CACHEWRITER: LEARNING TO WRITE LATENT MESSAGES FOR FROZEN LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language model (LLM) agents often hold private information, so a shared task requires one agent to pass what it knows to another. Text is the usual channel, but it costs the sender a decode and the receiver a re-encode, and the message must fit into the receiver's context. Transmitting key–value (KV) cache entries avoids the decode. However, existing cache handoff only *selects* among states that the sender computed for its own continuation, and under a tight position budget, selection leaves most of the available gain unrecovered. In this work, we propose CACHEWRITER, in which a question-blind sender *writes* a fixed number of KV entries for a frozen receiver. Specifically, each entry is a selected state plus a zero-initialised low-rank update, so that training starts exactly at selection. The update learns to match the receiver's output distribution given the full private context, without any answer labels. Extensive experiments across four model families and four single-document question-answering benchmarks demonstrate that 32-entry messages improve validation exact match by 5.3 percentage points on average over the strongest position-matched baseline. On answer-exclusive, randomly ordered five-document HotpotQA, 64 written entries more than double the best selection policy for Qwen3-8B (19.6% vs. 9.4% held-out exact match). Replacing the learned update with its document-independent mean further shows that the instance-specific part of the gain grows as the budget binds. Across model families, one fixed calibration per pair beats a position-matched text note by 8.6 points (Llama-3.1→Qwen3) and 14.6 points (Qwen3→Falcon3) of held-out exact match, and on the first pair it is statistically indistinguishable from same-family writing.

1 INTRODUCTION

Large language model (LLM) systems increasingly divide work across multiple agents, e.g., a retriever hands evidence to a reasoner, or a tool-use agent reports to a planner (Du et al., 2024; Wu et al., 2024; Hong et al., 2024). Text is the default medium for such handoff. Although robust, it costs the sender a decode and the receiver a re-encode, and a short note must also fit into a receiver-position budget. To avoid the decode, recent protocols transmit activations instead of text (Shi et al., 2026; Fu et al., 2026; Ramesh and Li, 2025; Zou et al., 2026). When two models share an architecture, the receiver can read a transmitted key–value (KV) cache as a latent message. This interface removes sender decoding and shrinks the receiver's position budget. For instance, a 32-entry message occupies 4.6 MB on average, which PCIe 4 carries in 0.15 ms (§4.6). We therefore target a binding receiver-position budget between co-located agents or across a fast interconnect.

Despite this promise, existing cache-handoff protocols share a restrictive design assumption: *a message must be selected from the states that the sender has already produced.* KVComm (Shi et al., 2026) selects layers, while eviction methods select positions (Xiao et al., 2024; Zhang et al., 2023; Li et al., 2024; Kim et al., 2025). Selection retains entries the receiver reads well and leaves them unrevised for the receiver's use. Under a tight position budget for Qwen3-8B on HotpotQA, the strongest of eleven deployable selection policies recovers 34% of the gain available from the full private context, and at a byte-matched budget KVComm recovers 15% of its corresponding gap. Concurrent work synthesises cache states instead of selecting them (Rossi et al., 2026; O'Neill et al., 2026; Liu et al., 2026), but none starts from an exact selection control or tests whether its gain depends on the transmitted instance (Table 24).

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.

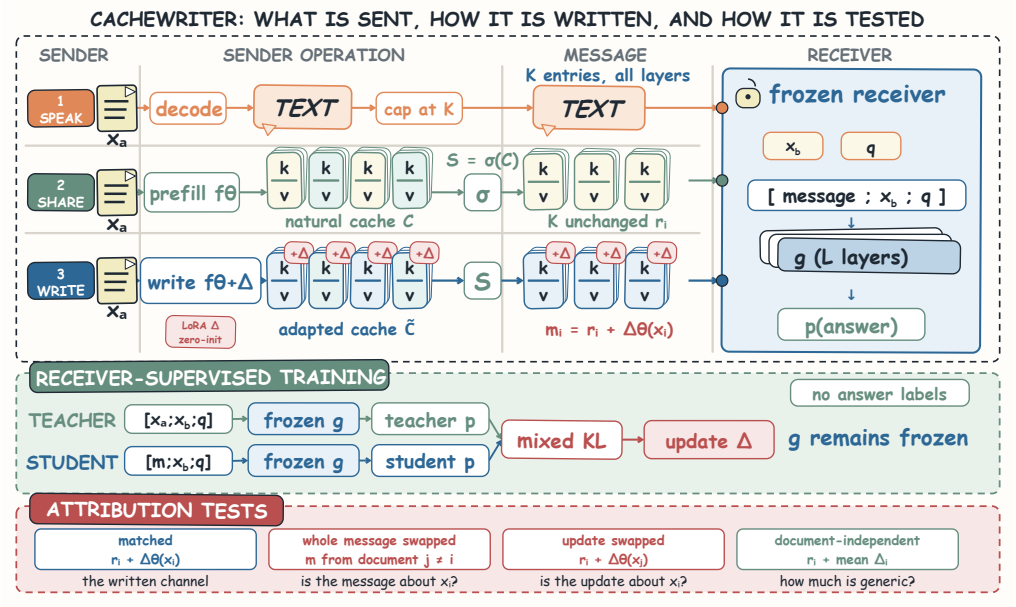


Figure 1: **CACHEWRITER writes a fixed K-entry cache message for a frozen receiver g.** Speaking sends a text note, sharing sends K unchanged entries r_i of the sender's cache, and CACHEWRITER writes $m_i = r_i + \Delta_\theta(x_i)$ at the same positions. Attribution tests: §4.4.

Moving from selecting cache states to writing them raises three challenges. (1) *Supervision*. The sender is question-blind and no gold answer is available, so it is unclear what a compact message should be trained to reproduce. (2) *Readability*. A message written outside the receiver's activation range degrades the receiver instead of informing it (Appendix H). (3) *Attribution*. A useful channel must convey content that depends on the sender's current document. Accuracy alone cannot test this, since a generic prefix can also improve the receiver. Concurrent audits find deployed systems in which removing a relayed cache costs many points, while replacing it with another example's cache costs almost nothing (Cheng et al., 2026; Zhang and Emu, 2026).

To address these challenges, we propose CACHEWRITER, a framework that writes a fixed number of cache entries for a frozen receiver. As shown in Figure 1, CACHEWRITER consists of three components. (1) *Receiver-supervised writing*. The frozen receiver's behaviour given the full private context defines the distribution that a compact message should reproduce, so training reads no gold-answer labels. (2) *Selection-anchored parameterisation*. We represent each message as a readable reference plus a learned update, $m_i = r_i + \Delta_\theta(x_i)$, where a zero-initialised low-rank adapter is active only during the sender's write pass. At initialisation, the message is thus bit-for-bit identical to the selection baseline it replaces. (3) *Derangement-based attribution*. Derangements swap in another instance's whole message or only its learned update. Together with a document-independent update as the generic control, they split the gain into document-independent and instance-specific parts.

We compare writing with selection, text, soft-token, and layer-sharing baselines at matched position or byte budgets. At $3.5\times$ receiver-position compression for Qwen3-8B on HotpotQA-Private, our private-view split of HotpotQA, writing recovers 58% of the available gap, compared with 32% for its initial selection baseline. We then increase the source to five documents while holding the message at 32 or 64 entries. In this answer-exclusive setting with randomly ordered sender documents, writing recovers 62% of the gap on held-out test at 64 entries and 37% at 32, whereas the best selection policy recovers 14% at 64. To summarise, our main contributions are three-fold:

- We formulate fixed-budget latent communication as **writing cache states** rather than selecting them, supervised by a frozen receiver's full-context behaviour without gold-answer labels (§3).
- We propose **CACHEWRITER**, which writes each entry as a selected cache state plus a zero-initialised low-rank update. Selection thus becomes an exact control, and **derangement-based attribution** splits the gain into document-independent and instance-specific parts (§3, §4.4).

- Extensive experiments demonstrate that at $K{=}32$ writing leads the strongest position-matched baseline by +0.053 exact match on average across four receivers and four benchmarks, winning 14 of 16 cells, and beats its selection arm in every random-order multi-document cell (§4.2–4.3). The instance-specific part of the gain grows as the budget binds, from none at 64 entries on single documents to $+0.088$ on five documents on validation, and across model families one fixed calibration beats a position-matched note by $+0.086$ on held-out test (§4.4–4.5).

2 Related Work

Cache selection and handoff. KVComm formulates the sender–receiver cache-handoff problem that we adopt, compressing a message by selecting layers from the sender's natural cache (Shi et al., 2026). Training-free eviction methods select entries from that same cache along the position axis (Xiao et al., 2024; Zhang et al., 2023; Li et al., 2024; Liu et al., 2023; Guo et al., 2024; Kim et al., 2025), and some also average evicted values into the kept ones (Zhang et al., 2024; Wan et al., 2025). These approaches keep or average existing states, while CacheWriter changes the states before transmission and holds their number fixed. Mapped-cache methods learn projection and fusion (Fu et al., 2026) or fit a ridge map (Heo et al., 2026) when both models process the same input, and concurrent methods synthesise cache states for the same model (O'Neill et al., 2026) or for a frozen reader across models and differing contexts (Rossi et al., 2026; Liu et al., 2026). We differ in the control and the setting. Our writer is a zero-initialised update on the selected cache, so the untrained channel reproduces its selection baseline bit-for-bit, and our private-view split asks whether a message can combine with evidence and a question unavailable to its sender (Appendix H).

Learned context compression and latent interfaces. KV-Distill learns selected-state adaptation for later use by the model that produced the cache (Chari et al., 2025). Gist tokens and autoencoders likewise compress context for self-use (Mu et al., 2023; Chevalier et al., 2023; Ge et al., 2024), soft prefixes adapt a frozen model to a task (Li and Liang, 2021; Lester et al., 2021), and frozen-reader bridges such as xRAG hide the source text from a reader that holds no private evidence (Cheng et al., 2024). Other latent media include expectation-weighted embeddings, spliced activations, recycled latent thoughts, relayed latents, and continuous reasoning (Pham et al., 2024; Ramesh and Li, 2025; Zou et al., 2026; Li et al., 2026; Hao et al., 2025). We write directly in the receiver's native cache format.

3 Method

Given a private document x_a, CacheWriter produces K key–value entries that a frozen receiver combines with its own evidence x_b and question q. Training first selects a readable reference from the sender's natural cache, then uses a zero-initialised adapter to update those entries, and finally matches the receiver's behaviour when it sees the full private context. At inference the sender transmits only the entries its adapted prefill writes, and the receiver prepends them to its own sequence.

3.1 Problem setting and evaluation quantities

The private-view setting. The sender observes a private document x_a, while the receiver, a frozen decoder-only transformer g with L layers, observes a different document x_b and a question q. The sender never sees q, and the receiver sees x_a only through the message. We build instances from four multi-hop and open-domain datasets by giving the gold answer-bearing passage to the sender and the rest to the receiver, which makes x_a relevant by construction, and §4.4 audits whether x_b alone suffices. Figure 1 contrasts the three handover protocols. When the agents share weights the mechanism resembles learned context compression. The private-view split makes it communication.

Three distinct budgets. A protocol can be constrained by receiver positions, transmitted bytes, or compute. We match text comparisons on receiver positions, our primary budget, and the KVComm comparison on transmitted bytes (Appendix B.4). On HotpotQA, a single private paragraph occupies 112 positions on average and the multi-document sender of §4.3 672, against 32 for the message.

A message contains K entries, with a key and value vector at every layer, $m = \{(k_t^{(l)}, v_t^{(l)})\}_{l \le L}$, and is prepended so the receiver attends over $[\, m;\, x_b;\, q\,]$. One Qwen3-8B entry is 144 KiB in bfloat16.

A floor and a full-context reference. Every measurement has two anchors. The floor is the receiver answering from x_b and q alone, and the reference gives it the entire private context as the uncompressed cache of x_a. Where its text form scores higher, we also report retained gain against it (Table 1 and Appendix D.3). We summarise a protocol by the fraction of the available gap it recovers,

$$R \;=\; \frac{\text{score(protocol)} - \text{score(no message)}}{\text{score(full private context)} - \text{score(no message)}}, \tag{1}$$

which makes budgets comparable across datasets with different floors and references. Two paired contrasts support source-instance attribution. *Utility*, $\text{score}(m_i) - \text{score}(\varnothing)$, measures whether a message helps. The *pairing effect*, $\text{score}(m_i) - \text{score}(m_{\pi(i)})$, measures whether it helps more than a message from a deranged instance $\pi(i)$, and we compute it for the whole message and for its learned update (§4.4). Utility without a pairing effect marks a generic prefix. Communication needs both.

3.2 Writing receiver-readable cache entries

A sharing protocol retains all positions at a subset of layers (Shi et al., 2026) or K positions at every layer (Zhang et al., 2023; Li et al., 2024), so its candidates come from a pass optimised for the sender's own continuation. CacheWriter lifts that constraint and keeps an exact selection control: it leaves the receiver unchanged, trains only the sender's write pass, and writes each message as a readable reference plus a learned update. The sender and receiver begin from the same pretrained model. We apply a rank-r low-rank adaptation (Hu et al., 2022) to the query, key, value, and output projections of every sender attention block, active only during the write pass over the private document and disabled for every receiver operation, so the receiver remains byte-identical to the base model. The write operation is

$$\tilde{C} \;=\; f_{\theta+\Delta}(x_a), \qquad m \;=\; \big\{(\tilde{k}_t^{(l)}, \tilde{v}_t^{(l)}) : t \in S\big\}, \qquad S = \sigma(C),\;\; C = f_\theta(x_a),\;\; |S| = K, \tag{2}$$

where θ denotes the frozen base weights and Δ the trainable low-rank update. For instance i with document x_i, $m_i = r_i + \Delta_i$, where $\Delta_i = \Delta_\theta(x_i)$ is the learned update and r_i the unadapted entries at S. Placing the parameters inside the attention projections, the locus of learned cache compression (Chari et al., 2025), keeps the message in the receiver's native cache format and lets every layer help form it. A message written outside the receiver's activation range degrades it instead (Appendix H).

The selector σ retains position 0, the attention sink of documented outsized mass (Xiao et al., 2024), and spends the remaining $K - 1$ slots on pooled-attention scores. We compute S from the *unadapted* sender, which prevents the selector from co-adapting with the writer, and zero-initialise the low-rank update. At step zero the message is therefore bit-for-bit identical to the selection baseline.

3.3 Learning from the frozen receiver

The frozen receiver supplies a training target without a gold-answer label. Let $p^{\text{teacher}} = g([\,x_a;\, x_b;\, q\,])$ be its next-token distribution when it receives the private document directly. Here $[\,x_a;\, x_b;\, q\,]$ denotes the receiver run with the uncompressed cache of x_a as a prefix, the cache-form reference that defines R in Eq. 1, so the student is trained toward the reference it is scored against. Let $p^{\text{student}} = g([\,m;\, x_b;\, q\,])$ be the distribution induced by the written message. The teacher is the same frozen network with a more informative input, requires neither a larger model nor an annotation, and is defined for raw text without a question (§4.2).

We minimise a mixed divergence,

$$\mathcal{L} \;=\; \sum_j w_j \Big[\lambda\, \mathrm{KL}\big(p_j^{\text{teacher}} \,\|\, p_j^{\text{student}}\big) \;+\; (1-\lambda)\, \mathrm{KL}\big(p_j^{\text{student}} \,\|\, p_j^{\text{teacher}}\big)\Big]. \tag{3}$$

The objective combines mass-covering and mode-seeking divergences, as is common when distilling generative behaviour (Gu et al., 2024; Agarwal et al., 2024). The positions j come from the receiver's greedy continuation under the current message, refreshed at every step, so the student is corrected on states it actually visits (Agarwal et al., 2024). Positions inside q receive a small weight, while the final prompt position receives full weight because it determines the initial answer distribution. Appendix C.3 compares this with supervising the teacher's own trajectory.

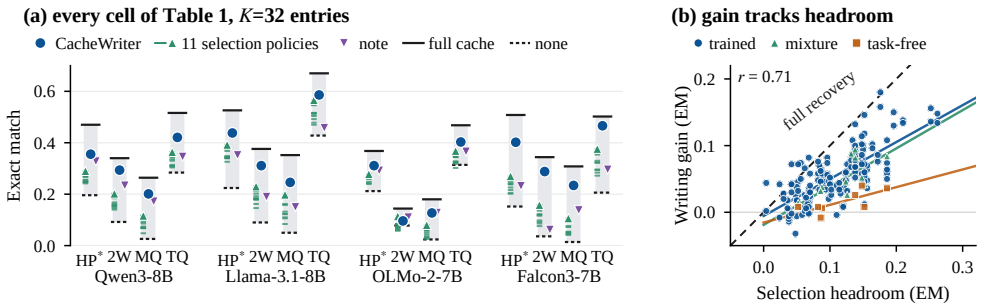


Figure 2: **Writing leads the best selection policy in 15 of** 16 **cells, and its gain tracks selection headroom. (a)** K=32, validation. Grey bars span no message to full cache. Dashes are the eleven deployable policies, the best filled. **(b)** Gain against selection headroom (Appendix G.1).

