Quantized Cache-to-Cache: Communication-Budgeted KV Transfer for Heterogeneous LLMs

Anonymous Authors

Abstract

We study communication-efficient transfer between heterogeneous large language models (LLMs) by quantizing Cache-to-Cache (C2C) KV-cache transfer. Our goal is to reduce bandwidth and memory while preserving accuracy. We present post-training quantization (INT8/INT4), cache-length reduction, and accuracy-versus-bytes curves for a heterogeneous model pair. Empirically, quantization is nearly lossless, while cache-length pruning reveals a strong front/back asymmetry that is critical for budgeted transfer. We release a reproducible evaluation pipeline and analysis scripts, and we outline a main-conference path toward sparse, projector-aware token selection and mixed precision.

1 Introduction

Large language models (LLMs) often communicate via text, which is slow and lossy. Cache-to-Cache (C2C) communicates via KV-cache projection and fusion, but does not address precision or bandwidth constraints. We ask: *How low can KV precision go before accuracy collapses, and can we recover performance under tight communication budgets?*

Contributions.

- We introduce a precision-aware C2C evaluation pipeline and quantify INT8/INT4 PTQ effects on C2C accuracy.
- We study cache-length reduction as a second budget axis and show that back-pruning consistently outperforms front-pruning.
- We report accuracy vs. communication-budget curves that jointly compare precision and cache length.
- We provide a reproducible benchmarking setup and analysis scripts to support extensions to QAT, mixed precision, heterogeneity, and selective transfer.

2 Background and Motivation

C2C projects sharer KV caches into receiver space and fuses them with learned gates, preserving rich semantics compared to text relay. However, KV caches are large: they scale with sequence length, KV heads, and head dimension. Quantization and cache-length reduction can shrink the communication footprint while retaining accuracy. This work reframes C2C through a communication-budget lens.

3 Related Work

C2C. Cache-to-Cache (C2C) enables direct semantic communication by projecting and fusing a sharer model's KV cache into a receiver's KV cache with learnable gates, avoiding intermediate text generation [1].

KV communication across agents. KVComm aligns KV caches across diverging prefixes using training-free offset correction with online anchors [2]. Q-KVComm adds adaptive layer-wise quantization, hybrid information extraction, and heterogeneous calibration for compressed KV transfer [3]. These works focus on multi-agent cache reuse/compression; our work studies quantization and cache-length pruning within the C2C projector+fuser pipeline.

Latent collaboration and cache alignment. KV cache alignment learns a shared latent space with adapters to align KV caches across models [4]. LatentMAS enables latent-space collaboration with shared working memory

without extra training [5]. Our approach stays within C2C's KV fusion but emphasizes communication budgets and precision/length tradeoffs.

Token selection and KV compression. Token-level KV selection and value-norm importance improve long-context inference for a single model (ZipCache, TokenSelect, VATP) [6, 7, 8]. We adopt the budget perspective for C2C rather than single-model KV compression.

4 Method

4.1 C2C Recap

Let the sharer model produce KV caches (K_ℓ^S, V_ℓ^S) and the receiver produce (K_ℓ^R, V_ℓ^R) at layer ℓ. C2C projects sharer KV into receiver space via Π_ℓ^K, Π_ℓ^V and fuses them through a learnable gate:

$$(K_\ell^{R\prime}, V_\ell^{R\prime}) = \mathcal{F}_\ell\left(K_\ell^R, V_\ell^R, \Pi_\ell^K(K_\ell^S), \Pi_\ell^V(V_\ell^S)\right).$$

This avoids intermediate text and transfers richer internal semantics.

4.2 Post-Training Quantization (PTQ)

We quantize the KV caches using INT8 or INT4/NF4 with per-head scaling. We evaluate accuracy and latency under fixed precision budgets. Our current implementation uses fake-quant (quantize then dequantize) to model quantization noise without bit-packing.

