ASYMCOMM: COST-AWARE COOPERATIVE COMMUNICATION BETWEEN ON-DEVICE AGENTS UNDER INFORMATION ASYMMETRY

Anonymous authors
Paper under double-blind review

ABSTRACT

As artificial intelligence and robotics advance, it is expected that on-device agents will be deployed in growing numbers, and that many situations will require these agents to cooperate. Such agents usually hold disjoint information, because each has observed a different aspect of the same phenomenon (place, event, etc.). The information between them is therefore asymmetric, and they must communicate to act together. We study how two such agents should communicate, through question answering, and introduce the AsymComm benchmark: two agents running the same model hold disjoint evidence, receive one question to solve together, and must reach a consensus answer, while every bit transmitted, every token read or generated and every round of communication is charged with the time it would cost on an edge device. The most direct protocol is context handover, in which one agent transmits its entire memory to the peer. However, we find that handover has a boundary. As the joint context grows, its accuracy declines well before the model's window is full, while its latency grows. We propose protocols for these constraints. Escalate, a protocol that sends ranked paragraphs in growing batches, is twice as accurate as handover at the largest size at a quarter of its latency. The boundary and a deadline on the answer combine into a rule for choosing a protocol. The rule itself forms a switching protocol, Guide, in which the two agents exchange only the sizes of their contexts and then run the protocol the rule selects; Guide obtains the accuracy of the best protocol at every context size, and because the rule is derived from the benchmark's measurements, a new protocol measured on the same axes enters it where it wins. The benchmark and the code that reproduces every experiment are released.

1 INTRODUCTION

Artificial intelligence and robotics have advanced rapidly, and intelligent agents are expected to be deployed in growing numbers across homes, warehouses, factories and roads. Because such an agent must act on the spot under limits of power and connectivity, its model runs on the device itself. Many situations will require these agents to cooperate, and they usually hold disjoint information, because each has observed a different aspect of the same phenomenon (place, event, etc.); the information between them is therefore asymmetric, and they must communicate to act together. Consider two service robots, Alice and Bob, that clean a large villa, Alice the west wing and Bob the east wing, each keeping in its context what it has observed during the day. When they meet in the hall and the owner asks where the guest left her glasses, the evidence is split between them: Alice saw glasses on a desk in the morning, and Bob later carried a tray from that room to the kitchen. Neither knows what the other holds, so they must communicate before they can act, and how they communicate decides whether they answer correctly and how long the owner waits. For tractability, we study this situation on question answering over text, which reproduces its structure and can be iterated quickly.

This exchange differs from a conversation between two chatbots in the cloud (Park et al., 2023; Wu et al., 2024). The model on each agent is small and fixed (Liu et al., 2024c), and it reads text far faster than it generates it, since reading is bound by compute and generation by memory bandwidth (Pope et al., 2022; Leviathan et al., 2023); a protocol that makes an agent compose its message therefore spends the scarcest resource on the device. The agents talk over a wireless link that delivers bits

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.

cheaply but adds a delay to every exchange, and their memory is limited. The situation also offers an asset: co-working agents are very likely the same product running the same released model, so they hold bit-identical weights and each can compute exactly what the other would predict.

Under these conditions the most natural protocol is for Alice to send Bob her entire memory or for the two to swap memories, which we call context handover. It appears difficult to improve upon: sending a context is a copy rather than a generation, the receiver obtains complete information, and because both agents then read the same text with the same weights, they reach the same answer without negotiation. However, handover has a boundary. Figure 1 shows two copies of an 8-billion-parameter model (Yang et al., 2025) swapping contexts of growing size and answering the same questions: accuracy begins to fall at a few thousand tokens, far below the model's 32K window, while the receiver's reading time grows from under a second to sixteen seconds on a processor of the kind current robots carry (NVIDIA, 2025). Earlier work has reported declines with input length and with added distractors (Levy et al., 2024; Liu et al., 2024a; Shi et al., 2023). Two problems are visible: the receiver cannot use what it is given, and reading thousands of tokens costs seconds, while any protocol that makes the agents write to each other costs more.

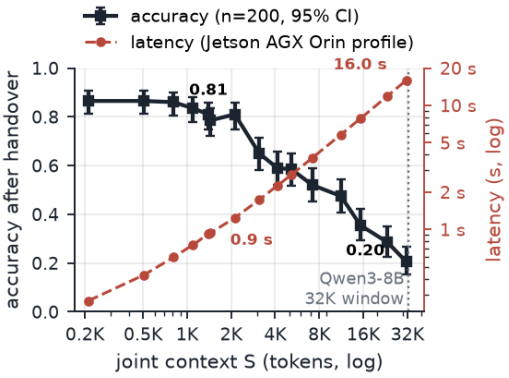


Figure 1: **Context handover has a boundary.** Consensus accuracy (left axis) and critical-path latency on the Jetson AGX Orin profile (NVIDIA, 2025) (right axis) of handover between two Qwen3-8B agents on the same 200 complementary questions at every size (Wilson 95% intervals). The dotted line is Qwen3-8B's native 32K context window (Yang et al., 2025).

The cloud's remedies do not transfer: a larger model is ruled out by power and certification, summarizing and thinking (Snell et al., 2025) are generation, prompt compression (Jiang et al., 2023; Mu et al., 2023) still runs a model over the whole context, and retrieval (Lewis et al., 2020) is not a substitute for communication but the communication problem itself, since the memory to retrieve from is on the other agent. The scarce resource is therefore the receiver's computation, not the wire, and the right protocol depends on how large the context is, how much of it the task needs and how long the user will wait.