4 Experiments

4.1 Experimental design

Benchmarks and models. We build private-view instances from HotpotQA (Yang et al., 2018), 2WikiMultihopQA (Ho et al., 2020), MuSiQue (Trivedi et al., 2022), and TriviaQA (Joshi et al., 2017) with one construction. We evaluate 500 validation instances by normalised exact match and paired bootstrap 95% intervals, then confirm headline results on 500 held-out test instances. The primary pair is Qwen3-8B (Yang et al., 2025). We replicate on Llama-3.1-8B (Grattafiori et al., 2024), OLMo-2-7B (OLMo et al., 2025), and Falcon3-7B (Falcon-LLM Team, 2024) instruct variants. Sender and receiver share a base model except in §4.5. The rank-64 adapter trains for up to 2 500 steps, about 7.9 hours on one L40S (Appendices A and G.2). Table 1 pools training seeds. Other results use the primary seed unless stated otherwise.

Baselines and controls. Text, selection and soft-token baselines use the message's K receiver positions. The text baseline is the sender's note capped at K tokens, plain or budget-aware. Selection covers eleven training-free policies that keep the attention sink and $K-1$ entries of the sender's natural cache, among them StreamingLLM, H2O, SnapKV, KVzip and CaM / D2O, which also merges evicted values into the kept entries (Xiao et al., 2024; Zhang et al., 2023; Li et al., 2024; Kim et al., 2025; Zhang et al., 2024; Wan et al., 2025), plus a query-aware oracle that ranks entries with the receiver's question. CIPHER sends K expectation-weighted embeddings (Pham et al., 2024), and KVComm shares whole sender layers, at $\rho = 0.3$ about as many bytes as a 32-entry message on HotpotQA (Shi et al., 2026). Every same-family channel also carries a derangement arm, a message written from another instance's document, and a corruption arm that destroys the entries. A deranged message alone lifts Falcon3 well above its floor, so content is read from the pairing effect, the margin over the deranged arm (Appendix A).

4.2 Primary effectiveness under fixed budgets

Fixed-budget frontier. At 32 entries, writing leads the strongest position-matched baseline of any medium by $+0.053$ on average and wins 14 of 16 cells (Table 1, Δ column). The strongest baseline is a text note in 6 cells and selection in 10. On the primary cell, Qwen3-8B on HotpotQA, the improvement over the selection baseline used for initialisation is $+0.074$ $[+0.046, +0.102]$ at 32 entries and $+0.054$ $[+0.030, +0.078]$ at 64. Across five training seeds of this cell, the seed-level margin at 32 entries is $+0.072$ $[+0.044, +0.099]$, and $+0.079$ $[+0.059, +0.099]$ on held-out test. Taking each seed's last checkpoint in place of the one a 48-example validation probe selects gives $+0.070$ $[+0.053, +0.086]$ and $+0.069$ $[+0.051, +0.087]$, so the gain holds without selecting on validation (Appendix A). The budget-aware note is the stronger text arm in 19 of 32 cells, and substituting it still leaves writing ahead by $+0.048$ at 32 entries, with 13 of 16 cells positive. The compressor LLMLingua-2 (Pan et al., 2024) trails the truncated note on the primary cell, 0.280 against 0.328 (Appendix B.3). Of all 171 written-minus-selection comparisons, 126 positive and 1 negative survive Benjamini–Hochberg (Benjamini and Hochberg, 1995) at q=0.05 (Appendix G.1).

			Position-matched baselines			Cache writing			
Receiver	Benchmark	None	Note	Sel.	CIPHER	CACHEWRITER	Δ	R_{KV}	Full KV
K = 32 entries (text notes: 32 tokens; CIPHER: 32 soft tokens)									
Qwen3-8B	HotpotQA*	0.196	0.328	0.290	0.250	0.356	**+0.028**	58%	0.470
	2Wiki	0.092	0.234	0.202	0.156	0.294	**+0.060**	81%	0.340
	MuSiQue	0.026	0.172	0.116	0.102	0.201	**+0.029**	74%	0.264
	TriviaQA	0.284	0.346	0.364	0.298	0.421	**+0.057**	59%	0.516
Llama-3.1-8B	HotpotQA*	0.224	0.352	0.392	0.308	0.438	**+0.046**	71%	0.526
	2Wiki	0.090	0.190	0.230	0.142	0.311	**+0.081**	77%	0.376
	MuSiQue	0.050	0.150	0.198	0.116	0.246	**+0.048**	65%	0.352
	TriviaQA	0.428	0.458	0.566	0.414	0.586	**+0.020**	65%	0.670
OLMo-2-7B	HotpotQA*	0.212	0.292	0.278	0.242	0.311	**+0.019**	63%	0.368
	2Wiki	0.078	0.110	0.114	0.068	0.096	−0.018	27%	0.144
	MuSiQue	0.024	0.128	0.080	0.062	0.127	−0.001	66%	0.180
	TriviaQA	0.314	0.366	0.368	0.312	0.403	**+0.035**	58%	0.468
Falcon3-7B	HotpotQA*	0.152	0.232	0.270	0.170	0.402	**+0.132**	70%	0.508
	2Wiki	0.036	0.062	0.158	0.052	0.288	**+0.130**	82%	0.344
	MuSiQue	0.014	0.138	0.106	0.066	0.234	**+0.096**	75%	0.308
	TriviaQA	0.206	0.296	0.376	0.230	0.466	**+0.090**	88%	0.502
Mean		0.152	0.241	0.257	0.187	0.324	**+0.053**	—	—

Table 1: **Exact match, four receivers × four benchmarks at K=32** (n=500 validation per cell). Channels train on HotpotQA-Private only (*), so the other rows are transfer. Sel.: the best of the eleven deployable selection policies. Green: best position-matched baseline, the reference for Δ. Blue: ours. R_{KV}: fraction of the gap from no message to the full cache recovered. OLMo-2 on HotpotQA, whose text reference is higher (0.404), recovers 52% of the gap to it.

The advantage concentrates where the budget binds. At 64 entries writing still leads the strongest position-matched baseline by +0.017 on average, as the extra 32 entries lift selection by +0.089 and writing by +0.041 (Appendix B.1). Figure 2(b) shows the pattern over the full evaluation pool.

No single training-free policy dominates: content salience is the per-cell best in 9 of 16 cells and pooled attention in 4, so the *Sel.* column of Table 1 takes the per-cell best. Writing leads it by +0.067 at 32 entries and wins 15 of 16 cells (Figure 2(a), with 64 entries in Appendix B.2). The query-aware oracle, which sees the receiver's question, is level with the per-cell best deployable policy (−0.001 at 32 entries, +0.003 at 64), so even question-aware selection stays below writing.

Byte-matched layer sharing. KVComm (Shi et al., 2026) sends every position of a fraction ρ of the sender's layers. We run the authors' implementation on our instances with the same Qwen3-8B pair and compare retained gain within each harness (Table 8). At byte parity, $\rho = 0.3$ on HotpotQA, KVComm recovers 15% of its gap against 59% for writing and needs $\rho = 0.5$, 1.8× the bytes, to exceed it. On 2WikiMultihopQA and MuSiQue, $\rho = 0.3$ recovers 24% and 10% against 78% and 71% for writing, and on MuSiQue even $\rho = 0.7$ stays below writing (Appendix B.4).

Model families and benchmarks. Channels train on HotpotQA-Private only, so twelve of the sixteen cells in Table 1 measure transfer, and writing leads the strongest baseline in ten of them, the exceptions being OLMo-2-7B on 2WikiMultihopQA and MuSiQue. On HotpotQA the gain over the selection each channel starts from runs from +0.037 (OLMo-2) to +0.146 (Falcon3) on validation and stays positive for every model family on held-out test, from +0.018 to +0.141 (Table 4). CIPHER, whose K embeddings carry 16–64× fewer bytes, averages 0.187 at 32 entries against 0.257 for the best selection policy and 0.324 for writing, and falls below selection in all 32 cells over both budgets.

Task exposure. Table 9 varies the training data. Trained on HotpotQA, the channel reaches 0.294 on 2WikiMultihopQA and 0.201 on MuSiQue, against 0.270 and 0.174 when trained on each directly, and at a matched 10,000-instance training size it still leads on MuSiQue by +0.044 [+0.018, +0.070] (Appendix C.1). A channel trained only to continue raw text, with no questions, answers or labelled

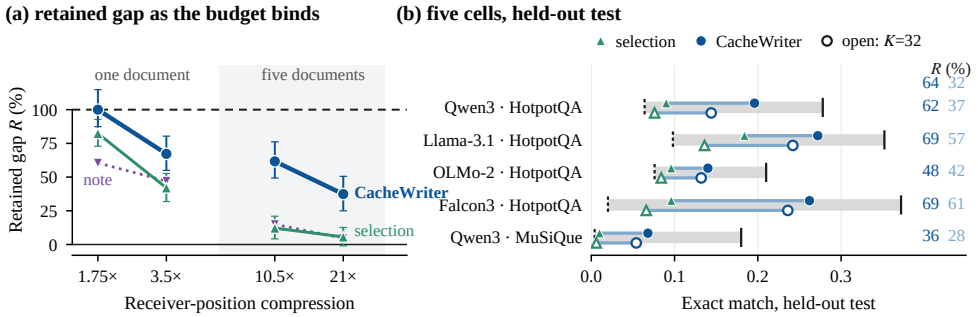


Figure 3: **Where the budget binds, writing retains at least twice the gap selection retains.** Held-out test. **(a)** Qwen3-8B on HotpotQA with one document and with five in random order, paired bootstrap 95% intervals. **(b)** The five random-order cells of Table 15. Grey bars span no message (dotted) to full cache (solid).

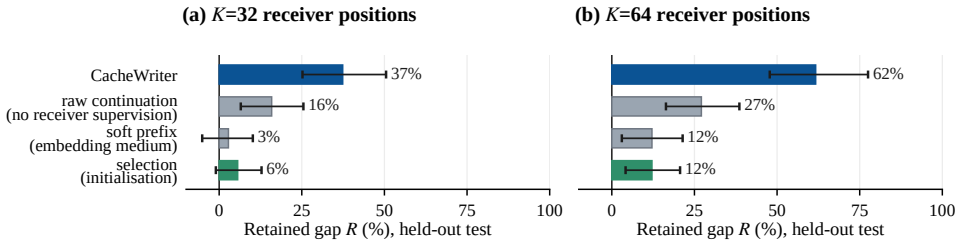


Figure 4: **Receiver supervision and the key–value medium carry the gain.** Qwen3-8B on HotpotQA, random order, held-out test, paired bootstrap 95% intervals.

checkpoint selection, recovers 42% of the gap against 68% for task training and still beats selection by $+0.018$ $[+0.009, +0.026]$ at 32 entries on validation rows never used in development (Appendix C).

4.3 The multi-document regime: when the budget binds

A single-document sender holds only the paragraph with the answer, so 32 entries compress it only $3.5\times$. Four added distractors bring the sender to 672 tokens on average, and the same budget to $21\times$.

The five documents are in random order, and the answer never appears in the receiver's documents. The message keeps the attention sink and the last $K-1$ positions, as in StreamingLLM (Xiao et al., 2024), instead of the pooled-attention rule of §3, because under causal masking only these positions have seen every document. On held-out test at 64 entries, the best of the eleven training-free policies scores 0.094, against 0.064 with no message and 0.278 with the full private context (Figure 7(c)). Writing scores 0.196, more than twice as much, and recovers **62%** of the gap at 64 entries and 37% at 32 (Figure 3(a)). It gains $+0.106$ $[+0.076, +0.136]$ over the selection arm it starts from, and a second seed gives the same score. The position-matched note stays at about the selection level, at 5% and 15% of the gap at 32 and 64 entries, and writing beats it by $+0.070$ $[+0.038, +0.102]$ and $+0.100$ $[+0.067, +0.134]$. A message written from another instance's documents drops the receiver to about the selection level (Appendix E), so the gain comes from the sender's own documents. Writing can even send what the kept positions miss: with the gold document first, the last positions lie in distractors, and at 64 entries on validation selection scores 0.048 against 0.046 with no message, yet writing recovers 37.7% of the gap (Appendix D).

Retrained under the random order for three more model families on HotpotQA and for Qwen3-8B on MuSiQue, writing beats both its selection arm and the position-matched note in all four cells at both budgets, with intervals excluding zero, and keeps 28% to 69% of the gap (Figure 3(b), Table 15).

Components. Figure 4 changes the objective and then the medium, with everything else matched. Raw-continuation distillation trains on the sender document's own continuation, without the receiver's evidence and question. Receiver supervision beats it by $+0.046$ $[+0.018, +0.076]$ at 32 entries

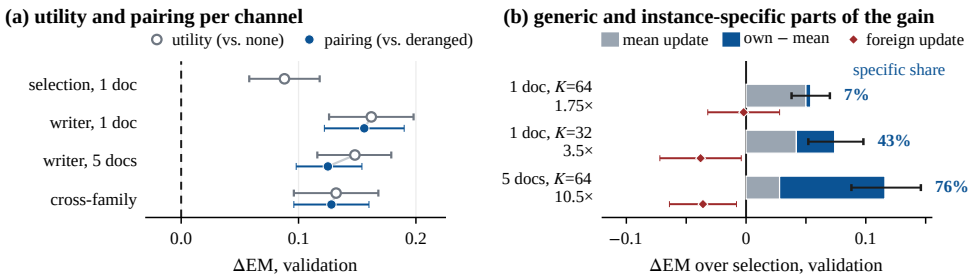


Figure 5: **The receiver reads the source, and the instance-specific share of the update grows as the budget binds.** Validation. **(a)** Utility (message minus none) and pairing effect (message minus a deranged message). Five documents: the random-order channel at $K{=}64$. Cross-family: the transported cache deranged. **(b)** Gain over selection as mean update plus instance-specific residual (95% interval, share printed). The diamond puts another instance's update on the same reference.

(Appendix C.2). A writer that sends K input embeddings, $18\times$ fewer values, recovers only 3% and 12% of the gap at 32 and 64 entries, about the selection level, so the gain needs the key–value medium (Appendix C.4). Both rollout sources work. Trained on the full-context teacher's answers instead of its own rollouts, the writer recovers 54% of the gap at 32 entries over two seeds, against 37%, and is level at 64. The lead at 32 entries lies in the document-independent part of the message (Appendix C.3).

4.4 Mechanism: dependence on the current instance

We test for instance dependence at two levels. At the whole-message level, another instance's message drops the receiver to within noise of the floor and destroying the entries yields 0.000 in the primary validation cell, so the receiver reads the channel and its content depends on the source (Figure 5(a)).

At the update level, holding r_i fixed and substituting another instance's update, $r_i + \Delta_{\pi(i)}$, reduces accuracy by 0.112 $[0.076, 0.148]$ and leaves the message 0.038 $[0.004, 0.072]$ below the initial selection baseline (Figure 5(b)). On held-out test that gap to selection is -0.038 $[-0.074, -0.004]$, and -0.011 $[-0.039, +0.016]$ averaged over 12 random derangements. Under 20 random derangements the pairing effect is $+0.106$ $[+0.080, +0.131]$, replicating on test at $+0.071$ $[+0.046, +0.098]$, and a matched update on another instance's reference scores -0.043 $[-0.060, -0.027]$ relative to deranging both, so the update is bound to its reference (Appendix E). A document-independent update, the per-slot mean of Δ_i over training documents, leaves an instance-specific residual that grows as the budget binds: null at 64 entries on single documents, $+0.032$ $[+0.010, +0.054]$ at 32, and $+0.088$ $[+0.060, +0.118]$ on the randomised multi-document channel at 64 entries (Figure 5(b), Table 18).

Answer exclusivity. In 171 of 500 HotpotQA validation instances the gold answer also appears in the receiver's documents, so we repeat the primary-cell analyses on the answer-exclusive remainder. Retained gain becomes 50% at 32 entries and 79% at 64. The whole-message pairing effect remains $+0.185$ $[+0.143, +0.228]$ and the update-level pairing effect $+0.100$ $[+0.064, +0.140]$, so the update's value still depends on its instance. A broader criterion with answer aliases and fuzzy matches keeps writing ahead of selection in both constructions (Appendix E.1). The five-document channel is answer-exclusive by construction, and its update-level pairing effect is $+0.144$ $[+0.114, +0.175]$ at 64 entries and $+0.098$ $[+0.070, +0.126]$ at 32 on held-out test (Appendix E).

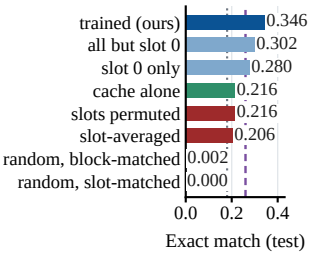
4.5 Communication across model families

Sharing cache entries assumes that the sender's entries line up with the receiver's, which fails across model families. A projection fitted once per pair (Heo et al., 2026) helps little on its own: 32 transported entries reach 0.206 against 0.196 with no message on validation, and 0.216 against 0.180 on test. The content crosses, but the receiver reads little of it.

What is missing is the same for every document. Our channel adds one fixed 9 MB tensor to every transported cache: the per-slot mean, over training documents, of a receiver-supervised correction to the projection. The receiver needs no prefill for it. The channel reaches 0.346 on held-out test against 0.260 for the position-matched note, $+0.086$ $[+0.050, +0.122]$, or 70% of the gap, and a

Table 2: **The cross-family channel on two pairs, and what its calibration has to be.** Left: exact match on HotpotQA-Private (n=500 per split, K=32), the second pair pooled over three training seeds. Right: the first pair on held-out test with the trained calibration replaced by six controls built from it. Dashed: position-matched note. Dotted: no message. Paired intervals: Appendix F.