4.3 Cache-Length Reduction

We prune KV tokens using a fixed ratio (e.g., 50%, 25%, 10%), reducing transmitted bytes further. We evaluate front-pruning and back-pruning to diagnose which instruction tokens are most valuable for cross-model transfer.

4.4 Selective and Compressed Cache Transfer (SparseC2C)

As a main-conference extension, we select a sparse subset of token positions to transfer and fuse. Let $I \subset \{1, \ldots, T\}$ be selected tokens and S_I the gather operator. We fuse only selected tokens and scatter updates back:

$$(\tilde{K}_\ell^R, \tilde{V}_\ell^R) = S_I^\top(K_\ell^R, V_\ell^R), \quad (\tilde{K}_\ell^S, \tilde{V}_\ell^S) = S_I^\top(K_\ell^S, V_\ell^S)$$
$$(\tilde{K}_\ell^{R\prime}, \tilde{V}_\ell^{R\prime}) = \mathcal{F}_\ell\big(\tilde{K}_\ell^R, \tilde{V}_\ell^R, \Pi_\ell^K(\tilde{K}_\ell^S), \Pi_\ell^V(\tilde{V}_\ell^S)\big).$$

We then scatter the update to the full cache. We use projector-aware token scoring by computing value norms in receiver space ("proj_vnorm_topk"), tying selection to the cross-model mapping.

4.5 Communication-Budget Curves

We report accuracy as a function of transmitted bytes, enabling fair comparison under equal communication constraints. For a sequence of length T, the approximate bytes are

$$\text{bytes} \approx T \cdot p \cdot 2 \cdot L \cdot H_{kv} \cdot d_h \cdot b/8,$$

where p is the retained cache proportion, L the number of layers, H_{kv} KV heads, d_h head dim, and b bits per element. We use this accounting for consistent budget curves.

5 Experiments

5.1 Setup

We evaluate on OpenBookQA and ARC-C with a Qwen3-0.6B receiver and Qwen2.5-0.5B sharer. We follow the C2C eval protocol: temperature 0, max_new_tokens 64, no CoT, unified chat template. All models are frozen; only the projector is trained when QAT is enabled. The OpenBookQA test split has 500 samples and ARC-C has 1150 samples.

5.2 Main Results

All results below are full runs. PTQ is effectively lossless relative to FP16, and cache pruning shows a strong front/back asymmetry.

Table 1: Baseline vs. PTQ (full-cache, %).

Setting	OpenBookQA	ARC-C
FP16 baseline	52.8	55.1
INT8 PTQ	52.8	55.0
INT4 PTQ	52.6	55.4

Table 2: OpenBookQA accuracy (%, 500 samples) for cache-length pruning (INT8).

Order mode	75%	50%	25%	10%
Front	44.6	43.0	38.8	38.6
Back	52.2	52.0	50.8	49.2

Table 3: ARC-C accuracy (%, 1150 samples) for cache-length pruning (INT8).

Order mode	75%	50%	25%	10%
Front	40.2	46.3	38.3	40.7
Back	55.7	57.2	56.2	53.7

5.3 Communication-Budget Curve

Figure 1 and Figure 2 report accuracy versus effective transmitted bytes. We plot the communication budget on a log scale when the dynamic range is large; each point is annotated with the retained cache proportion. These curves provide a single, comparable view across precision (FP16/INT8/INT4) and cache-length reduction.

5.4 Order-Mode Ablation

Across all cache lengths, **back-pruning** (keeping later instruction tokens) consistently outperforms **front-pruning**. At 50% cache length, for example, back-pruning retains near-baseline accuracy while front-pruning degrades sharply. This suggests late instruction tokens carry higher utility for cross-model KV fusion, a useful design signal for future selective transfer methods.

