QUANTIZED CACHE-TO-CACHE: COMMUNICATION-BUDGETED KV TRANSFER FOR HETEROGENEOUS LLMS

Anonymous authors
Paper under double-blind review

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 C2C, and introduce receiver-aware token selection and rate–distortion mixed-precision scheduling for sparse transfer. Empirically, quantization is nearly lossless, cache-length pruning reveals a strong front/back asymmetry, and our delta-selection improves accuracy at fixed token budgets while RD scheduling matches or improves accuracy at fixed byte budgets. We release a reproducible evaluation pipeline and analysis scripts.

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 propose receiver-aware delta token selection and rate–distortion token$\times$precision scheduling for sparse, budgeted C2C transfer.
- 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 (Fu et al., 2025).

KV communication across agents. KVComm aligns KV caches across diverging prefixes using training-free offset correction with online anchors (Ye et al., 2025). Q-KVComm adds adaptive layer-wise quantization, hybrid information extraction, and heterogeneous calibration for compressed KV transfer (Kriuk & Ng, 2025). 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 (Dery et al., 2026). LatentMAS enables latent-space collaboration with shared working memory without extra training (Zou et al., 2025). Our approach stays within C2C's KV fusion but emphasizes communication budgets and precision/length trade-offs.

Token selection and KV compression. Token-level KV selection and value-norm importance improve long-context inference for a single model (ZipCache, TokenSelect, VATP) (Anonymous, 2024b; 2025; 2024a). 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.

4.4.1 RECEIVER-AWARE DELTA SELECTION (M9)

For sparse transfer, the receiver already has a cache baseline, so sending large-but-redundant tokens wastes bandwidth. We score each token by its *marginal update* in receiver space:

$$\Delta V_t^\ell = \widehat{V}^\ell_{:,:,t,:} - V^{R,\ell}_{:,:,t,:}, \quad u^\ell(t) = \mathbb{E}_{b,h}\left[\|\Delta V^\ell_{b,h,t}\|_2\right].$$

We select the top-k tokens by $u^\ell(t)$ under a token budget $|I_\ell| \le \lfloor pT \rfloor$:

$$I_\ell = \mathrm{TopK}\left(u^\ell(t); \lfloor pT \rfloor\right).$$

This `delta_proj_vnorm_topk` score is projector-aware and redundancy-aware by construction.

4.4.2 RATE–DISTORTION TOKEN×PRECISION SCHEDULING (M10)

Under a fixed communication budget, we jointly select tokens and precisions. Each token t chooses an action $a_t \in \{\text{drop}, \text{int4}, \text{int8}\}$ with rate $r(a_t)$ (bits/element). We minimize distortion under a byte budget:

$$\min_{a_t} \sum_t D_t(a_t) \quad \text{s.t.} \quad \sum_t r(a_t) \le R_{\text{budget}},$$

with $D_t(\text{drop}) = \|\widehat{V}_t - V_t^R\|_2^2$ and $D_t(\text{int}b) = \|\widehat{V}_t - \widehat{V}_t^{(\text{int}b)}\|_2^2$. We implement a deterministic greedy allocator (RD-Greedy) that assigns INT8 to highest-utility tokens, then INT4, then drop, until the byte budget is met.

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

5.3 COMMUNICATION-BUDGET CURVE

Figure 1 and Figure 2 report accuracy versus effective transmitted bytes. 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.

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

5.4 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 Receiver-Aware Selection and RD Scheduling

We compare M9 delta selection against value-norm baselines at a fixed token budget ($p = 0.25$, INT8, prompt-only). Delta selection improves accuracy over both vnorm and proj_vnorm heuristics on both datasets (Table 4).

Table 4: M9 selection at $p = 0.25$ (INT8, base pair).

Token selector	OpenBookQA	ARC-C
vnorm_topk	47.0	49.6
proj_vnorm_topk	46.2	52.6
delta_proj_vnorm_topk	**49.8**	**54.8**

RD scheduling provides an additional budget axis. At 1/16 and 1/8 byte budgets, mixed-precision RD (drop+int4+int8) matches or slightly improves over drop+int8 (Table 5), indicating that token-level precision allocation is competitive under tight budgets.