	Llama-3.1→Qwen3		Qwen3→Falcon3	
Arm	Val.	Test	Val.	Test
No message	0.196	0.180	0.152	0.138
Position-matched 32-token note	0.280	0.260	0.272	0.214
Transported cache (32 entries)	0.206	0.216	0.194	0.190
+ fixed calibration (*ours*)	**0.328**	**0.346**	**0.336**	**0.360**
+ calibration, wrong document	0.200	0.234	0.261	0.291
+ closed-form calibration	0.220	0.242	0.194	0.186
Receiver's own full cache	0.470	0.418	0.508	0.508



second seed agrees. The transported cache still carries its document: with another document's, the channel falls to 0.234 on test, 0.112 $[0.080, 0.144]$ lower. The calibration must be learned through the receiver. Fitted in closed form to the receiver's own cache, it loses 0.104 $[0.070, 0.138]$. Random directions of its size destroy the receiver, and permuted slots remove the gain (Appendix F).

A second pair, Qwen3-8B into Falcon3-7B, replicates this (Table 2). Pooled over three seeds, it reaches 0.360 on test against the note's 0.214, $+0.146$ $[+0.103, +0.188]$. On the same 500 test rows, the first pair's channel scores -0.013 $[-0.042, +0.015]$ against the five-seed same-family writer, so the two are statistically indistinguishable (Appendix F).

A fixed part also appears inside a model family. There the mean update recovers 57% of the update's value at 32 entries and 93% at 64. The rest depends on the instance only where the budget binds, $+0.032$ $[+0.010, +0.054]$ at 32 entries, and not at all across model families (Table 18, Appendix F).

4.6 Selection headroom and cost

Headroom is the gap selection leaves below the full private context. Writing gains more where headroom is larger: across 171 cells from 28 checkpoints, gain and headroom correlate at Pearson $r = 0.71$ $[0.53, 0.80]$. Both quantities contain the selection score, but pairing each cell's selection with another cell's written score gives only $r = 0.11$. Controlling for budget and training regime leaves a partial correlation of 0.59 (Appendix G.1).

Producing a written message, both sender passes included, is $5.4\times$ faster than decoding a position-matched note on a single document and at least $4.3\times$ faster on five. Against free-length notes, one training run pays for itself after about $13{,}600$ single-document or $5{,}100$ five-document exchanges. At int4 the message keeps its quality in 1.22 MB, while int2 destroys the channel. The 32-entry message crosses PCIe 4 in 0.15 ms and 10 GbE in 3.7 ms (Appendix G.2).

5 Conclusion and Limitations

CacheWriter changes fixed-budget latent communication from choosing among sender-produced states to constructing states for a frozen receiver, and on average outperforms selection, text and soft-token channels at matched positions across four model families and four benchmarks. Within a model family, the improvement depends on the sender's current document, increasingly so as the budget binds. Across model families, the learned correction reduces to one fixed calibration, and the transported cache still carries the document. Latent-channel evaluation should therefore measure both utility and dependence on the transmitted instance.

The receivers we test are of similar size, and the benchmarks cover one task type, short-answer English question answering scored by exact match. Future work should test more complex tasks. Each channel serves one sender–receiver pair. A 32-entry message takes 4.6 MB against 122 bytes for a 32-token note, so on slow links text is cheaper to send, even at int4.

REPRODUCIBILITY STATEMENT

Code for building the benchmarks, training and evaluating CACHEWRITER, and running the baselines of the main text is available at https://anonymous.4open.science/r/cachewriter-anon--3DAB/, and its README gives the command for each experiment. Appendix A describes the benchmark construction, training settings, controls and statistics, and hardware.

USE OF AI ASSISTANTS

In this work, we used generative AI tools to provide feedback on research experiments and to clean and reformat datasets. We have not used generative AI tools to generate synthetic data sets, help develop theoretical models or conceptual frameworks, propose or refine hypotheses, or assist with translation, and formulating or proving mathematical claims and qualitative or thematic data analysis are not applicable to this work. Additionally, we used generative AI tools to identify relevant literature and polish the paper to improve readability. We have reviewed all AI-assisted work: every reported number traces to an archived result file, the authors checked each cited work against its published version, and the authors reviewed all AI-edited text against the experiments it reports. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=3zKtaqxLhW.

Yoav Benjamini and Yosef Hochberg. Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society: Series B (Methodological)*, 57(1):289–300, 1995.

Vivek Chari, Guanghui Qin, and Benjamin Van Durme. KV-Distill: Nearly lossless learnable context compression for LLMs, 2025. URL https://arxiv.org/abs/2503.10337.

Jiaming Cheng, Subhransu Das, and Rajiv Ramnath. When does latent communication pay? A causal audit of relayed KV caches in multi-agent LLMs, 2026. URL https://arxiv.org/abs/2608.04893.

Xin Cheng, Xun Wang, Xingxing Zhang, Tao Ge, Si-Qing Chen, Furu Wei, Huishuai Zhang, and Dongyan Zhao. xRAG: Extreme context compression for retrieval-augmented generation with one token. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=6pTlXqrO0p.

Alexis Chevalier, Alexander Wettig, Anirudh Ajith, and Danqi Chen. Adapting language models to compress contexts. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3829–3846, 2023. doi: 10.18653/v1/2023.emnlp-main.232. URL https://aclanthology.org/2023.emnlp-main.232/.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. Improving factuality and reasoning in language models through multiagent debate. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp, editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 11733–11763. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/du24e.html.

Falcon-LLM Team. The Falcon 3 family of open models, December 2024. URL https://huggingface.co/blog/falcon3.

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

Tao Ge, Jing Hu, Lei Wang, Xun Wang, Si-Qing Chen, and Furu Wei. In-context autoencoder for context compression in a large language model. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/0b276510ec2d3f6613a8b60c41ff0438-Paper-Conference.pdf.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, et al. The Llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. MiniLLM: Knowledge distillation of large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=5h0qf7IBZZ.

Zhiyu Guo, Hidetaka Kamigaito, and Taro Watanabe. Attention score is not all you need for token importance indicator in KV cache reduction: Value also matters. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 21158–21166, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.1178. URL https://aclanthology.org/2024.emnlp-main.1178/.

Shibo Hao, Sainbayar Sukhbaatar, DiJia Su, Xian Li, Zhiting Hu, Jason E Weston, and Yuandong Tian. Training large language models to reason in a continuous latent space. In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=Itxz7S4Ip3.

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, 2026. URL https://arxiv.org/abs/2608.03893.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps. In Donia Scott, Nuria Bel, and Chengqing Zong, editors, *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6609–6625, Barcelona, Spain (Online), December 2020. International Committee on Computational Linguistics. doi: 10.18653/v1/2020.coling-main.580. URL https://aclanthology.org/2020.coling-main.580/.

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 *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=VtmBAGCN7o.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In Regina Barzilay and Min-Yen Kan, editors, *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1147. URL https://aclanthology.org/P17-1147/.

Jang-Hyun Kim, Jinuk Kim, Sangwoo Kwon, Jae W. Lee, Sangdoo Yun, and Hyun Oh Song. KVzip: Query-agnostic KV cache compression with context reconstruction. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=JFygzwx8SJ.

Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, Online and Punta Cana, Dominican Republic, November

2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.243. URL https://aclanthology.org/2021.emnlp-main.243/.

Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.353. URL https://aclanthology.org/2021.acl-long.353/.

Yiping Li, Zhiyu An, and Wan Du. When less latent leads to better relay: Information-preserving compression for latent multi-agent LLM collaboration, 2026. URL https://arxiv.org/abs/2604.13349.

Yuhong Li, Yingbing Huang, Bowen Yang, Bharat Venkitesh, Acyr Locatelli, Hanchen Ye, Tianle Cai, Patrick Lewis, and Deming Chen. SnapKV: LLM knows what you are looking for before generation. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37, pages 22947–22970, 2024.

Jiyao Liu, Qi Zhang, Yaoyi Jia, Ziwen Kan, and Song Wang. Dual-cache latent space communication between heterogeneous language models, 2026. URL https://arxiv.org/abs/2608.20617.

Zichang Liu, Aditya Desai, Fangshuo Liao, Weitao Wang, Victor Xie, Zhaozhuo Xu, Anastasios Kyrillidis, and Anshumali Shrivastava. Scissorhands: Exploiting the persistence of importance hypothesis for LLM KV cache compression at test time. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, pages 52342–52364, 2023.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. In *International Conference on Learning Representations*, 2017.

Jesse Mu, Xiang Lisa Li, and Noah Goodman. Learning to compress prompts with gist tokens. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=2DtxPCL3T5.

Team OLMo, Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Shane Arora, Akshita Bhagia, Yuling Gu, Shengyi Huang, Matt Jordan, et al. 2 OLMo 2 furious, 2025. URL https://arxiv.org/abs/2501.00656.

Charles O'Neill, Alex Sandomirsky, Harry Partridge, Mudith Jayasekara, and Max Kirkby. Still: Amortized KV cache compaction in a single forward pass, 2026. URL https://arxiv.org/abs/2606.07878.

Zhuoshi Pan, Qianhui Wu, Huiqiang Jiang, Menglin Xia, Xufang Luo, Jue Zhang, Qingwei Lin, Victor Rühle, Yuqing Yang, Chin-Yew Lin, H. Vicky Zhao, Lili Qiu, and Dongmei Zhang. LLMLingua-2: Data distillation for efficient and faithful task-agnostic prompt compression. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Findings of the Association for Computational Linguistics: ACL 2024*, pages 963–981, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.57. URL https://aclanthology.org/2024.findings-acl.57/.

Chau Pham, Boyi Liu, Yingxiang Yang, Zhengyu Chen, Tianyi Liu, Jianbo Yuan, Bryan Plummer, Zhaoran Wang, and Hongxia Yang. Let models speak ciphers: Multiagent debate through embeddings. In *International Conference on Learning Representations*, 2024.

Vignav Ramesh and Kenneth Li. Communicating activations between language model agents. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=W6RPXUUFic.

Maximillian Rossi, Prajwal Raghunath, and Eugene Wu. Latent cache flow: Model-to-model communication without text, 2026. URL https://arxiv.org/abs/2605.22863.

Xiangyu Shi, Marco Chiesa, Gerald Q. Maguire, Jr., and Dejan Kostic. KVComm: Enabling efficient LLM communication through selective KV sharing. In *International Conference on Learning Representations*, 2026.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. MuSiQue: Multihop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554, 2022. doi: 10.1162/tacl_a_00475. URL https://aclanthology.org/2022.tacl-1.31/.

Zhongwei Wan, Xinjian Wu, Yu Zhang, Yi Xin, Chaofan Tao, Zhihong Zhu, Xin Wang, Siqi Luo, Jing Xiong, Longyue Wang, and Mi Zhang. D_2O: Dynamic discriminative operations for efficient long-context inference of large language models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=HzBfoUdjHt.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, Ahmed Hassan Awadallah, Ryen W White, Doug Burger, and Chi Wang. AutoGen: Enabling next-gen LLM applications via multi-agent conversations. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=BAakY1hNKS.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=NG7sS51zVF.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1259. URL https://aclanthology.org/D18-1259/.

Huixiang Zhang and Mahzabeen Emu. Do latent channels actually communicate? A causal audit of latent multi-agent LLM communication, 2026. URL https://arxiv.org/abs/2607.26773.

Yuxin Zhang, Yuxuan Du, Gen Luo, Yunshan Zhong, Zhenyu Zhang, Shiwei Liu, and Rongrong Ji. CaM: Cache merging for memory-efficient LLMs inference. In *International Conference on Machine Learning (ICML)*, pages 58840–58850, 2024. URL https://proceedings.mlr.press/v235/zhang24n.html.

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, Zhangyang Wang, and Beidi Chen. H_2O: Heavy-hitter oracle for efficient generative inference of large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, pages 34661–34710, 2023.

Jiaru Zou, Ruizhong Qiu, Gaotang Li, Xiyuan Yang, 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. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=syG9I9ofd8.

Roadmap. The appendix follows the order of the main text. Appendix A gives the experimental details and the statistical conventions. Appendices B and C extend §4.2–4.3 with the full fixed-budget record, the selection and text ladders, the KVComm comparison, and the objective, corpus and medium ablations. Appendix D extends §4.3, Appendix E extends §4.4, Appendix F extends §4.5, and Appendix G extends §4.6 with headroom and cost. Appendix H positions the method against its neighbours.

A EXPERIMENTAL DETAILS AND REPRODUCIBILITY

Benchmark construction. Each private-view instance comes from an existing multi-hop or open-domain question answering example. The paragraph containing the gold answer becomes the sender's document x_a. The remaining gold paragraphs and six shuffled distractors become the receiver's document x_b (on MuSiQue, seven paragraphs in all). The original question becomes q. The receiver's documents are shuffled. The multi-document sender adds four distractors to the answer-bearing paragraph and keeps only answer-exclusive instances (Appendix E.1). The released builders place the answer-bearing paragraph first, so the channels of §4.3 shuffle the sender's five documents per instance (Appendix D). For TriviaQA, we split long evidence documents into 700-character chunks and assign the answer-bearing chunk to the sender and up to seven others to the receiver. All four datasets use the same downstream format. The receiver prompt is byte-identical across message types except for the prefix contents. We score answers by normalised exact match after lowercasing and removing articles and punctuation. For Qwen3-8B, private contexts average 112 cache entries on HotpotQA, 191 on 2WikiMultihopQA, 127 on MuSiQue and 167 on TriviaQA.

Two properties of this construction affect interpretation. The floor is above zero on every benchmark because the receiver can guess or know the answer from the question and distractors (for Qwen3-8B, TriviaQA's is highest at 0.284), and the full-context reference is below 1.0, at 0.470 on HotpotQA-Private. Retained gain (Eq. 1) makes budgets comparable across these brackets.

Channel. The sender uses a rank-64 LoRA on the query, key, value, and output projections of every attention block. The LoRA-B matrices are zero-initialised. Adapters are enabled only for the forward pass that produces the transmitted cache. They are disabled during selection scoring, the teacher pass, and every receiver forward. A unit test verifies that an untrained adapter produces a message bitwise identical to the training-free selection message. The reported gain over selection therefore isolates the effect of training.

Selected entries are re-indexed to contiguous positions $0 \dots K-1$ by rotating their keys under the receiver's rotary encoding. The receiver therefore sees a compact prefix. This step assumes a single rotary stream over the full head dimension, which excludes partial-rotary models, models without positional encoding, and sliding-window models with a second positional stream.

Objective. We set $\lambda = 0.6$ in Eq. 3. At every step, the student decodes a fresh greedy continuation of up to 12 tokens. The divergence over continuation positions receives unit weight. Positions of q and of the chat template after it receive weight 0.2, except for the final prompt position, which receives unit weight because it determines the initial answer distribution. Positions inside x_b receive none. Training runs for $2\,500$ steps.

During task-trained channel training, a 48-instance validation probe scores the channel every 500 steps. Task-trained evaluations use the checkpoint with the highest probe exact match. The probe reads validation data only, and the held-out test split does not affect selection. The probe contains the first 48 of the 500 validation instances used for the headline results. On the remaining 452 the primary channels' improvements over selection are +0.069 [+0.040, +0.097] and +0.060 [+0.035, +0.086], against $+0.074$ and $+0.054$ on all 500, so the estimates hold without the probe instances. Every reported result re-evaluates the selected checkpoint on the full n=500 validation set.

Training-seed variance. Table 3 reports five training seeds of the primary cell. The seed-level written-minus-selection margin at 32 entries is $+0.072$ $[+0.044, +0.099]$, a mean with a t-based interval over n=5 seeds, and the written score has an across-seed standard deviation of 0.022. On the held-out test split the same seed-level margin is $+0.079$ $[+0.059, +0.099]$ over 5 seeds. All 5 seeds also carry a last-step battery. On validation the last step scores identically to the probe-selected checkpoint on 2 seeds, higher on 2 and lower on 1. Its seed-level margin is $+0.070$ $[+0.053, +0.086]$ on validation and $+0.069$ $[+0.051, +0.087]$ on test, and all 10 of its per-seed intervals exclude zero. Selecting by the probe therefore adds $+0.002$ to the margin on validation and $+0.010$ on test. The 64-entry answer-first channel of Table 13 diverged after step $1\,000$ without recovering, so its reported checkpoint precedes the divergence and the probe acts as early stopping.

Two engineering changes allow training on one accelerator: the language-model head is applied only to the supervised positions, which avoids materialising $[B, T, |V|]$ logit tensors, and the static key–value prefix is a stateless object that gradient checkpointing recomputes.

Table 3: **Five training seeds of the primary cell** (Qwen3-8B, HotpotQA-Private, K=32, n=500 per split). Written exact match and the paired written-minus-selection margin at the probe-selected checkpoint and at the last step. Table 1 pools all five, which share one recipe, and every seed is reported here at seed level.