5.5 Main-Conference Extensions (Preliminary)

We report early results for main-conference extensions. Mixed precision (INT8 with FP16 in the last layers) remains near baseline. Projector-only QAT (INT8) currently degrades accuracy (39.6/40.2), indicating that longer training or recipe tuning is needed. An alignment-only ablation (same model pair, alignment enabled) reduces accuracy, suggesting alignment should be reserved for heterogeneous pairs. For a heterogeneous pair (Qwen3→Llama3.2), alignment-on yields 44.2/47.8; alignment-off was unstable and is omitted. For SparseC2C (token selection), vnorm/knorm scoring preserves accuracy under aggressive token budgets. At p=0.5, INT8 vnorm achieves 52.4/56.2 (OpenBookQA/ARC-C) while front pruning drops to 44.8/47.6. Full grids are reported in the main-conference track.

Receiver-aware delta selection (M9). Let $(\widehat{K}^{\ell}, \widehat{V}^{\ell}) = P_{\ell}(K^{S,m}, V^{S,m}; K^{R,\ell}, V^{R,\ell})$ be the projected cache in receiver space (using the same quantize→dequantize path as transfer). We score each token by its marginal update:

$$u^{\ell}(t) = \mathbb{E}_{b,h}\left[\left\|\widehat{V}^{\ell}_{b,h,t} - V^{R,\ell}_{b,h,t}\right\|_2\right], \qquad I_{\ell} = \mathrm{TopK}\big(u^{\ell}(t), \lfloor pT \rfloor\big).$$

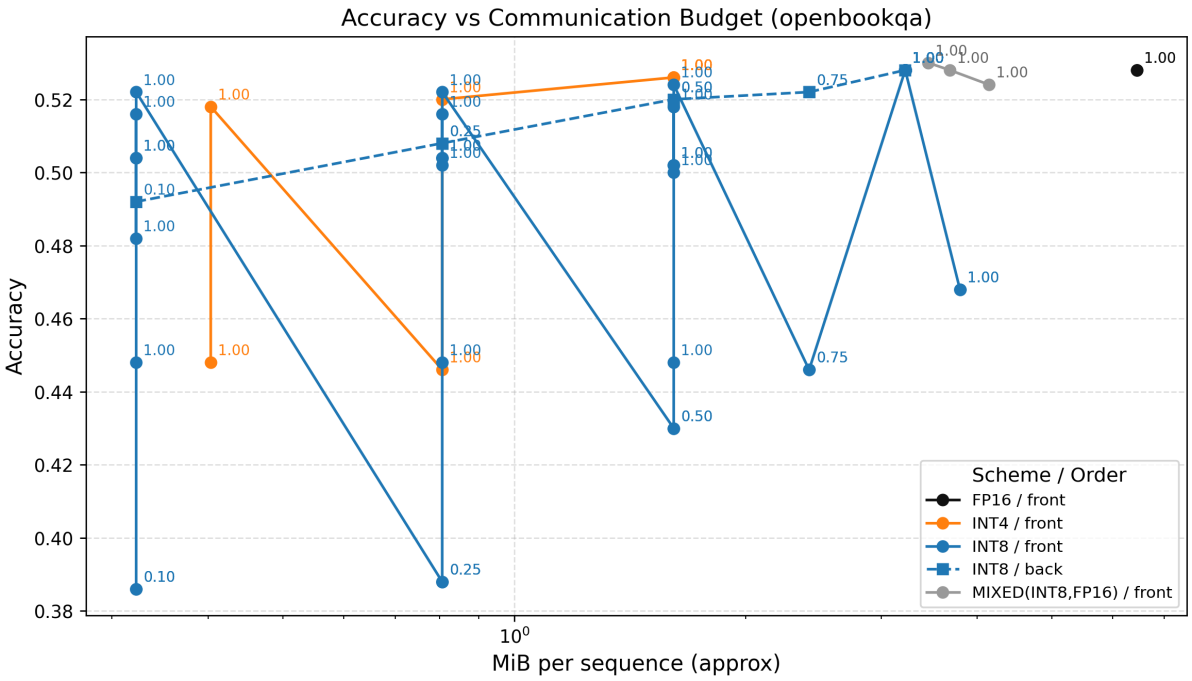

Figure 1: Accuracy vs. communication budget (OpenBookQA).