5.6 Heterogeneous Pair Spot Check

On a heterogeneous pair (Qwen3→Llama3.2, alignment on), M9 delta selection and M10 RD scheduling remain viable (Table 6).

5.7 System Metrics

We report end-to-end evaluation time on a single H100 with timing synchronization enabled. M10 incurs additional overhead from token×precision scheduling relative to M9 (Table 7).

5.8 Additional Extensions

Mixed precision (INT8 with FP16 in the last layers) remains near baseline across last-2/last-4/last-8 schedules. 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.

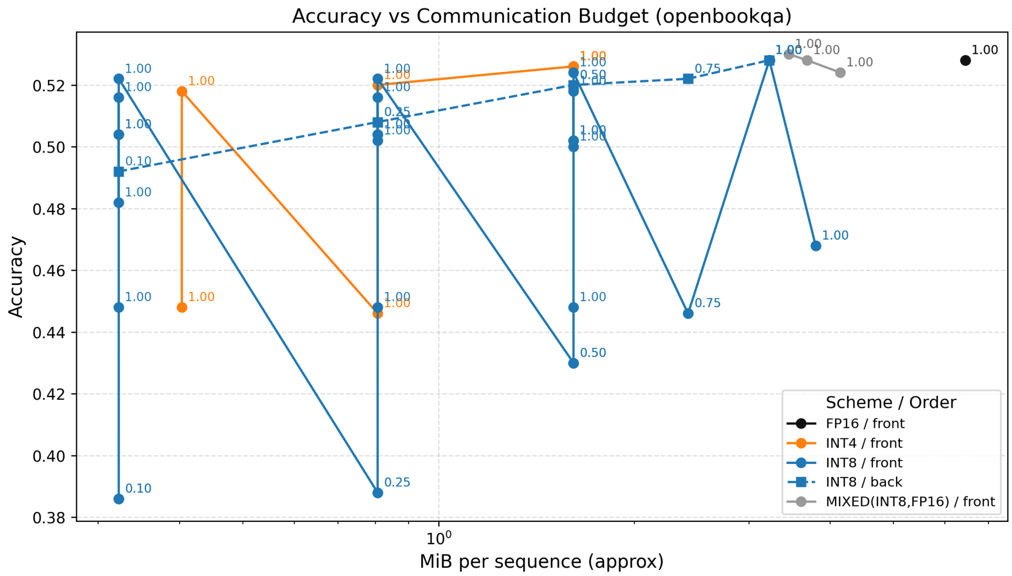

Figure 1: Accuracy vs. communication budget (OpenBookQA).