Seed	val	val − sel.	test	test − sel.
Probe-selected checkpoint				
0	0.358	+0.074 [+0.046, +0.102]	0.340	+0.060 [+0.032, +0.090]
1	0.346	+0.062 [+0.036, +0.090]	0.358	+0.078 [+0.048, +0.108]
2	0.392	+0.108 [+0.078, +0.140]	0.380	+0.100 [+0.070, +0.132]
3	0.348	+0.064 [+0.040, +0.090]	0.370	+0.090 [+0.062, +0.120]
4	0.334	+0.050 [+0.026, +0.076]	0.348	+0.068 [+0.040, +0.098]
Last step				
0	0.358	+0.074 [+0.046, +0.102]	0.340	+0.060 [+0.032, +0.090]
1	0.374	+0.090 [+0.062, +0.120]	0.354	+0.074 [+0.044, +0.104]
2	0.348	+0.064 [+0.034, +0.094]	0.332	+0.052 [+0.022, +0.082]
3	0.348	+0.064 [+0.040, +0.090]	0.370	+0.090 [+0.062, +0.120]
4	0.340	+0.056 [+0.028, +0.086]	0.348	+0.068 [+0.042, +0.096]

Controls and statistics. The derangement arm, labelled wrong pair in the tables of other protocols, gives each receiver a message written from another instance's private document while holding message length and position layout fixed. The corruption arm destroys the transmitted entries. Under free decoding, corruption tests prefix destruction because every subsequent query attends to the corrupted columns, including the receiver's encoding of x_b. The derangement arm is the content control. When a non-content prefix effect is present, we report the margin over that arm. On Falcon3-7B HotpotQA at 32 entries, for instance, the deranged message scores 0.258 against a 0.152 floor and a 0.256 selection arm. All per-cell intervals use a paired bootstrap with 10^4 resamples of the same examples, and seed-level intervals are t-based. In the appendix, • marks a paired interval excluding zero. Where a statistic pools training seeds, the per-example differences are averaged over the seeds before resampling, so its interval is conditional on the trained seeds.

Hardware. Every job runs in bfloat16 on one 48 GB accelerator, except some cross-family fitting and evaluation jobs, which use two.

B Extended results for the fixed-budget frontier

This appendix reports the fixed-budget record behind the comparisons of §4.2. Constructions differ in their brackets, so each table carries its own floor and ceiling and retained gains are not comparable across them. Table 4 gives the four model families on single documents, on both splits.

Table 4: **Four model families, single-document construction**, HotpotQA-Private, exact match at 32 entries. Writing improves over the selection it starts from in every model family and matches or exceeds a position-matched text note in every model family at this budget, significantly so on Llama-3.1 and Falcon3. Every test Δ is positive. Rows carrying two seeds average the per-seed arm means, and their paired difference is computed on the elementwise seed-mean per-example vectors. The Qwen3-8B row stays on the frozen primary seed, whereas Table 1 pools its five seeds to 0.356.

		Validation (n=500)					Test (n=500)		
Receiver	Seeds	None	Text 32	Selection	CacheWriter	Full cache	Selection	CacheWriter	Δ
Qwen3-8B	0	0.196	0.328	0.284	**0.358**	0.470	0.280	0.340	+0.060
Llama-3.1-8B	0,1	0.224	0.352	0.376	**0.438**	0.526	0.394	0.475	+0.081
OLMo-2-7B	0,1	0.212	0.292	0.274	**0.311**	0.368	0.244	0.262	+0.018
Falcon3-7B	0,1	0.152	0.232	0.256	**0.402**	0.508	0.246	0.387	+0.141

B.1 The larger budget

Doubling the budget to 64 entries (Table 5, Figure 6) reduces the average margin over the strongest position-matched baseline from $+0.053$ to $+0.017$, and 5 of 16 cells are not positive. Baseline headroom falls from 0.126 to 0.049, while the average share of it converted by writing falls from 34% to 18%. The additional 32 entries improve the best selection policy by $+0.089$ but writing by $+0.041$. Writing captures 46% of the improvement that the larger payload gives its baseline.

			Position-matched baselines			Cache writing			
Receiver	Benchmark	None	Note	Sel.	CIPHER	CACHEWRITER	Δ	R_{KV}	Full KV
K = 64 entries (text notes: 64 tokens; CIPHER: 64 soft tokens)									
Qwen3-8B	HotpotQA*	0.196	0.364	0.382	0.330	0.433	**+0.051**	86%	0.470
	2Wiki	0.092	0.236	0.258	0.196	0.319	**+0.061**	91%	0.340
	MuSiQue	0.026	0.218	0.228	0.190	0.262	**+0.034**	99%	0.264
	TriviaQA	0.284	0.412	0.446	0.354	0.481	**+0.035**	85%	0.516
Llama-3.1-8B	HotpotQA*	0.224	0.456	0.480	0.400	0.482	**+0.002**	85%	0.526
	2Wiki	0.090	0.290	0.296	0.194	0.337	**+0.041**	86%	0.376
	MuSiQue	0.050	0.226	0.294	0.188	0.295	**+0.001**	81%	0.352
	TriviaQA	0.428	0.502	0.638	0.490	0.615	−0.023	77%	0.670
OLMo-2-7B	HotpotQA*	0.212	0.326	0.332	0.316	0.317	−0.015	67%	0.368
	2Wiki	0.078	0.134	0.104	0.102	0.122	−0.012	67%	0.144
	MuSiQue	0.024	0.150	0.162	0.152	0.173	**+0.011**	96%	0.180
	TriviaQA	0.314	0.396	0.444	0.372	0.433	−0.011	77%	0.468
Falcon3-7B	HotpotQA*	0.152	0.278	0.456	0.210	0.476	**+0.020**	91%	0.508
	2Wiki	0.036	0.092	0.318	0.104	0.316	−0.002	91%	0.344
	MuSiQue	0.014	0.156	0.238	0.086	0.279	**+0.041**	90%	0.308
	TriviaQA	0.206	0.370	0.448	0.236	0.489	**+0.041**	96%	0.502
Mean		0.152	0.288	0.345	0.245	0.364	**+0.017**	—	—

Table 5: **Exact match, four receivers × four benchmarks at** K=64 (n=500 validation per cell). Channels train on HotpotQA-Private only (*), so the other rows are transfer. Writer arms average all training seeds: three for every Qwen3 cell and two otherwise. Sel. is the best of the eleven deployable policies of Figure 7(b) and the writer's own selection arm, a separate decode of pooled attention that raises it in three cells by at most 0.004. Green : best position-matched baseline, the reference for Δ. Blue : ours. R_{KV}: fraction of the gap from no message to the full cache recovered. CIPHER sends 16–64× fewer bytes.

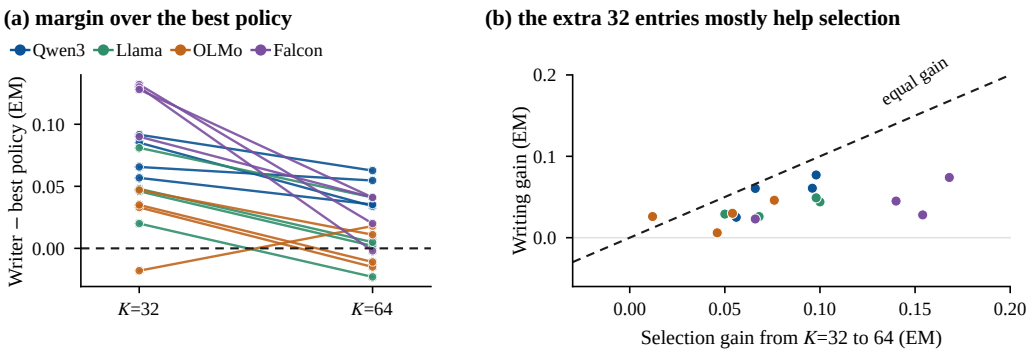


Figure 6: **Doubling the budget helps selection more than writing. (a)** CACHEWRITER minus the best deployable selection policy on each of the 16 cells, validation, at 32 and 64 entries, coloured by receiver. **(b)** What the extra 32 entries add to the selection arm the writer starts from and to the writer itself. Below the diagonal, selection gains more.

The 64-entry channels reach a lower final divergence than the 32-entry channels in every model family we inspect, so the larger-budget channels match the distillation target more closely while achieving a smaller downstream margin. Writing has its largest advantage when the compression ratio strongly constrains selection, and that advantage recedes as selection approaches the ceiling.

B.2 TRAINING-FREE SELECTION POLICIES

Figure 7(a) lists the sink-retaining policies at 32 entries on every cell of Table 1. Each keeps the attention sink and $K-1$ entries of the sender's own cache ranked by its rule: StreamingLLM (Xiao et al., 2024), H2O (Zhang et al., 2023), SnapKV (Li et al., 2024), value-norm scoring (Guo et al., 2024), KVzip (Kim et al., 2025) and CaM / D2O (Zhang et al., 2024; Wan et al., 2025). KVzip runs with our scorer, which uses our own repeat prompt, and with the official template, and CaM / D2O, which also merges each evicted value into its nearest kept entry, runs on value-norm ranking as well.

Uniform stride spreads the $K-1$ entries evenly over the sender, and content salience spreads them over tokens that contain a capital letter or a digit, a proxy for names and numbers. Pooled attention is the rule that initialises CACHEWRITER, and the query-aware oracle ranks with the receiver's question, which the sender does not hold. Figure 7(b) repeats all twelve at 64 entries (Table 5), and Figure 7(c) carries them to the five random-order multi-document cells, where the writer is the sink-plus-recent channel of §4.3. Three sink-free variants collapse toward the floor at tight budgets in an n=300 study on the primary cell (0.193–0.233 at $K \in \{8, 16\}$ against its 0.213 floor), so every listed policy reserves one entry for the sink. No single policy dominates: at 32 entries content salience is the per-cell best in 9 of 16 cells, pooled attention in 4 and H2O in 3. At 64 entries content salience leads in 7 and value norm in 6. Averaged over the 16 cells the deployable policies run from 0.193 (StreamingLLM) to 0.250 (content salience) at 32 entries, with a per-cell spread of 0.071, and the query-aware oracle differs from the best deployable policy by -0.001 at 32 entries and $+0.003$ at 64, so on single documents ranking with the question adds nothing over the best deployable policy. Pooling attention mass over a width-7 window before ranking is worth +0.013 over unpooled attention at 64 entries on average, because evidence useful to a reader arrives in contiguous spans.

On the five random-order cells, held-out test, StreamingLLM is the per-cell best in 3 of 5 cells at 32 entries and SnapKV in 2 and StreamingLLM in 2 at 64, and writing leads the best policy in all five cells at both budgets, by +0.085 and +0.083 on average (Figure 7(c)). On the Qwen3 HotpotQA channel the best deployable policy reaches 0.076 at 32 entries and 0.094 at 64 on held-out test (StreamingLLM and SnapKV), against 0.076 and 0.090 for the sink-plus-recent initialisation arm. On single documents the ladder beats the initialisation rule in 24 of 32 cells. A 672-token sender leaves a training-free policy far less to exploit, which is the observation the headroom analysis of Appendix G.1 reports from the other side.

B.3 TEXT BASELINES

Table 6 reports the text ladder on selected cells. The same frozen sender decodes a note about its private document, and the note occupies the same position in the receiver prompt as the latent message.

The position-matched arm caps the note at K tokens, matching a K-entry latent message in their common unit of receiver positions. The free-length arm removes this cap. LLMLingua-2 (Pan et al., 2024), a learned task-agnostic compressor, does not outperform a truncated note at either budget.

Budget-aware note. The note above is decoded from a prompt that never states the budget and is then truncated to K receiver tokens, so it is cut mid-sentence and spends part of its allowance on framing such as "Here's a note to Agent B". We re-decode all 32 main-table cells with the budget stated in the sender's prompt and the preamble suppressed, changing nothing else. The budget-aware note is the stronger arm in 19 of 32 cells. Its mean effect is $+0.017$ at 32 entries, where it wins 11 of 16 cells, and -0.005 at 64, where it wins 8, so a tighter budget is where budget-blindness costs most. Over the 16 cells decoded both ways, suppressing the preamble is worth $+0.031$ and stating the budget -0.022: the gain comes from not spending receiver positions on boilerplate. Substituting this note into Table 1's position-matched maximum would move the mean advantage from $+0.053$ to $+0.048$ at 32 entries, with 13 of 16 cells positive against 14 under the budget-blind note, and would improve the 64-entry aggregate. The main tables report the budget-blind note.

Table 7 compares latent and text messages on the same examples with paired intervals for the single-document cells, and Table 15 does so on held-out test for the random-order multi-document cells. Figure 8 visualises the complete family on validation, with the seed-1 single-document cells and the random-order cells entered per training seed. Across 31 cells, the written channel is significantly higher in 20, statistically tied in 11, and significantly lower in 0, and 19 cells survive Benjamini–Hochberg control at q=0.05. Eight of the eleven random-order cells belong to the first group, and on the held-out test split writing exceeds the note with an interval excluding zero in four of four transfer cells at 64 entries and four at 32. Ties occur in single-document cells, in the OLMo-2 random-order cells, and in the MuSiQue random-order cell at 64 entries. No cell is significantly below text. The closest is the second seed of the single-document OLMo-2 cell at 64 entries, whose interval includes zero. With 95% familywise confidence over this family, no cell favours the note by more than 0.100.

(a) single document, K=32, validation

	Qwen3-8B				Llama-3.1-8B				OLMo-2-7B				Falcon3-7B			
	HP*	2W	MQ	TQ	HP*	2W	MQ	TQ	HP*	2W	MQ	TQ	HP*	2W	MQ	TQ
StreamingLLM	.248	.146	.070	.308	.326	.146	.084	.470	.242	.058	.032	.324	.216	.076	.042	.294
H2O	.272	**.202**	.112	.344	.384	.212	.192	**.566**	.268	.060	.060	.352	.252	**.158**	.098	.362
SnapKV	.254	.158	.068	.310	.340	.162	.110	.516	.240	.072	.026	.328	.240	.086	.054	.324
value norm	.268	.162	**.116**	.354	.384	.182	.158	.510	.266	.084	.078	**.368**	.256	.116	.076	.330
KVzip (our scorer)	.242	.154	.068	.314	.348	.178	.124	.492	.262	.084	.050	.318	.202	.098	.052	.284
KVzip	.248	.140	.074	.304	.340	.184	.106	.478	.260	.080	.042	.324	.204	.092	.048	.272
CaM/D2O	.268	.184	.092	.352	.368	.192	.162	.548	.260	.068	.058	.302	.246	.152	.094	.354
CaM/D2O on value norm	.256	.152	.086	.330	.346	.182	.132	.504	.268	**.114**	.070	.346	.238	.120	.072	.322
uniform stride	.246	.168	.048	.336	.342	.202	.122	.516	.248	.064	.034	.346	.226	.122	.044	.296
content salience	**.290**	.192	**.116**	.362	**.392**	**.230**	**.198**	.546	**.278**	.074	.074	.356	**.270**	.136	**.106**	**.376**
pooled attention	.284	**.202**	**.116**	**.364**	.376	.212	.196	.554	.274	.082	**.080**	.344	.256	.148	.098	.368
query-aware oracle†	.302	.210	.112	.380	.408	.224	.202	.588	.248	.060	.060	.316	.296	.170	.124	.390
CacheWriter	**.356**	**.294**	**.201**	**.421**	**.438**	**.311**	**.246**	**.586**	.311	.096	**.127**	**.403**	**.402**	**.288**	**.234**	**.466**

(b) single document, K=64, validation

	Qwen3-8B				Llama-3.1-8B				OLMo-2-7B				Falcon3-7B			
	HP*	2W	MQ	TQ	HP*	2W	MQ	TQ	HP*	2W	MQ	TQ	HP*	2W	MQ	TQ
StreamingLLM	.340	.182	.128	.360	.420	.250	.196	.566	.274	.094	.094	.386	.388	.244	.158	.356
H2O	.364	**.256**	.204	.428	.466	.284	.256	.598	.310	.072	.150	.398	.402	.316	.206	.386
SnapKV	.332	.220	.138	.378	.424	.262	.204	.572	.276	.070	.102	.378	.414	.276	.170	.410
value norm	**.378**	.230	.184	**.446**	**.480**	**.296**	.246	**.638**	.328	.096	**.162**	.438	.408	.292	.214	.426
KVzip (our scorer)	.332	.214	.156	.364	.438	.280	.220	.578	.276	.084	.128	.380	.434	.292	.182	.358
KVzip	.336	.208	.150	.366	.430	.274	.202	.556	.286	.088	.108	.390	.406	.292	.192	.368
CaM/D2O	.366	.250	.188	.394	.468	.284	.276	.594	.310	.086	.134	.408	.450	.312	**.238**	.414
CaM/D2O on value norm	.368	.224	.180	.422	.460	.282	.236	.622	**.332**	**.104**	.142	.434	.426	.304	.216	.434
uniform stride	.324	.220	.148	.374	.432	.284	.200	.594	.296	.072	.094	.382	.408	.298	.180	.384
content salience	**.378**	.252	**.228**	.430	.470	.292	.270	.632	**.332**	.094	.150	**.444**	**.456**	**.318**	.232	**.448**
pooled attention	**.378**	**.256**	.214	.430	.468	.292	**.290**	.604	.306	.090	.150	.398	.444	**.318**	.232	.434
query-aware oracle†	.392	.262	.200	.474	.504	.296	.306	.652	.330	.062	.126	.434	.494	.322	.234	.470
CacheWriter	**.433**	**.319**	**.262**	**.481**	**.482**	**.337**	**.295**	.615	.317	**.122**	**.173**	.433	**.476**	.316	**.279**	**.489**

(c) five documents in random order, held-out test

	K=32 entries					K=64 entries				
	Q-HP	L-HP	O-HP	F-HP	Q-MQ	Q-HP	L-HP	O-HP	F-HP	Q-MQ
StreamingLLM	**.076**	**.136**	**.084**	.066	.006	.090	**.184**	**.100**	.096	.010
H2O	.072	**.136**	.072	.062	**.014**	.086	.174	.084	.098	.024
SnapKV	.072	.134	.074	**.074**	.010	**.094**	.162	.096	**.120**	.012
value norm	.064	.120	.074	.048	**.014**	.088	.148	.072	.080	.022
KVzip (our scorer)	.060	.130	.066	.050	.004	.074	.146	.076	.094	.006
KVzip	.064	.128	.070	.058	.004	.078	.144	.070	.082	.006
CaM/D2O	.066	.124	.060	.052	.002	.062	.142	.072	.098	.006
CaM/D2O on value norm	.064	.124	.078	.040	.004	.062	.128	.078	.088	.004
uniform stride	.064	.126	.080	.054	.004	.062	.148	.076	.086	.002
content salience	.070	**.136**	.070	.062	.006	.078	.162	.078	.112	.010
pooled attention	.074	.124	.082	.056	.012	.088	.160	.084	.112	**.026**
query-aware oracle†	.082	.204	.090	.120	.008	.134	.270	.108	.188	.036
CacheWriter	**.144**	**.242**	**.132**	**.236**	**.054**	**.196**	**.272**	**.140**	**.262**	**.068**

Figure 7: **Every training-free selection policy on every cell**, exact match, n=500 per cell. Shading is each cell's retained gap. A green box marks the best deployable policy per cell, and CACHEWRITER is bold where it exceeds it. †Ranks with the receiver's question, so it bounds selection from above. **(a)** The 16 cells of Table 1: HP* HotpotQA (in-distribution), 2W 2Wiki, MQ MuSiQue, TQ TriviaQA. **(b)** The same cells at K=64. **(c)** The five random-order cells of Table 15, held-out test, with Q, L, O, F for Qwen3-8B, Llama-3.1-8B, OLMo-2-7B and Falcon3-7B.