Table 4: Preliminary extension results (% accuracy).

Setting	OpenBookQA	ARC-C
Mixed precision (INT8 + last-4 FP16)	52.8	55.0
QAT (projector-only, INT8)	39.6	40.2
Alignment ablation (same pair)	46.8	49.6
Hetero pair (Qwen3→Llama3.2, align on)	44.2	47.8

Delta selection improves low-budget accuracy (e.g., +4.2/+3.3 points at $p = 0.10$ vs vnorm on OpenBookQA/ARC-C).

RD-C2C (M10). Under a byte budget R, assign each token $a_t \in \{\text{drop}, \text{int4}, \text{int8}\}$ with cost $r(a_t)$ and distortion $D_t(a_t)$:

$$\min_{\{a_t\}} \sum_{t=1}^{T} D_t(a_t) \quad \text{s.t.} \quad \sum_{t=1}^{T} r(a_t) \le R,$$

with $D_t(\text{drop}) = \|\widehat{V}_t^{\ell} - V_t^{R,\ell}\|_2^2$ and $D_t(\text{intb}) = \|\widehat{V}_t^{\ell} - \widehat{V}_t^{\ell,(b)}\|_2^2$. We use a deterministic greedy allocator (int8→int4→drop) by $u^{\ell}(t)$.

6 Discussion

Quantized C2C provides large bandwidth reductions with limited accuracy drop. Receiver-aware delta selection and RD-C2C improve low-budget performance, suggesting redundancy-aware selection is a key lever for cache transfer. A main-conference path includes QAT recovery, mixed-precision schedules, heterogeneous model pairs, and system-level latency measurements.