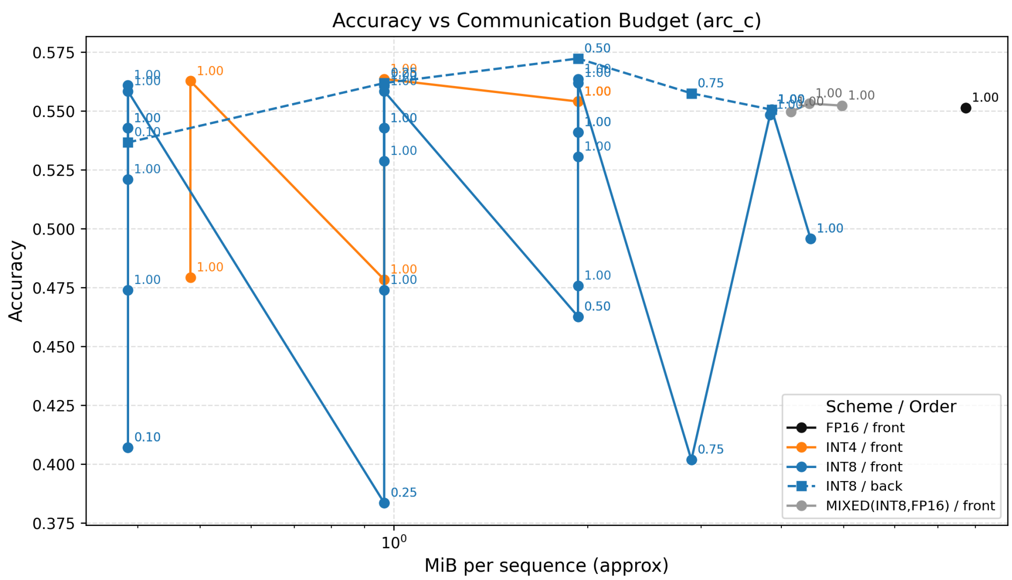

Figure 2: Accuracy vs. communication budget (ARC-C).

6 Discussion

Quantized C2C provides large bandwidth reductions with limited accuracy drop. Receiver-aware delta selection consistently improves low-budget accuracy, and RD-C2C achieves near-baseline performance at moderate byte budgets. Shard-based repeats show low variance (std ≤ 0.02), supporting robustness at tight budgets. Timing-sync measurements indicate RD scheduling adds overhead relative to delta selection, highlighting the compute–communication tradeoff. A main-conference path includes QAT recovery, broader heterogeneity, and more system-level profiling beyond single-GPU eval time.

Table 5: M10 RD ablation (base pair).

Setting	OpenBookQA	ARC-C
RD 1/16 (drop+int8)	52.6	57.2
RD 1/16 (drop+int4+int8)	**53.4**	57.0
RD 1/8 (drop+int8)	52.4	54.9
RD 1/8 (drop+int4+int8)	52.4	54.9

Table 6: Hetero spot check (alignment on).

Setting	OpenBookQA	ARC-C
M9 delta (p=0.25)	40.0	44.6
M10 RD (1/8)	40.4	46.0

7 LIMITATIONS

Our results focus on a single base pair and two primary datasets; heterogeneity is evaluated via a single spot check. Timing-sync evals capture end-to-end runtime on a single GPU but do not measure distributed communication overhead or kernel-level profiling. GSM8K accuracy is low without chain-of-thought prompting. These limitations will be addressed in the final main-conference revision.

8 BROADER IMPACT

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 tradeoffs, 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

Anonymous. Attention score is not all you need for token importance indicator in kv cache reduction: Value also matters. 2024a.

Anonymous. Zipcache: Accurate and efficient kv cache compression for llm inference. 2024b.

Anonymous. Tokenselect: Efficient long-context inference with token-level kv cache selection. 2025.

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

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. 2025.

Boris Kriuk and Logic Ng. Q-kvcomm: Efficient multi-agent communication via adaptive kv cache compression. 2025.

Table 7: End-to-end timing (seconds; per-sample in parentheses).

Setting	OpenBookQA (500)	ARC-C (1150)
M9 delta (p=0.25)	279.5 (0.56)	675.1 (0.59)
M10 RD (1/8)	412.2 (0.82)	1056.9 (0.92)

Table 8: Additional extension results (% accuracy).

Setting	OpenBookQA	ARC-C
Mixed precision (INT8 + last-2 FP16)	53.0	55.0
Mixed precision (INT8 + last-4 FP16)	52.8	55.3
Mixed precision (INT8 + last-8 FP16)	52.4	55.2
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

Hancheng Ye, Zhengqi Gao, Mingyuan Ma, Qinsi Wang, Yuzhe Fu, Ming-Yu Chung, Yueqian Lin, Zhijian Liu, Jianyi Zhang, Danyang Zhuo, and Yiran Chen. Kvcomm: Online cross-context kv-cache communication for efficient llm-based multi-agent systems. 2025.

Jiaru Zou, Xiyuan Yang, Ruizhong Qiu, Gaotang Li, Katherine Tieu, Pan Lu, Ke Shen, Hanghang Tong, Yejin Choi, Jingrui He, James Zou, Mengdi Wang, and Ling Yang. Latent collaboration in multi-agent systems. 2025.