Table 6: **Selected text-message baselines**, n=500 validation instances. The sender decodes a note about its private document under a token budget, and the frozen receiver reads the note in the same prompt slot the latent message occupies. *Free length* removes the budget and *full context* hands over the private document verbatim. The two right-hand columns are unconstrained reference arms, not ordered bounds.

Receiver	Benchmark	None	32 tok	64 tok	Free length	Full context
Qwen3-8B	HotpotQA	0.196	0.328	0.364	0.408	0.446
Qwen3-8B	MuSiQue	0.026	0.172	0.218	0.244	0.262
Qwen3-8B	2Wiki	0.092	0.234	0.236	0.246	0.274
Qwen3-8B	TriviaQA	0.284	0.346	0.412	0.452	0.446
Llama-3.1-8B	HotpotQA	0.224	0.352	0.456	0.468	0.520
OLMo-2-7B	HotpotQA	0.212	0.292	0.326	0.340	0.404
Falcon3-7B	HotpotQA	0.152	0.232	0.278	0.270	0.308
LLMLingua-2 (Pan et al., 2024) compressed notes, identical receiver						
Qwen3-8B	HotpotQA	0.196	0.280	0.358	—	0.446

Table 7: **Latent against text on the same examples**, n=500 per cell. The text arm is a K-token note from the same sender read by the same frozen receiver through a byte-identical prompt, so the comparison isolates the medium. • marks a paired interval excluding zero. The rows are the primary seed of each single-document cell, which Table 1 pools with the later seeds. The multi-document notes are paired with the writer in Table 15. A tie is a failure to reject at the cell's sample size.

Receiver	Benchmark	K	Text	Written	Full	Written − text	
Qwen3-8B	HotpotQA	32	0.328	**0.358**	0.470	$+0.030$ $[-0.006, +0.064]$	
Qwen3-8B	HotpotQA	64	0.364	**0.436**	0.470	$+0.072$ $[+0.034, +0.110]$	•
Qwen3-8B	MuSiQue	32	0.172	**0.196**	0.264	$+0.024$ $[-0.012, +0.062]$	
Qwen3-8B	MuSiQue	64	0.218	**0.258**	0.264	$+0.040$ $[+0.004, +0.076]$	•
Qwen3-8B	2Wiki	32	0.234	**0.286**	0.340	$+0.052$ $[+0.016, +0.088]$	•
Qwen3-8B	2Wiki	64	0.236	**0.316**	0.340	$+0.080$ $[+0.042, +0.118]$	•
Qwen3-8B	TriviaQA	32	0.346	**0.416**	0.516	$+0.070$ $[+0.036, +0.104]$	•
Qwen3-8B	TriviaQA	64	0.412	**0.476**	0.516	$+0.064$ $[+0.028, +0.100]$	•
Llama-3.1-8B	HotpotQA	32	0.352	**0.428**	0.526	$+0.076$ $[+0.038, +0.114]$	•
Llama-3.1-8B	HotpotQA	64	0.456	**0.470**	0.526	$+0.014$ $[-0.026, +0.054]$	
OLMo-2-7B	HotpotQA	32	0.292	**0.314**	0.368	$+0.022$ $[-0.014, +0.060]$	
OLMo-2-7B	HotpotQA	64	0.326	**0.346**	0.368	$+0.020$ $[-0.022, +0.062]$	
Falcon3-7B	HotpotQA	32	0.232	**0.392**	0.508	$+0.160$ $[+0.118, +0.202]$	•
Falcon3-7B	HotpotQA	64	0.278	**0.458**	0.508	$+0.180$ $[+0.132, +0.226]$	•

B.4 BYTE-MATCHED LAYER SHARING: KVCOMM

KVComm is the closest published formulation of our problem. We run the authors' released implementation on our private-view instances with the same model pair, avoiding dataset and model-pair confounds while retaining implementation and metric-pipeline differences. Table 8 reports results on three benchmarks, and Figure 9 plots the HotpotQA budget curve.

Metric reconciliation. The released harness labels its primary metric F1, but computes a binary match rate. It thresholds token F1 at 0.5 for each example and averages the resulting indicator. The published Llama-3.1-8B HotpotQA values of 0.33, 0.60, and 0.69 at $\rho \in \{0.3, 0.5, 0.7\}$ use this match rate. We save per-example predictions and rescore them with strict exact match and token F1 (Table 8).

What the comparison shows. At $\rho = 0.3$, KVComm matches the bytes in our 32-entry message on HotpotQA to within two percentage points of the full cache, and sends more on the two benchmarks with longer private contexts, 30% against 17% on 2WikiMultihopQA and 25% on MuSiQue. §4.2 compares the retained gains. As in those published values, most of the gain arrives between $\rho = 0.3$ and $\rho = 0.5$ on HotpotQA and 2WikiMultihopQA, and between $\rho = 0.5$ and $\rho = 0.7$ on MuSiQue.

Scope of the comparison. The two methods compress orthogonal dimensions of the same object. At $\rho = 1.0$, KVComm transmits every layer and position, reaches 0.544, and exceeds its full-text reference of 0.496.

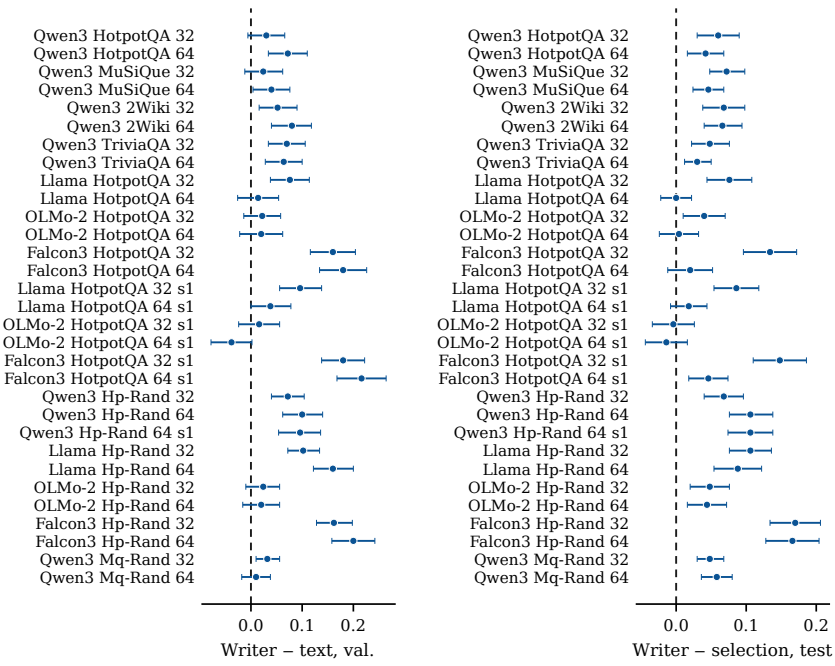


Figure 8: **The full paired record as intervals.** Left: written minus position-matched text on validation, the complete 31-cell paired family (seed-1 single-document and random-order multi-document cells per training seed). Right: written minus selection on the held-out test split, every cell of the family with a test battery. Hp-Rand and Mq-Rand are the random-order HotpotQA and MuSiQue cells. Test-split notes for the random-order cells are in Table 15. Whiskers are paired bootstrap 95% intervals, estimating sampling uncertainty given the training seeds.

KVComm also carries instance-specific content. At $\rho = 0.5$, a wrong-pair control reduces its score from 0.390 to 0.236, close to its 0.216 floor. A concurrent preregistered five-seed audit agrees, with deranged-cache controls (Cheng et al., 2026). KVComm has no corruption arm here.

Within-harness layer sharing. We also implement layer-selective sharing in our own harness. Calibrated on 8 examples it selects late layers and gains $+0.002$ over the floor at $\rho = 0.3$. The official run calibrates on 1 example, the released default, selects early layers and gains $+0.042$, and 3 runs of the authors' implementation on the same input returned 3 different layer sets. Decoded with the official layer lists on 500 Qwen3-8B HotpotQA-Private validation instances, our implementation gains $+0.030$, 0.012 below the official run, and its wrong-pair arm stays at the floor, 0.192. The layer set therefore explains most of the gap between the two implementations. Official KVComm results give the primary comparison and our implementation a within-harness check.

C Objective, training corpus and medium

This appendix ablates the training signal of §3, holding the budget, selector and optimisation fixed: the training corpus, the presence of the receiver's evidence and question in the objective, the rollout source, the message medium, and the task-free channel under a fully label-free protocol. Figure 4 in §4.3 summarises the objective and medium ablations.

C.1 Training corpus

Table 9 varies the data used to train the channel at a fixed architecture and budget, and §4.2 reports the comparison. HotpotQA has twice as many distinct documents, so the matched-size comparison retrains the transferred channel on a $10{,}000$-instance subset, which ties native training on 2Wiki and leads it on MuSiQue by $+0.044$ $[+0.018, +0.070]$, and a three-dataset mixture adds no average gain.

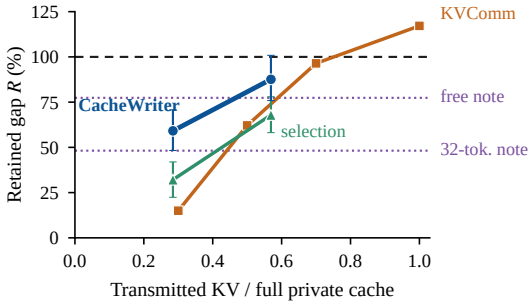


Figure 9: **Retained gap against transmitted bytes**, HotpotQA-Private, Qwen3-8B, validation. CACHEWRITER and its selection arm at 32 and 64 entries with paired bootstrap 95% intervals, KVComm at $\rho \in \{0.3, 0.5, 0.7, 1.0\}$ in its own harness and bracket, and the text notes as reference levels. KVComm exceeds 100% where it beats the full-text reference of its harness.

Table 8: **Cross-harness comparison with KVComm**, Qwen3-8B, n=500 validation instances, run with the authors' released implementation on our private-view data. KVComm rows are scored in its own harness against its own floor and ceiling, and CACHEWRITER rows in ours. R is each protocol's retained gain within its own harness. ρ is the fraction of layers transmitted at all positions. Payload is the transmitted fraction of the full private cache. On HotpotQA the payloads of ρ=0.3 and of our K=32 differ by under two percentage points. On MuSiQue, writing at a 25% payload stays above KVComm even at ρ=0.7, more than twice the payload.

Benchmark	Protocol	Payload	EM	F1	R
HotpotQA	no message	—	0.216	0.338	0%
	KVComm ρ=0.3	30%	0.258	0.393	15%
	KVComm ρ=0.5	50%	0.390	0.549	62%
	KVComm ρ=0.7	70%	0.486	0.633	96%
	CACHEWRITER K=32	28%	**0.358**	0.539	**59%**
	full private context	100%	0.496	0.717	100%
2Wiki	no message	—	0.140	0.220	0%
	KVComm ρ=0.3	30%	0.200	0.288	24%
	KVComm ρ=0.5	50%	0.308	0.396	67%
	KVComm ρ=0.7	70%	0.350	0.454	83%
	CACHEWRITER K=32	17%	**0.286**	0.403	**78%**
	full private context	100%	0.392	0.554	100%
MuSiQue	no message	—	0.022	0.125	0%
	KVComm ρ=0.3	30%	0.048	0.161	10%
	KVComm ρ=0.5	50%	0.064	0.190	16%
	KVComm ρ=0.7	70%	0.192	0.349	64%
	CACHEWRITER K=32	25%	**0.196**	0.374	**71%**
	full private context	100%	0.288	0.514	100%
HotpotQA	KVComm ρ=1.0 (all layers)	100%	0.544	0.712	117%
HotpotQA	KVComm ρ=0.3, wrong pair	30%	0.230	0.348	5%
HotpotQA	KVComm ρ=0.5, wrong pair	50%	0.236	0.334	7%

Table 9: **Training-corpus sensitivity at a fixed architecture and budget.** Qwen3-8B, K=32, one evaluation bracket per benchmark. Row 1 is the channel of Table 1. †Row 4 sees only raw-document continuation, selects its checkpoint without labels, and is evaluated on validation rows 500 and later (Appendix C.5), so compare it through R. TriviaQA has no training split, so row 2 has no entry there. Row 1 pools 5 training seeds per benchmark, rows 2–4 pool 1–5 (row 2 on HotpotQA repeats row 1).

Training corpus	HotpotQA	2Wiki	MuSiQue	TriviaQA	Avg R
Trained on HotpotQA only	0.356	0.294	0.201	0.421	68%
Trained on this benchmark	0.356	0.270	0.174	—	—
Three-dataset mixture (TriviaQA held out)	0.368	0.296	0.168	0.406	64%
Raw text only (task-free)†	0.270	0.228	0.108	0.434	42%

C.2 Receiver supervision against raw-continuation distillation

Raw-continuation distillation clears the receiver's evidence and question and supervises the sender document's own continuation, so it ablates our supervision signal. Table 10 trains it on the random-order multi-document corpus with the selector, budget, adapter rank, step count and effective batch of the channel of §4.3. The final 500 characters of each five-document sender serve as its continuation target, and the label-free rule of Appendix C.5, run on 200 held-out senders of the same corpus, chooses its checkpoint. Raw continuation reaches 0.122 against 0.196 for receiver supervision at 64 entries, 27% against 62% of the gap, and 0.098 against 0.144 at 32, paired $+0.074$ $[+0.046, +0.104]$ • and $+0.046$ $[+0.018, +0.076]$ • on the same rows. KV-Distill keeps the instruction visible to its decoder and takes its divergence over answer tokens (Chari et al., 2025). Appendix C.3 reports that instruction-conditioned comparison, which keeps the receiver's evidence and question in place and changes only the rollout source.

Table 10: **The objective under the random order.** Raw-continuation distillation on the random-order multi-document corpus with the random-order channel's selector, budget, adapter and step count, checkpoint by the label-free rule of Appendix C.5. Held-out test, exact match with retained gap in parentheses. Floor, selection and full arms are identical per example across the two runs. • marks a paired bootstrap 95% interval excluding zero.

Budget	Selection	Raw continuation	Receiver supervision	Raw − Sel.	Receiver − raw
K=64	0.090 (12%)	0.122 (27%)	**0.196** (62%)	$+0.032$ $[+0.012, +0.054]$ •	$+0.074$ $[+0.046, +0.104]$ •
K=32	0.076 (6%)	0.098 (16%)	**0.144** (37%)	$+0.022$ $[+0.004, +0.042]$ •	$+0.046$ $[+0.018, +0.076]$ •

C.3 The rollout source

Raw-continuation distillation changes two things at once relative to our objective: it removes the receiver's evidence and question, and it supervises a corpus continuation instead of an answer trajectory. This arm changes only the second. We retrain the channel of §4.3 with on-policy rollouts disabled and every other setting unchanged, including the 48-example probe rule for checkpoint selection. The supervised trajectory then comes from the full-context teacher while the receiver still holds x_b and q. This is the controlled ablation of the rollout source, and also the closer analogue of KV-Distill's setting, in which the context is compressed, the instruction remains uncompressed for the decoder, and the divergence is taken over answer tokens (Chari et al., 2025). Table 11 reports it, with two off-policy seeds at 32 entries and one at 64.

At 32 entries the off-policy writer is ahead. Pooled over both seeds it recovers 54% of the test gap against 37% on-policy, ahead by $+0.036$ $[+0.013, +0.059]$ • on test and $+0.026$ $[+0.005, +0.048]$ • on validation. The second seed alone is ahead by $+0.040$ $[+0.014, +0.066]$ • on test and level on validation, $+0.004$ $[-0.020, +0.028]$. Deranging the whole message costs $+0.057$ $[+0.036, +0.079]$ •, so the off-policy writer transmits the instance as the on-policy channel does. At 64 entries the two are level: the off-policy writer recovers 64% against 62%, a paired $+0.004$ $[-0.020, +0.028]$ on test and $+0.004$ $[-0.017, +0.025]$ against both on-policy seeds pooled. Both rollout sources work.