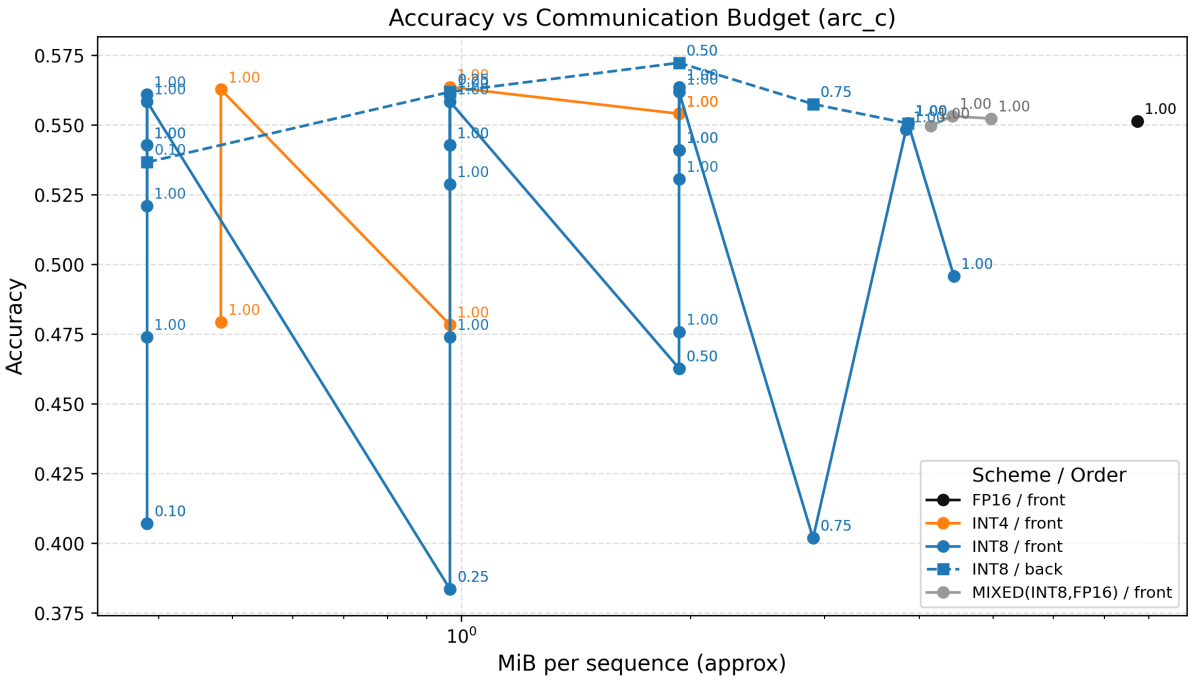

Figure 2: Accuracy vs. communication budget (ARC-C).

7 Limitations

Our results currently focus on a single model pair and two datasets. We do not yet report end-to-end latency or FLOP measurements for the fuser, and heterogeneity results for M9/M10 are pending. These limitations will be addressed in the main-conference track.

8 Broader Impacts

Communication-efficient multi-LLM systems can reduce compute and latency, but they may also enable higher-throughput deployment of models. We emphasize reproducible evaluation, careful reporting of accuracy/latency trade-offs, and responsible deployment in sensitive domains.

9 Conclusion

We introduce precision-aware C2C and report accuracy vs. bytes curves. This establishes a communication-budget perspective for cross-model KV transfer and opens the door to low-latency, low-bandwidth agent collaboration.

Acknowledgments

Placeholder.

References

[1] T. Fu et al. Cache-to-Cache: Direct Semantic Communication Between Large Language Models. Preprint, 2025. https://arxiv.org/abs/2510.03215.

Table 5: M9 delta selection vs. baselines (accuracy).

Setting	OpenBookQA	ARC-C
vnorm_topk (p=0.10)	0.422	0.478
proj_vnorm_topk (p=0.10)	0.432	0.475
delta_proj_vnorm_topk (p=0.10)	0.464	0.511
vnorm_topk (p=0.25)	0.470	0.496
proj_vnorm_topk (p=0.25)	0.462	0.526
delta_proj_vnorm_topk (p=0.25)	0.498	0.548
vnorm_topk (p=0.50)	0.508	0.560
proj_vnorm_topk (p=0.50)	0.504	0.546
delta_proj_vnorm_topk (p=0.50)	0.540	0.573

Table 6: M10 RD budgets (accuracy).

Setting	OpenBookQA	ARC-C
RD budget 0p03125	0.498	0.548
RD budget 0p0625	0.534	0.570
RD budget 0p125	0.524	0.549
RD budget 0p25	0.528	0.550

[2] H. Ye et al. KVCOMM: Online Cross-context KV-cache Communication for Efficient LLM-based Multi-agent Systems. Preprint, 2025. https://arxiv.org/abs/2510.03346.

[3] B. Kriuk and L. Ng. Q-KVComm: Efficient Multi-Agent Communication via Adaptive KV Cache Compression. Preprint, 2025. https://arxiv.org/abs/2512.17914.

[4] L. M. Dery et al. Latent Space Communication via K-V Cache Alignment. Preprint, 2026. https://arxiv.org/abs/2601.06123.

[5] J. Zou et al. Latent Collaboration in Multi-Agent Systems. Preprint, 2024. https://arxiv.org/abs/2405.16103.

[6] Y. Liu et al. ZipCache: Accurate and Efficient KV Cache Compression for LLMs. Preprint, 2024. https://arxiv.org/abs/2405.14256.

[7] TokenSelect: Efficient Long-Context Inference via Token Selection. EMNLP, 2025. https://aclanthology.org/2025.emnlp-main.1079/.

[8] J. Wang et al. Attention Score is not All You Need for Token Importance Indicator in KV Cache Reduction: Value Also Matters. Preprint, 2024. https://arxiv.org/abs/2406.12335.