell-wise best values are bold. Mean across the nine model pairs: Full 65.01; −RP 64.86; −RX 64.93; LCFP 65.02.

Sharer	Receiver	Full	−RP	−RX	LCFP
Qwen-0.6B	Qwen-0.6B	61.4	**63.2**	61.7	**63.2**
Qwen-0.6B	Gemma-1B	58.2	58.2	57.3	**59.1**
Qwen-0.6B	Llama-3B	73.6	72.1	**73.8**	72.6
Gemma-1B	Qwen-0.6B	62.0	63.2	61.7	**64.4**
Gemma-1B	Gemma-1B	58.8	**59.9**	57.9	57.9
Gemma-1B	Llama-3B	74.1	73.0	**77.1**	73.6
Llama-3B	Qwen-0.6B	**65.8**	62.6	61.7	62.6
Llama-3B	Gemma-1B	57.3	58.5	58.5	**59.1**
Llama-3B	Llama-3B	73.9	73.0	**74.7**	72.7

Table 21: StrategyQA ablation latency in milliseconds per example. Entries are translator/total; lower is better and cell-wise minima are bold. Mean across the nine model pairs: Full 6.1/134.6; −RP 4.8/121.2; −RX 3.7/121.7; LCFP 42.4/201.9.

Sharer	Receiver	Full	−RP	−RX	LCFP
Qwen-0.6B	Qwen-0.6B	10/122	5/**118**	**4**/122	78/564
Qwen-0.6B	Gemma-1B	5/140	4/140	**3/139**	32/169
Qwen-0.6B	Llama-3B	6/109	5/**103**	**4**/104	34/134
Gemma-1B	Qwen-0.6B	6/126	5/**123**	**4**/128	38/161
Gemma-1B	Gemma-1B	5/**146**	4/148	**3**/149	36/179
Gemma-1B	Llama-3B	6/112	6/116	**4/108**	43/152
Llama-3B	Qwen-0.6B	6/217	5/**111**	**4**/114	42/150
Llama-3B	Gemma-1B	5/**134**	4/135	**3**/136	37/167
Llama-3B	Llama-3B	6/105	5/97	**4/95**	42/141

Table 22: Overall score comparison on five datasets (45 model-pair cells). The native score is F1 for generative QA and accuracy for classification. Tied best values at the displayed precision are counted for each tied method.

Method	Macro native score	Best datasets	Best/tied-best cells	Avg. rank
LCF-X	49.76	0/5	4/45	2.29
XKV	52.37	4/5	31/45	1.37
T2T	48.86	1/5	11/45	2.34

Table 23: Overall efficiency on five datasets (45 model-pair cells). Communication is fusor-only latency for LCF-X and translator-only latency for XKV; it is not defined for T2T. XKV is 26.4% faster end-to-end than LCF-X, 6.8× faster than T2T, and uses 76.1% fewer trainable parameters than LCF-X.

Method	Communication (ms)	E2E (ms)	E2E relative to XKV	Trainable params
LCF-X	59.9	227.9	1.36×	19.03M
XKV	5.8	167.6	1.00×	4.55M
T2T	–	1139.5	6.80×	0

Table 24: Per-dataset XKV improvements recomputed from the displayed detailed entries. Score deltas use F1 for generative QA and accuracy for classification; positive values favor XKV. Speedups are ratios of baseline E2E latency to XKV E2E latency.

Dataset	Δ vs. LCF-X	Δ vs. T2T	Speedup vs. LCF-X	Speedup vs. T2T
ROPES	+4.20	+7.68	1.24×	8.90×
MuSiQue	+1.12	+2.67	1.59×	5.09×
QASC	+3.22	-0.27	1.35×	5.45×
StrategyQA	+3.84	+1.10	1.26×	9.99×
HotpotQA-bridge	+0.66	+6.41	1.25×	6.01×

Table 25: Ablation summary over 27 fully matched cells on QASC, ROPES, and StrategyQA, recomputed from the displayed detailed entries. Tied best values are counted for each tied variant.

Variant	Δ score	Avg. rank	Best/tied-best	Component (ms)	E2E (ms)	Params
Full	+0.00	1.87	13/27	5.8	131.7	4.55M
−RP	-0.46	2.35	5/27	4.7	128.7	3.58M
−RX	-0.83	3.02	5/27	3.7	127.1	3.43M
LCFP	-0.94	2.76	5/27	39.1	183.7	3.53M
−RP vs. Full	-0.46	–	–	-18.5%	-2.3%	-21.2%
−RX vs. Full	-0.83	–	–	-36.9%	-3.5%	-24.5%
LCFP vs. Full	-0.94	–	–	+572.6%	+39.4%	-22.3%