A post-hoc split locates the 32-entry lead. The instance-specific part of the message, written minus deranged, is level between the two rollout sources, $+0.007$ $[-0.015, +0.030]$ on test and -0.004 $[-0.027, +0.019]$ on validation, while the deranged message itself scores higher by $+0.029$ $[+0.017, +0.042]$ • and $+0.030$ $[+0.013, +0.048]$ •. The off-policy writer's advantage is therefore a document-independent part of the message, the analogue of the mean update of Figure 5(b).

C.4 The message medium at a matched objective

Table 11 also holds the objective fixed and changes the message. The soft-prefix writer keeps the K/V writer's LoRA parameterisation (a fresh adapter of the same rank and locus), selector, mixed on-policy divergence, random-order corpus, $2\,500$ steps and 48-example probe, and writes K input embeddings instead of K key–value entries: the receiver's own embeddings of the selected sender tokens plus a rank-64 zero-initialised map of the adapted sender's last-layer states, scaled into the receiver's input range. At step 0 the message equals that token prefix exactly. The embeddings occupy the message slot at the positions CIPHER uses, so the arm is position-matched to the paper's soft-token baseline. The soft prefix recovers 3% of the test gap at 32 entries and nothing on validation (-1%), against 37% and 31% for the key–value writer, a paired -0.074 $[-0.102, -0.048]$ • on test, with a pairing effect of $+0.000$ $[-0.006, +0.006]$ on validation and $+0.010$ $[+0.002, +0.020]$ • on

Table 11: **Rollout source and message medium under the random order.** Each arm retrains the channel of §4.3 at the budget shown, seed 0 unless pooled, and changes one thing. The soft prefix's reference is the token prefix of the selected sender tokens. Held-out test, first 500 rows of the shuffled file. R uses the on-policy channel's bracket, and pairing is the written message minus a message written for another document. Validation contrasts are given in the text. • marks a paired bootstrap 95% interval excluding zero.

Arm	Reference	Written	R	Written − on-policy K/V	Pairing
K=32 *receiver positions*					
on-policy K/V writer (§4.3)	0.076	0.144	37%	—	$+0.050$ $[+0.028, +0.074]$ •
off-policy K/V writer	0.076	0.176	52%	$+0.032$ $[+0.006, +0.058]$ •	$+0.058$ $[+0.034, +0.084]$ •
seeds 0 and 1 pooled	0.076	0.180	54%	$+0.036$ $[+0.013, +0.059]$ •	$+0.057$ $[+0.036, +0.079]$ •
soft prefix (embeddings)	0.056	0.070	3%	-0.074 $[-0.102, -0.048]$ •	$+0.010$ $[+0.002, +0.020]$ •
K=64 *receiver positions*					
on-policy K/V writer (§4.3)	0.090	0.196	62%	—	$+0.102$ $[+0.074, +0.130]$ •
off-policy K/V writer	0.090	0.200	64%	$+0.004$ $[-0.020, +0.028]$	$+0.088$ $[+0.060, +0.118]$ •
soft prefix (embeddings)	0.050	0.090	12%	-0.106 $[-0.138, -0.076]$ •	-0.002 $[-0.010, +0.004]$

test. At 64 entries it recovers 12% of the test gap against 62%, a paired -0.106 $[-0.138, -0.076]$ •, and its pairing effect of -0.002 $[-0.010, +0.004]$ shows that what it keeps does not depend on the sender's document. On test it is level with the key–value selection arm the writer starts from at both budgets, -0.006 $[-0.026, +0.014]$ and $+0.000$ $[-0.024, +0.024]$, and on validation it sits at or below it, -0.018 $[-0.038, +0.000]$ and -0.012 $[-0.036, +0.012]$. Wire cost for reference: 32×4096 bfloat16 values, 0.26 MB against 4.7 MB for a full 32-entry key–value message.

C.5 The task-free channel: label-free selection and confirmatory evaluation

The task-free claim requires label-free model selection as well as a label-free objective. We therefore choose checkpoints using only held-out raw text. A checkpoint is eligible when its matched message yields lower continuation NLL than a message written from a different chunk. Among eligible checkpoints we take the lowest matched NLL, one per budget, and if none is eligible the arm counts as unsuccessful. The criterion reuses the training objective on WikiText chunks (Merity et al., 2017) disjoint from training at the article level.

Confirmation uses validation rows unseen in development, rows 500–1999 on HotpotQA and 500–999 elsewhere, each evaluated once. Table 12 reports a primary aggregate of $+0.018$ $[+0.009, +0.026]$. Five of eight secondary cells survive Benjamini–Hochberg, and TriviaQA at K=32 is null.

Table 12: **Task-free channel, confirmatory evaluation** on validation instances never inspected during development (rows 500 and later, n=1500 on HotpotQA and 500 elsewhere), under the label-free checkpoint rule stated in the text. The primary endpoint is the equal-weight mean of the four K=32 written-minus-selection differences, with a joint per-benchmark bootstrap. The eight individual cells are secondary, with Benjamini–Hochberg control at q=0.05.

Budget	Benchmark	n	Written − Sel.	p	BH q=.05
K=32	HotpotQA	1500	$+0.020$ $[+0.009, +0.031]$	0.0001	reject
K=32	2Wiki	500	$+0.018$ $[-0.002, +0.038]$	0.0850	n.s.
K=32	MuSiQue	500	$+0.030$ $[+0.012, +0.048]$	0.0004	reject
K=32	TriviaQA	500	$+0.002$ $[-0.016, +0.020]$	0.9134	n.s.
K=64	HotpotQA	1500	$+0.018$ $[+0.007, +0.029]$	0.0014	reject
K=64	2Wiki	500	$+0.030$ $[+0.010, +0.052]$	0.0050	reject
K=64	MuSiQue	500	-0.002 $[-0.018, +0.014]$	0.7208	n.s.
K=64	TriviaQA	500	$+0.042$ $[+0.022, +0.062]$	0.0001	reject
Primary endpoint: equal-weight mean, K=32			$+0.018$ $[+0.009, +0.026]$	interval excludes zero	

D The multi-document regime

D.1 The random-order channel and the sender's document order

The multi-document builders place the answer-bearing document first and shuffle only the receiver's documents, so the sender's gold paragraph is at index 0 in every released instance: 1303/1303 HotpotQA validation rows, 1257/1257 test rows, 888/888 MuSiQue test rows, and every sampled

training row of both. A channel trained under that layout depends on the cue. The experiment of Figure 10 and Table 13 rewrites only x_a, holding the example identifier, the receiver's document, the question and the metadata byte-identical, so the paired bootstrap runs on the same rows and the identity permutation, asserted byte-identical to the released corpus, serves as the control. It reports each channel with the gold document first, in a per-instance random order, and last, on validation and on the held-out test split. The answer-first channel collapses when its cue moves. Its shuffled margin over selection retains 0.15 of the gold-first value on test against 0.29 on validation, a second training seed retains 0.30, and the MuSiQue channel 0.20 with its reference unchanged. On validation the full-context reference is unchanged by reordering, so only the channel degrades. On test the reference falls by 0.038 $[0.010, 0.068]$ while the channel falls by 0.160.

The cause is a property of the scoring rule. Accumulated attention mass rewards early positions because, under causal masking, position j is attended by only the $T - j$ queries at or after it. Under the pooled-attention rule the fraction of the K selected entries lying inside the gold document's token span runs 0.557 with the gold document first and 0.026 with it last, and the share of senders retaining any gold entry falls from 1.000 to 0.075, so on most instances the message never contained the evidence. Coverage is not itself the mechanism. A selector taking the sink and the final $K - 1$ positions holds only 0.015 of the gold document when it is first, less than pooled attention holds when it is last, yet swapped into the answer-first channel at 32 entries, with nothing retrained, it returns 24.0% of the gap, against 1.5% for that channel's own pooled attention with the gold document last. What separates them is whether the selected positions can *see* the gold document under causal masking. The channel of §4.3 is built on that selector, the only positions that attend to every document, and trained on a randomised corpus. Its retained gap still depends on the layout: at the primary seed it returns 37.7%, 51.0% and 88.8% of the gap across the three layouts at 64 entries (Table 13, with two seeds pooled in Figure 10), the mirror of the answer-first channel's front bias. Each channel of Table 13 was decoded once on held-out test.

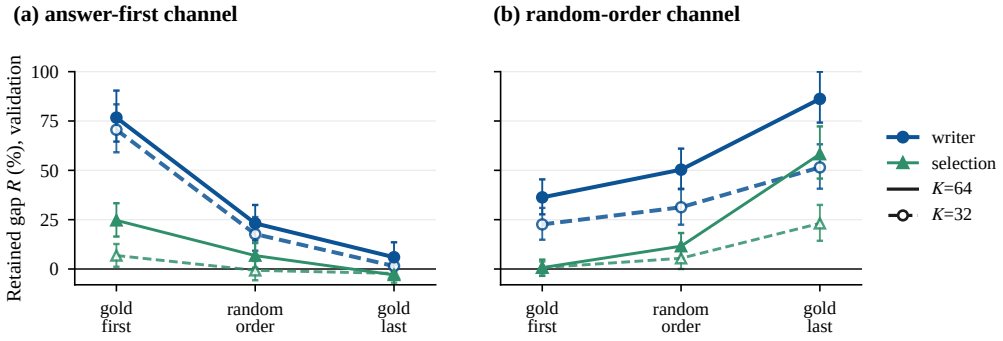


Figure 10: **Where the gold document sits decides what an answer-first channel can do.** Retained gap on validation for **(a)** the channel trained with the gold document first and **(b)** the random-order channel of §4.3, each with the selection arm it is initialised from, as the sender's gold document moves from first to last. Whiskers are paired bootstrap 95% intervals, and the random-order $K{=}64$ points pool two training seeds (Table 13).

Table 13: **The sender's document order.** Each row decodes a channel on the same 500 rows with only x_a reordered. Per cell: exact match, retained gap, and the paired margin over that cell's own selection arm, with • for an interval excluding zero. The answer-first channel uses the pooled-attention selector, and the random-order channel of §4.3 the sink plus the final $K - 1$ positions. The random-order channel is decoded on held-out test in the random order only (Figure 3).

		gold first			random order			gold last		
Channel	Selector	EM	R	Δsel	EM	R	Δsel	EM	R	Δsel
Answer-first channel, $K{=}32$	pooled attention	0.252	70.5%	+0.186 •	0.098	17.7%	+0.054 •	0.050	1.5%	+0.010
same channel, held-out test	pooled attention	0.260	77.8%	+0.174 •	0.100	16.8%	+0.026 •	0.068	1.9%	+0.006
second training seed	pooled attention	0.296	85.6%	+0.230 •	0.112	22.4%	+0.068 •	0.062	6.0%	+0.022 •
MuSiQue channel, held-out test	pooled attention	0.102	59.0%	+0.082 •	0.028	13.6%	+0.016	0.004	0.0%	+0.000
Answer-first channel, $K{=}64$	pooled attention	0.270	76.7%	+0.152 •	0.114	23.1%	+0.048 •	0.062	6.0%	+0.024 •
Random-order channel, $K{=}32$	sink + recent	0.112	22.6%	+0.064 •	0.138	31.3%	+0.076 •	0.184	51.5%	+0.076 •
Random-order channel, $K{=}64$	sink + recent	0.156	37.7%	+0.108 •	0.196	51.0%	+0.116 •	0.284	88.8%	+0.082 •
second training seed	sink + recent	0.148	34.9%	+0.100 •	0.192	49.7%	+0.112 •	0.270	83.6%	+0.068 •

D.2 TEXT NOTES UNDER THE RANDOMISED ORDER

Table 14 decodes the notes of the random-order channel of §4.3 on its own construction: the same Qwen3-8B sender writes its notes on the shuffled files, over the 500 rows the writer is scored on, and the note harness reproduces the writer harness's floor of 0.046 and 0.064 to the digit on all eight records. Table 15 does the same for the transfer cells, where each cell's own model writes the note. The position-matched note collapses with the selector. On test it reaches 0.074 at 32 entries and 0.096 at 64, 4.7% and 15.0% of the gap, the range selection occupies, where writing returns 37.4% and 61.7%. The sampled notes show the mechanism: the sender enumerates its documents in reading order and spends the budget on the first one, which a random order makes a distractor in most instances, so the note is order-sensitive for the same reason the pooled-attention selector is. Writing therefore leads the position-matched note at both budgets on both splits, by $+0.070$ $[+0.038, +0.102]$ and $+0.100$ $[+0.067, +0.134]$ on test and $+0.072$ $[+0.040, +0.104]$ and $+0.098$ $[+0.061, +0.136]$ on validation. The budget-aware prompt of Appendix B.3 does not change that reading (Table 14).

Table 14: **Text notes under the randomised sender order.** The first 500 rows of the shuffled files, the rows and order Table 13 scores the writer on. The same Qwen3-8B sender writes budget-blind and budget-aware notes (Appendix B.3), and one free-length note, decoded without a position budget, is compared against each budget's writer. R uses the cache-form reference, the higher of the two, and the text-form reference is printed above it. CACHEWRITER averages its training seeds per example before pairing at $K{=}64$. • marks a paired bootstrap 95% interval excluding zero.

	validation			held-out test		
Arm	EM	R	CACHEWRITER − arm	EM	R	CACHEWRITER − arm
No message	0.046	0.0%		0.064	0.0%	
Full context, text form	0.322	93.9%		0.274	98.1%	
Full context, cache form	0.340	100.0%		0.278	100.0%	
$K{=}32$ receiver positions						
CACHEWRITER	0.138	31.3%		0.144	37.4%	
Note, budget-blind prompt	0.066	6.8%	$+0.072$ $[+0.040, +0.104]$ •	0.074	4.7%	$+0.070$ $[+0.038, +0.102]$ •
Note, budget-aware prompt	0.076	10.2%	$+0.062$ $[+0.032, +0.094]$ •	0.078	6.5%	$+0.066$ $[+0.036, +0.098]$ •
Free-length note	0.190	49.0%	-0.052 $[-0.092, -0.012]$ •	0.194	60.7%	-0.050 $[-0.090, -0.010]$ •
$K{=}64$ receiver positions						
CACHEWRITER	0.194	50.3%		0.196	61.7%	
Note, budget-blind prompt	0.096	17.0%	$+0.098$ $[+0.061, +0.136]$ •	0.096	15.0%	$+0.100$ $[+0.067, +0.134]$ •
Note, budget-aware prompt	0.092	15.6%	$+0.102$ $[+0.067, +0.138]$ •	0.104	18.7%	$+0.092$ $[+0.062, +0.123]$ •
Free-length note	0.190	49.0%	$+0.004$ $[-0.037, +0.045]$	0.194	60.7%	$+0.002$ $[-0.036, +0.039]$

D.3 TRANSFER ACROSS MODEL FAMILIES AND BENCHMARKS

Every multi-document transfer result below uses the random order of §4.3, and the last paragraph's diagnostic of the text reference uses the answer-first files.

Four transfer cells. Table 15 retrains the channel of §4.3 at both budgets for Llama-3.1-8B, OLMo-2-7B and Falcon3-7B on HotpotQA and for Qwen3-8B on MuSiQue, and decodes it on shuffled held-out test rows. Writing beats its selection arm with an interval excluding zero in four of four cells at 64 entries and four at 32, the update stays paired with its instance in four and four, and retained gaps run from 36% to 69% and from 28% to 61% against the Qwen3 channel's 62% and 37%. On the same rows, the sender's position-matched note stays below writing with an interval excluding zero in four of four cells at 64 entries and four at 32. The text-form reference is higher than the cache-form one in three of the four cells, and against it writing retains 56%, 30% and 34% of the gap at 64 entries for Llama-3.1, OLMo-2 and MuSiQue. OLMo-2's channel was decoded at batch 4 because batch 8 exceeded GPU memory, which moves its floor by two of 500 examples per split. Every arm of that channel shares the batch, and its note, decoded at batch 8 by the text harness, has the same test floor.

Where the text reference puts the sender's document. Full-cache and full-text references differ across model families, and decoding the text-form reference twice on the answer-first multi-document construction, on validation, locates most of that difference in layout. The text form places x_a beside the question at the end of the receiver's prompt, a recency advantage the prepended cache-form message cannot have. Prepending x_a to x_b instead, with the message slot left empty, removes 80% of OLMo-2's excess over its cache reference, 0.330 falling to 0.252 against 0.232, and 63% of Llama-3.1's, 0.434 falling to 0.400 against 0.380. Qwen3 is the control whose two references agree,

Table 15: **The random-order channel of §4.3 on the transfer cells, held-out test.** Each row retrains that recipe for the receiver, benchmark and budget named, seed 0, and decodes the first 500 rows of the shuffled test file. *Sel.* is the selection arm the writer starts from, *Note* the budget-blind note truncated to the same K receiver positions, *deranged* Δ another instance's update on the instance's own entries under one derangement (Appendix E averages eight, e.g. $+0.144$ and $+0.098$ for the Qwen3 channel), and R the retained gap. • marks a paired bootstrap 95% interval excluding zero.

Model / data	Floor	Full KV	Sel.	Note	CacheWriter	CacheWriter − Sel.	CacheWriter − Note	CacheWriter − deranged Δ	R
K=64 receiver positions									
Qwen3 / Hotpot	0.064	0.278	0.090	0.096	**0.196**	+0.106 [+0.076, +0.136] •	+0.100 [+0.066, +0.136] •	+0.148 [+0.114, +0.182] •	62%
Llama-3.1 / Hotpot	0.098	0.352	0.184	0.116	**0.272**	+0.088 [+0.056, +0.122] •	+0.156 [+0.118, +0.194] •	+0.124 [+0.090, +0.160] •	69%
OLMo-2 / Hotpot	0.076	0.210	0.096	0.102	**0.140**	+0.044 [+0.016, +0.072] •	+0.038 [+0.008, +0.068] •	+0.054 [+0.026, +0.082] •	48%
Falcon3 / Hotpot	0.020	0.372	0.096	0.052	**0.262**	+0.166 [+0.130, +0.204] •	+0.210 [+0.170, +0.250] •	+0.174 [+0.138, +0.210] •	69%
Qwen3 / MuSiQue	0.004	0.180	0.010	0.028	**0.068**	+0.058 [+0.036, +0.080] •	+0.040 [+0.016, +0.064] •	+0.058 [+0.036, +0.082] •	36%
K=32 receiver positions									
Qwen3 / Hotpot	0.064	0.278	0.076	0.074	**0.144**	+0.068 [+0.040, +0.096] •	+0.070 [+0.038, +0.102] •	+0.104 [+0.074, +0.134] •	37%
Llama-3.1 / Hotpot	0.098	0.352	0.136	0.114	**0.242**	+0.106 [+0.076, +0.138] •	+0.128 [+0.094, +0.164] •	+0.088 [+0.058, +0.118] •	57%
OLMo-2 / Hotpot	0.076	0.210	0.084	0.096	**0.132**	+0.048 [+0.020, +0.076] •	+0.036 [+0.004, +0.068] •	+0.046 [+0.018, +0.074] •	42%
Falcon3 / Hotpot	0.020	0.372	0.066	0.030	**0.236**	+0.170 [+0.134, +0.206] •	+0.206 [+0.170, +0.244] •	+0.100 [+0.068, +0.132] •	61%
Qwen3 / MuSiQue	0.004	0.180	0.006	0.024	**0.054**	+0.048 [+0.030, +0.068] •	+0.030 [+0.006, +0.054] •	+0.042 [+0.022, +0.062] •	28%

0.346 against 0.342, and the move changes it by -0.020. Falcon3 reverses the pattern, 0.234 against 0.354, and placement does not account for it: moving the document takes its text reference further down, to 0.212, so the reversal belongs to that receiver. Retained gain against the cache reference is the like-for-like measure, with the text-reference figure as the stricter bound.

E Source-instance attribution

This appendix audits answer exclusivity, then reports the derangement battery of §4.4: whole-message and component derangement, the crossed two-by-two, and the document-independent update.

E.1 Answer exclusivity

The construction gives the sender the answer-bearing paragraph and leaves the receiver's documents unconstrained. Table 16 measures the resulting answer leakage and repeats the primary cells on each stratum. The primary definition counts an instance as leaked when the normalised gold answer occurs at a word boundary in the normalised receiver document. It uses the metric normaliser and includes yes/no answers.

Raw substring matching counts 191 of the first 500 HotpotQA validation instances. Word-boundary matching removes 20, including 15 yes/no answers embedded in longer words. All primary results use the word-boundary definition. We call the remainder answer-exclusive.

A broader union criterion combines raw substring matching, sliding-window token F1 of 0.8, and word-boundary matching over every released answer alias. The counted cases rise from 171 to 193 of 500 on HotpotQA validation, from 57 to 81 on 2WikiMultihopQA, from 35 to 69 on MuSiQue, and from 13 to 20 on TriviaQA. Most of the MuSiQue increase comes from aliases omitted by the primary definition. Under the union criterion, the answer-exclusive multi-document stratum loses 31 of 500 validation and 30 of 500 test instances, and on the rest the random-order channel still leads its selection arm on test, by $+0.072$ $[+0.043, +0.104]$ at 32 entries and $+0.113$ $[+0.081, +0.145]$ at 64. The single-document primary margins remain $+0.088$ and $+0.059$ at 32 and 64 entries. The MuSiQue multi-document construction in Appendix D.3 uses the union definition.

E.2 Whole-message and component derangement

Whole-message derangement. In the primary validation cell (Qwen3-8B, HotpotQA-Private, $K = 32$, $n = 500$), the written message scores 0.358. A message from a different instance scores 0.202, close to the 0.196 no-message floor and below the 0.470 full-context reference. Destroying the entries yields 0.000. The useful content is therefore specific to the source document. On test the written, deranged, floor and full-context scores are 0.340, 0.230, 0.180 and 0.418.

Component derangement. Because selector positions are computed from the *unadapted* sender, the message decomposes exactly into a reference and a learned update, $m_i = r_i + \Delta_i$, where r_i is the selection arm itself. Holding r_i fixed and substituting $\Delta_{\pi(i)}$ from another instance in the same evaluation chunk gives 0.246. Writing score$(\cdot)$ for exact match, the writer's gain over its own

Table 16: **Lexical answer exclusivity.** *Top*: the fraction of the first 500 evaluation instances in which the normalised gold answer occurs at a word boundary in the receiver documents. *Bottom*: the primary single-document cells on the lexically leaky and answer-exclusive strata. Intervals are paired bootstrap over 10^4 resamples within the stratum. R is omitted for the leaky stratum, whose floor lies within 0.10 of the full-cache reference, so the ratio is unstable.

Benchmark	val leaky	val %	test leaky	test %
HotpotQA	171/500	34.2%	196/500	39.2%
2Wiki	57/500	11.4%	64/500	12.8%
MuSiQue	35/500	7.0%	9/500	1.8%
TriviaQA	13/500	2.6%	5/500	1.0%

Cell	stratum	n	none / sel. / written / full	R	written − sel.
primary K=32	leaky	171	0.479 / 0.544 / **0.602** / 0.579	—	$+0.059$ $[+0.006, +0.111]$
primary K=32	clean	329	0.049 / 0.149 / **0.231** / 0.413	50.0%	$+0.082$ $[+0.049, +0.116]$
primary K=64	leaky	171	0.479 / 0.573 / **0.626** / 0.579	—	$+0.053$ $[+0.018, +0.094]$
primary K=64	clean	329	0.049 / 0.283 / **0.337** / 0.413	79.2%	$+0.055$ $[+0.021, +0.088]$

reference decomposes exactly:

$$\underbrace{\text{score}(r_i{+}\Delta_i) - \text{score}(r_i)}_{\text{matched-update gain } +0.074\ [+0.046,+0.102]} = \underbrace{\text{score}(r_i{+}\Delta_i) - \text{score}(r_i{+}\Delta_{\pi(i)})}_{\text{pairing } +0.112\ [+0.076,+0.148]} + \underbrace{\text{score}(r_i{+}\Delta_{\pi(i)}) - \text{score}(r_i)}_{\text{mismatched update } -0.038\ [-0.072,-0.004]}.$$

The mismatched-update term is negative and its interval excludes zero. Combining a reference with an update learned for another document reduces performance below the unmodified reference. We call this quantity the mismatched-update effect because it measures that specific intervention.

Diagnostics. Across 32 full-length documents, the pairwise cosine similarity between learned updates is 0.148, showing substantial instance variation. The decomposition above uses a cyclic shift within each evaluation chunk and therefore has no fixed points. Table 17 repeats every arm under random derangements. We match entry counts within each pair, zero-padding a shorter partner update to preserve the reference. All intervals are paired bootstraps, 10^4 resamples of the 500 instances.

E.3 The full two-by-two under random derangements

Figure 11 draws each cell as a two-by-two. The first two cells of Table 17 are single-document, and the last three repeat the decomposition on the randomised channel's held-out test split. The decomposition admits two further arms, a matched update on a deranged reference, $r_{\pi(i)} + \Delta_i$, and both parts deranged, $r_{\pi(i)} + \Delta_{\pi(i)}$, and the table runs all four under random within-chunk derangements, 20 draws on the primary validation cell and 12 on test, with one permutation shared by the three deranged arms. On single documents the mismatched update reduces the message to its selection reference or below, -0.032 $[-0.058, -0.005]$ on validation and -0.011 $[-0.039, +0.016]$ on test, significantly on validation, while the test interval includes zero. A matched update placed on another instance's reference is worse than deranging both parts, by -0.043 $[-0.060, -0.027]$ on validation, so the update is bound to the reference it was written for, and the matched reference retains value on its own, $+0.047$ $[+0.021, +0.074]$ above deranging both, because the reference is the selection arm. On the randomised held-out test cells, at 8 draws each, the mismatched update differs from the selection reference by -0.038 $[-0.061, -0.015]$ and -0.043 $[-0.067, -0.021]$ at 64 entries over two seeds and -0.030 $[-0.054, -0.007]$ at 32, every interval excluding zero, and the matched message stays above the mismatched update by $+0.144$ $[+0.114, +0.175]$, $+0.149$ $[+0.119, +0.181]$ and $+0.098$ $[+0.070, +0.126]$. A matched update on a foreign reference is below deranging both at 64 entries on both seeds, -0.011 $[-0.029, +0.007]$ and -0.036 $[-0.055, -0.017]$, significantly on the second, and the two are level at 32 entries, -0.001 $[-0.020, +0.018]$. The deranged arms range from 0.046, below the no-message floor of 0.064, to 0.106, near the selection reference, at 64 entries on the second seed, and draw-to-draw variation is small: the mismatched-update arm ranges over 0.238–0.278 across the twenty validation permutations.

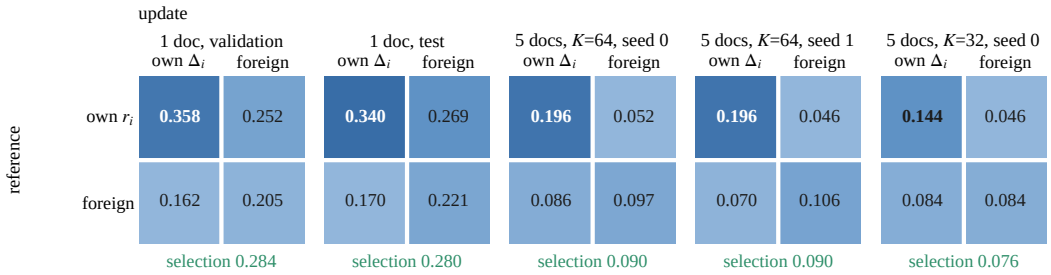


Figure 11: **The update is bound to its reference.** Exact match for each combination of the reference r (rows) and the update Δ (columns), own or taken from another instance, in every cell of Table 17. Shading is the share of the cell's floor-to-full bracket, the matched message is bold, and the selection reference is printed below each block. In every cell the own update on a foreign reference scores at or below deranging both, and the foreign update on the own reference scores below selection.

Table 17: **The full two-by-two under random derangements.** The message is $m_i = r_i + \Delta_i$. *ud* $= r_i + \Delta_{\pi(i)}$ deranges the update, *rd* $= r_{\pi(i)} + \Delta_i$ the reference, and *wm* both, each averaged per instance over P random within-chunk permutations shared by the three arms. Paired bootstrap 95% intervals over n=500 instances. The last three rows are the randomised held-out test cells, at P=8.

Cell	P	sel.	matched	ud	rd	wm	matched − ud	ud − sel.	rd − wm
Single-document, validation	20	0.284	**0.358**	0.252	0.162	0.205	+0.106 [+0.080, +0.131]	−0.032 [−0.058, −0.005]	−0.043 [−0.060, −0.027]
Single-document, test	12	0.280	**0.340**	0.269	0.170	0.221	+0.071 [+0.046, +0.098]	−0.011 [−0.039, +0.016]	−0.051 [−0.067, −0.035]
Randomised multi-document, K=64, seed 0, test	8	0.090	**0.196**	0.052	0.086	0.097	+0.144 [+0.114, +0.175]	−0.038 [−0.061, −0.015]	−0.011 [−0.029, +0.007]
Randomised multi-document, K=64, seed 1, test	8	0.090	**0.196**	0.046	0.070	0.106	+0.149 [+0.119, +0.181]	−0.043 [−0.067, −0.021]	−0.036 [−0.055, −0.017]
Randomised multi-document, K=32, seed 0, test	8	0.076	**0.144**	0.046	0.084	0.084	+0.098 [+0.070, +0.126]	−0.030 [−0.054, −0.007]	−0.001 [−0.020, +0.018]

E.4 A DOCUMENT-INDEPENDENT UPDATE

A deranged update cannot separate an update that carries this document from one that would help any document, because a foreign update is harmful or at best neutral: on validation it differs from the reference it is added to by -0.038 $[-0.072, -0.004]$ at 32 entries on single documents and -0.036 $[-0.064, -0.010]$ on the randomised multi-document channel, and is level with it at 64 entries, -0.002 $[-0.032, +0.026]$. Table 18 supplies the control that can. We replace Δ_i with its per-slot mean over 2 000 training documents, one tensor identical for every instance and computed once, and add it to the same reference r_i, so nothing is computed per instance and the receiver still prefills nothing. Of the writer's gain over selection the mean update carries 93% at 64 entries on single documents and all of it on test, leaving a null residual, $+0.004$ $[-0.012, +0.020]$ and -0.012 $[-0.032, +0.008]$. At 32 entries it carries 57% and 57%, leaving $+0.032$ $[+0.010, +0.054]$ and $+0.026$ $[+0.002, +0.052]$. On the randomised multi-document channel it carries 24% and 32%, leaving $+0.088$ $[+0.060, +0.118]$ and $+0.072$ $[+0.042, +0.102]$. The instance-specific residual therefore grows as the budget binds, across $1.75\times$, $3.5\times$ and $10.5\times$ receiver-position compression, and the pairing effects above bound it from above.

Table 18: **A document-independent update.** The learned update Δ_i is replaced by its per-slot mean over the first 2 000 training documents and added to the same selection reference r_i. The single-document rows are the primary Qwen3-8B HotpotQA-Private records at each budget, seed 0, and the multi-document row is the randomised channel of Table 13, 500 rows per split. The last column is the document-independent share of the gain over selection, a ratio of point estimates without an interval that exceeds 100% where the residual is negative. • marks a paired bootstrap 95% interval excluding zero. Wrong Δ is the mismatched-update arm of Appendix E.

Split	sel.	+mean Δ	+own Δ	mean − sel.	own − mean	mean − wrong Δ	doc.-indep. share
Single document, K=64, 1.75× receiver-position compression							
validation	0.382	0.432	0.436	+0.050 [+0.026, +0.076] •	+0.004 [−0.012, +0.020]	+0.052 [+0.030, +0.076] •	93%
held-out test	0.376	0.430	0.418	+0.054 [+0.028, +0.080] •	−0.012 [−0.032, +0.008]	+0.028 [+0.002, +0.054] •	129%
Single document, K=32, 3.5× receiver-position compression							
validation	0.284	0.326	0.358	+0.042 [+0.016, +0.070] •	+0.032 [+0.010, +0.054] •	+0.080 [+0.050, +0.112] •	57%
held-out test	0.280	0.314	0.340	+0.034 [+0.008, +0.062] •	+0.026 [+0.002, +0.052] •	+0.072 [+0.042, +0.104] •	57%
Randomised multi-document, K=64, 10.5× receiver-position compression							
validation	0.080	0.108	0.196	+0.028 [+0.002, +0.054] •	+0.088 [+0.060, +0.118] •	+0.064 [+0.038, +0.092] •	24%
held-out test	0.090	0.124	0.196	+0.034 [+0.010, +0.060] •	+0.072 [+0.042, +0.102] •	+0.076 [+0.050, +0.104] •	32%

F COMMUNICATION ACROSS MODEL FAMILIES

This appendix gives the cross-family construction, its record, and the controls that make each of its two components necessary.

Channel construction. The message is the sender's own K entries, carried into the receiver's geometry. We strip rotary encoding at their true positions with the sender's inverse frequencies, map them per head through a ridge projection fitted once for the pair (Heo et al., 2026), and re-encode with the receiver's inverse frequencies at slots $0 \dots K-1$. A kept entry's net rotation is then the receiver's own encoding at its destination slot, so the two model families' rotary bases never have to agree. The projection takes $k_{\text{top}}{=}4$ source layers per target layer, holds 302M coefficients at 1.13 GiB, and is fitted once per pair and frozen, with details below. The fit requires the same number of key–value channels per layer in both model families, though not the same head split: every matmul flattens the head axis first, so a target head reads across all source heads. Qwen3-8B, Llama-3.1-8B and Falcon3-7B all carry 1024 channels and are mutually admissible, and OLMo-2-7B at 4096 is not. Layer counts need not match.

To the transported cache we add one tensor of shape $[L, 2, H_{\text{kv}}, K, d_h]$, 9 MB at this budget and identical for every document. It is the per-slot mean, over 2 000 training documents, of a rank-32 correction to the projection trained for 2 500 steps against the frozen receiver's full-context behaviour under the same divergence the same-family writer uses, without gold-answer labels. After training the correction is discarded and only its mean is kept, so the deployed channel runs the frozen projection and adds a constant. The receiver prefills nothing, at training time or at evaluation.

Record. Table 19 gives every arm on both splits. The channel reaches 0.328 on validation and 0.346 on held-out test, beating the sender's position-matched note by $+0.048$ $[+0.010, +0.086]$ and $+0.086$ $[+0.050, +0.122]$ and the transported cache alone by $+0.122$ $[+0.086, +0.158]$ and $+0.130$ $[+0.096, +0.166]$. On held-out test it is -0.072 $[-0.110, -0.034]$ from the receiver's own cache of the document. On the second pair, Qwen3-8B into a frozen Falcon3-7B receiver, the same construction pooled over three training seeds reaches 0.336 on validation and 0.360 on test, beating the note by $+0.064$ $[+0.022, +0.106]$ and $+0.146$ $[+0.103, +0.188]$ and the transported cache alone by $+0.142$ $[+0.108, +0.176]$ and $+0.170$ $[+0.135, +0.207]$. Its seeds span 0.320–0.346 on validation and 0.338–0.384 on test.

Both components are necessary. The transported cache alone is worth $+0.010$ $[-0.008, +0.028]$ over an empty message on validation and $+0.036$ $[+0.016, +0.058]$ on test, and the calibration alone cannot be evaluated, since a constant added to nothing is not a channel. With another document's transported cache the channel falls to 0.234 on test (§4.5), so the transported entries carry the document even though they deliver little of it unaided.

The calibration must be receiver-supervised, and it is direction-specific and slot-locked. Fitting the same constant in closed form, as the per-slot mean displacement from the transported cache to the receiver's own cache of the same document, changes exact match by -0.104 $[-0.138, -0.070]$ on test: it recovers $+0.026$ over the transported cache where the receiver-supervised constant recovers $+0.130$. Moving toward the receiver's own states is the wrong objective, which is what the fitted projection optimises and why it needs correcting at all, and the readout diagnostic agrees: training drives the attention-output cosine between mapped and true states from 0.88 to 0.61 while the per-instance correction gains $+0.104$ $[+0.068, +0.142]$ over the transported cache. That per-instance correction carries no instance: giving each document another document's correction changes exact match by -0.018 $[-0.048, +0.010]$, and a constant averaged over the evaluation documents gains more, $+0.124$ $[+0.090, +0.160]$, all on validation. Two training seeds yield tensors whose cosine is 0.14 and whose norms differ by 17%, disagreeing on 8–9$\%$ of individual answers, yet both reach 0.328 and 0.346, with a paired difference of 0.000 $[-0.024, +0.024]$ on test. Six controls built from the trained calibration and decoded in its place, the channel otherwise unchanged, show what that tensor has to be (Table 20), on held-out test unless stated. A random direction at the calibration's own magnitude destroys the receiver, 0.000 and 0.002 exact match against 0.216 for the transported cache alone (-0.216 $[-0.252, -0.180]$ and -0.214 $[-0.250, -0.178]$), whether that magnitude is matched slot by slot or block by block, so the direction carries the correction. The direction must also sit in its slot: the slot-averaged component, 90% of the norm with no slot-specific content, sits at the transported cache alone on test, -0.010 $[-0.044, +0.024]$, and below it on validation, -0.042

$[-0.078, -0.008]$. The calibration with its slot axis deranged, which keeps every per-slot vector and 78% of the direction, also sits at it, $+0.000$ $[-0.038, +0.038]$. Within that constraint the calibration is distributed: slot 0 alone recovers 49% of the gain over the transported cache with 7% of the norm, $+0.064$ $[+0.036, +0.092]$, the other slots recover 66%, $+0.086$ $[+0.056, +0.118]$, and both reach the position-matched note on test, $+0.020$ $[-0.014, +0.056]$ and $+0.042$ $[+0.008, +0.076]$ (on validation the two shares are 21% and 52%). The shares are ratios of point estimates without intervals, and on test they sum to more than 100% because the gains of the two halves overlap. The note itself scores 0.260 on test. Training finds one of many equivalent calibrations.

Against the same-family channel. Both channels drive the same frozen Qwen3-8B receiver on the same 500 rows at K=32, and neither costs it a prefill, so the two can be paired example by example. The cross-family channel is -0.013 $[-0.042, +0.015]$ from the same-family writer on test and -0.028 $[-0.060, +0.004]$ on validation, against the same-family cell averaged over its five training seeds.

Arm	Val. EM	Test EM
Llama-3.1-8B → Qwen3-8B		
No message	0.196	0.180
Sender's 32-token note (position-matched)	0.280	0.260
Transported cache, K=32	0.206	0.216
+ fixed calibration (*ours*)	**0.328**	**0.346**
+ fixed calibration, wrong document's cache	0.200	0.234
+ closed-form calibration	0.220	0.242
Receiver's own full KV cache	0.470	0.418
Qwen3-8B → Falcon3-7B, pooled over three training seeds		
No message	0.152	0.138
Sender's 32-token note (position-matched)	0.272	0.214
Transported cache, K=32	0.194	0.190
+ fixed calibration (*ours*)	**0.336**	**0.360**
+ fixed calibration, wrong document's cache	0.261	0.291
+ closed-form calibration	0.194	0.186
Receiver's own full KV cache	0.508	0.508

Table 19: **The cross-family channel.** HotpotQA-Private, n=500 per split, K=32. Every channel arm is a key–value prefix on the receiver, so the ceiling is the receiver's own cache of the document, and the text note is the position-matched baseline. The calibration is one tensor shared by every document, and the receiver prefills nothing in any channel row. The second pair averages its three training seeds per example before pairing, and its transported cache, floor and ceiling are identical across seeds. Pairs carry their own floors and ceilings.

Table 20: **What the cross-family calibration has to be.** Six controls, each built from the trained seed-0 calibration and decoded in its place, on validation and held-out test, 500 rows. Cosine and norm share are measured against that calibration. The two random arms keep its magnitude profile at two granularities, *slot mean broadcast* keeps the component common to all slots, *right slot vectors, wrong slots* moves each per-slot vector to another slot, and the last two split its slot axis at the sink. • marks a paired bootstrap 95% interval excluding zero.

			Validation		Held-out test	
Construction	cos to calibration	Norm share	EM	arm − cache alone	EM	arm − cache alone
Transported cache alone, no calibration	—	—	0.206	—	0.216	—
Random direction, per-slot norm matched	+0.001	100.0%	0.000	-0.206 $[-0.242, -0.172]$ •	0.000	-0.216 $[-0.252, -0.180]$ •
Random direction, per-block norm matched	+0.005	100.0%	0.000	-0.206 $[-0.242, -0.172]$ •	0.002	-0.214 $[-0.250, -0.178]$ •
Slot mean broadcast, no slot structure	+0.896	89.6%	0.164	-0.042 $[-0.078, -0.008]$ •	0.206	-0.010 $[-0.044, +0.024]$
Right slot vectors, wrong slots	+0.782	100.0%	0.182	-0.024 $[-0.060, +0.012]$	0.216	$+0.000$ $[-0.038, +0.038]$
Slot 0 only	+0.067	6.7%	0.232	$+0.026$ $[+0.002, +0.050]$ •	0.280	$+0.064$ $[+0.036, +0.092]$ •
Every slot but slot 0	+0.998	99.8%	0.270	$+0.064$ $[+0.032, +0.096]$ •	0.302	$+0.086$ $[+0.056, +0.118]$ •
Trained calibration (*ours*)	+1.000	100.0%	0.328	$+0.122$ $[+0.086, +0.158]$ •	0.346	$+0.130$ $[+0.096, +0.166]$ •

Fitting the projection. We fit the closed-form per-head ridge map of Heo et al. (2026), following their §3, on 147k WikiText tokens (Merity et al., 2017). The pair does not share a tokenizer, so calibration includes only tokens whose character endpoints agree, which covers 97% of cases, and because the receiver never sees x_a the map is fitted on shared text and applied to the sender's private cache. Held-out R^2 reaches 0.71 for keys and 0.56 for values, against the in-sample 0.79 and 0.65 they report for Qwen3-14B→32B, and used directly as a channel the map recovers 12–30% of the gap with the whole mapped cache and 4–15% under the 32-entry budget (Table 21). Their nonlinear variant targets pairs whose out-of-sample ridge R^2 is negative, while this pair already has high linear fidelity, which is why the correction we add is a rank-32 linear one.

Table 21: **A closed-form cross-model KV map across the model-family boundary.** The per-head ridge map of Heo et al. (2026) fit for Llama-3.1-8B → Qwen3-8B on shared WikiText and applied to the sender's private cache, n=500. *Full* transmits the whole mapped cache, and budgeted rows truncate it to K entries with the writer's selection rule. Floors and ceilings are the receiver's own. †From the cell's earlier decode, before a padding fix, where the mapped arm scored 0.214.

Message	Budget	Split	EM	R	wrong pair	mapped − wrong
No message	—	val	0.196	0%	—	—
No message	—	test	0.180	0%	—	—
whole mapped cache (their protocol)	≈112	val	0.230	12%	0.200	+0.030 [+0.006, +0.054]
held-out split	≈112	test	0.252	30%	0.206	+0.046 [+0.020, +0.072]
mapped, document-spanning selection	32	val	0.206	4%	0.196	+0.010 [−0.006, +0.028]
held-out split	32	test	0.216	15%	0.184†	+0.030 [+0.010, +0.050]†
Full private context (ceiling)	≈112	val	0.470	100%	—	—
Full private context (ceiling)	≈112	test	0.418	100%	—	—

G When does cache writing help: headroom and cost

G.1 Headroom by training regime

Across the 171 cells of Figure 2(b), writing gain and selection headroom correlate at Pearson $r = 0.71$ $[0.53, 0.80]$. Both contain the selection score, so permuting the written arm across cells, which keeps that shared term, gives the correlation it alone produces, $r = 0.11$. The partial correlation controlling for budget and training regime is 0.59. Table 22 pools the cells across benchmarks and model families to compare training regimes. The same 171 written-minus-selection cells carry the paper's multiplicity correction: 126 positive and 1 negative cells survive Benjamini–Hochberg at q=0.05, 83 survive two-sided Bonferroni, and the deranged-arm margin survives both in 156.

The ordering holds at both budgets. Under probe selection, task-trained channels recover about three times as much headroom as task-free channels at 32 entries and about ten times as much at 64. The confirmatory cells of the label-free protocol of Appendix C.5 recover about 11% of mean headroom at 32 entries, against 51% for task-trained channels. The task-free channel sees only document continuation, so much of receiver readability is generic to how the frozen model attends to prefixes.

Table 22: **How much of the remaining headroom each training regime recovers.** Headroom is the gap the training-free selection leaves below the full private context. Recovery is the realised gain as a percentage of it. Cells are pooled across benchmarks and model families, so the ordering by regime survives averaging out both.

Training regime	K	Cells	Mean headroom	Mean gain	Recovered
Task-trained	32	83	+0.146	+0.075	51%
Task-trained	48	2	+0.104	+0.051	49%
Task-trained	64	66	+0.063	+0.030	48%
Universal mixture	32	8	+0.143	+0.064	45%
Universal mixture	64	4	+0.077	+0.023	30%
Task-free corpus	32	4	+0.156	+0.028	18%
Task-free corpus	64	4	+0.077	+0.004	5%

G.2 Cost

Producing a written message costs the adapted prefill plus the scoring pass, 0.160 s, which is 1.07× the training-free selection it beats and 5.4× faster than decoding a position-matched 32-token note. Whole-exchange totals are 0.499 s for writing, 1.229 s for the note, and 0.395 s for raw-document forwarding, which remains fastest. Table 23 gives the full breakdown. The same timing probe on 20 five-document senders, 653 entries on average, gives 4.3× against a position-matched note and 5.7 s to decode a free-length one. It includes the pooled-attention scoring pass, which the sink-plus-recent selector of §4.3 does not run, so it overstates that channel's write cost.

Receiver latency spans 68 ms across the message types of Table 23, and the written message decodes 20 ms faster than the full private cache. Answer decoding dominates receiver latency, while bandwidth differs by four orders of magnitude in the opposite direction, so co-located agents pay latency and agents on a constrained network pay transmission cost.

Wire codecs. Figure 12 quantises the primary 32-entry message and evaluates the receiver on the dequantised entries: int8 and int4 lose nothing at 2.0× and 3.9× fewer bytes, and int2 pushes both media below the no-message floor.

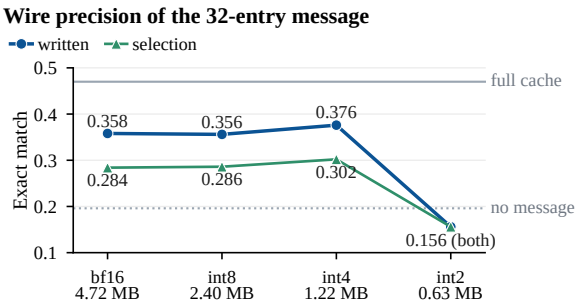


Figure 12: **Wire codecs for the written message.** Primary validation cell, n=500, symmetric absmax quantisation with one fp16 scale per (layer, stream, head, entry) group over head dimension, bytes including the scales for a full 32-entry message (Table 23 gives the probe-sample mean, 4.6 MB), and the selection message quantised identically. The bf16-minus-codec difference is $+0.002$ $[-0.004, +0.010]$ at int8 and -0.018 $[-0.036, -0.002]$ at int4, the written message leads the identically quantised selection message by $+0.070$ $[+0.042, +0.098]$ and $+0.074$ $[+0.046, +0.104]$, and at int2 both fall below the no-message floor.

Table 23: **Cost of one 32-entry exchange**, Qwen3-8B on one 48 GB accelerator, median of five runs after a warm-up over 20 instances averaging 92 entries of private context (validation mean 112).

Stage	Seconds / instance	Peak memory (GB)
Sender		
natural cache prefill	0.0779	15.87
adapted (written) cache prefill	0.0883	16.60
selection scoring pass	0.0720	16.68
text note decode, 32-token budget	0.8579	16.65
text note decode, free length	2.2389	16.65
forward the document verbatim	0.0000	—
Receiver (answer decode)		
no message	0.3895 (0.3889–0.3972)	21.78
32-token note	0.3713 (0.3706–0.3718)	21.84
32 selected entries	0.4063 (0.4058–0.4064)	21.88
32 written entries	0.3385 (0.3383–0.3387)	21.88
full private cache	0.3584 (0.3579–0.3588)	22.21
document forwarded as text	0.3945 (0.3940–0.4016)	21.94
Transmitted payload (bytes) and receiver positions		
32-token note (UTF-8 / token ids)	122 / 128	32
free-length note (UTF-8)	276	—
the private document, forwarded verbatim	350	92
32 cache entries (measured mean)	4,644,864	32
full private cache	13,565,952	92

Transmission and amortisation. The receiver timings are end to end. A measured 32-entry payload crosses NVLink4 in 0.005 ms, PCIe 4 in 0.15 ms, 10 GbE in 3.7 ms, and a 100 Mb wide-area link in 372 ms, so co-located agents and fast interconnects move the message in milliseconds or less. One training run takes about 7.9 L40S-hours and pays for itself against free-length note decoding after about 13,600 single-document or 5,100 five-document exchanges.

H Relation to neighbouring methods

Learned cache compression. KV-Distill compresses a cache for later use by the model that produced it: it selects cache positions, adapts them by a low-rank update, and matches the frozen full-cache model under a mixed forward–reverse divergence (Chari et al., 2025). We reuse its parameter locus and teacher construction for a different reader: a frozen receiver that never sees the source, supervised by its answer distribution given the private document with its own evidence and question.

Cross-model cache mapping. Cache-to-cache fusion (Fu et al., 2026) and the ridge map of Heo et al. (2026) assume sender and receiver read the same input, so neither has a private view. Our cross-family channel builds on that ridge projection. Heo et al. (2026) find that calibration R^2 does not predict downstream retention (Pearson -0.20) while the cosine between attention outputs of mapped and true states does ($+0.57$). In our channel, receiver supervision lowers that cosine while it raises exact match (Appendix F), so neither fidelity measure is the training target.

Latent communication protocols. Naive hidden-state recycling carries no detectable content at a fixed budget. Appending raw sender hidden states to the full private cache costs 0.017 exact match for 32 and 0.127 for 64 on 300 validation instances, because their magnitudes fall outside the receiver's pretraining range. Rescaling them removes the loss without recovering content. Across model families, a random direction at the calibration's norm gives 0.000 exact match (Table 20). A zero-initialised update inside the sender's attention projections starts from entries the receiver already reads.

Table 24: **Where cache writing sits.** Audits: Appendix B.4 and Cheng et al. (2026) for KVComm, Cheng et al. (2026) for Cache-to-cache, Table 21 for the ridge map.

Method	Goal	Message	Frozen reader	Instance attribution
KVComm (Shi et al., 2026)	private handoff	selected natural layers	yes	audited: transmits
Cache-to-cache (Fu et al., 2026)	model-to-model communication	projected + fused cache, same input	fuser trained	audited: no pairing effect detected
Ridge map (Heo et al., 2026)	prefill reuse	mapped cache, same input	yes	audited: content crosses, split-unstable
KV-Distill (Chari et al., 2025)	self-compression	learned compressed cache, instruction uncompressed	yes (self)	n/a
Latent Cache Flow (Rossi et al., 2026)	model-to-model handoff	pooled position-free summary, fused per layer	yes	not assessed
Still (O'Neill et al., 2026)	self-compression	Perceiver-synthesised keys and values	yes (self)	n/a
XKV (Liu et al., 2026)	heterogeneous handoff	dual-cache latent space	yes	not assessed
CACHEWRITER (ours)	private handoff	written entries, zero-initialised at selection	yes	whole-message and component derangement