What related work leaves out. Multi-agent frameworks let models exchange messages as if messages were free and the division of information were known (Wu et al., 2024; Cadet et al., 2025); long-context studies measure how much context one model can use, not how two models should share it (Liu et al., 2024a; Hsieh et al., 2024); and model-driven coding between devices has been proposed but not built (Ercetin & Chraiti, 2026). None of them charges a protocol for what it makes an on-device agent read, write and wait for, which is the quantity that decides the question above.

In this paper, we consider the two-agent case of this problem and make the following contributions.

- **The AsymComm benchmark.** Two agents hold disjoint evidence from multi-hop question-answering data (Yang et al., 2018; Trivedi et al., 2022), must agree on an answer, and are charged for every bit, token and round with the time it would cost on an edge device; the same questions appear at every context size and evidence density, so every comparison is paired and any protocol can be plotted on the same axes.
- **The boundary of handover.** It consists of two context size thresholds, one for latency set by the device and one for accuracy set by the model.
- **Protocols for these constraints.** Escalate sends ranked paragraphs in growing batches until both answers stop changing; it is twice as accurate as handover at the largest size at a quarter of the latency, and holds its accuracy as the evidence density varies.
- **A HUSH codec.** A lossless codec that cuts the bits on the wire several-fold.
- **Guide.** The thresholds and the deadline combine into a rule that the agents apply after exchanging only their context sizes; it attains the best protocol at every size, and protocols measured on the benchmark later enter the rule where they win.

2 Related Work

Communication between LLM agents. CAMEL (Li et al., 2023), AutoGen (Wu et al., 2024) and MetaGPT (Hong et al., 2024) let models cooperate by exchanging messages, and multiagent debate (Du et al., 2024) improves answers similarly, without accounting for what a message costs on a device; Omidvar & Akhlaghi (2026) assign a cost to the reliability of such messages, and learned protocols in multi-agent reinforcement learning (Foerster et al., 2016) optimize what to send with a trainable model, whereas our agents' model is fixed. AsymPuzl (Cadet et al., 2025) and iAgents (Liu et al., 2024b) address information asymmetry with free messages and a known division of information, so sharing everything is their best strategy, which we show to have a boundary.

Long-context degradation. The decline of accuracy with longer inputs and added distractors is well established (Levy et al., 2024; Liu et al., 2024a; Shi et al., 2023), RULER (Hsieh et al., 2024) and LongBench (Bai et al., 2024) measure how much context a single model can use, and attention sinks (Xiao et al., 2024) handle unbounded streams without making more context usable. They treat the phenomenon as a capability to be measured; we treat it as a communication cost to be allocated.

Language models as compressors. Delétang et al. (2024), LLMZip (Valmeekam et al., 2023) and Kunde et al. (2026) drive an arithmetic coder (Witten et al., 1987) with a language model on one machine or over a channel, realizing the source-coding bound of Shannon (1948); Ercetin & Chraiti (2026) propose predictive-state communication between two devices holding the same predictor as a theory without an implementation, and PMATIC (Adler & Tang, 2026) handles probability mismatch between the two devices' predictors. HUSH is, to our knowledge, the first measured implementation of model-driven coding between two devices that conditions on their common context and charges the decoding; in the terms of Slepian & Wolf (1973) and Wyner & Ziv (1976), the shared context is side information at the decoder.

Communication theory. Semantic communication (Xie et al., 2021) optimizes bits on the channel, which are cheap here; event-triggered estimation (Trimpe & D'Andrea, 2014) transmits only when the receiver's prediction can no longer be trusted, which Escalate's stopping rule extends to the belief of a language model.

3 System Model and Problem Formulation

3.1 Agents, shards, information topology and information profile

Two agents A and B hold private shards X_A and X_B of a document set and receive one question Q; one question with its two shards is an *episode*. Where the evidence E lies relative to the shards defines the *information topology*: **solo** ($E \subseteq X_A$), **complementary** (E split between the shards), **redundant** (E in both, so the correct amount of communication is zero) and **null** (E in neither, so the correct joint answer is *unknown*). The agents are never told the topology, so no agent can decide locally whether its own information is enough. Whether an episode realizes its topology is verified rather than assumed: for every episode we measure, with the same model, the correctness of the answer with no context, with X_A only, with X_B only and with the union, the four indicators $(p_\varnothing, p_A, p_B, p_{AB})$, and this *information profile* must match the topology (complementary is $(0, 0, 0, 1)$) and the question must not be memorized ($p_\varnothing = 0$).

3.2 Device cost model, task and meter

Each agent runs on an edge device with a decode rate r_d and a prefill rate r_p (tokens generated and read per second), with $r_p \gg r_d$ because decoding is bound by memory bandwidth and prefill by compute (Pope et al., 2022); the devices share a wireless link with a given bandwidth and round-trip time, and both run the same weights. Costs are converted to time analytically, that is, from published rates of the device rather than from measurements on hardware. We instantiate the model with two device profiles defined in this paper: the *Jetson AGX Orin profile*, the module NVIDIA positions for next-generation robots (NVIDIA, 2025), and the *Jetson Orin NX profile*, the smaller module of drone mission computers (Neousys Technology, 2025). The first profile's constants are rounded from published measurements (NVIDIA, 2024; Arya & Simmhan, 2025) and the second's are scaled from them (Appendix A).

An episode is scored by *hard consensus*, both answers correct and identical after normalization, because the two agents can take only one action. Under the *clock rule*, an agent's own shard, the question and its instructions are prefilled before the clock starts, since they are its own perception, and everything after the encounter is charged. The *meter* records bits in each direction, tokens decoded and prefilled by each agent and rounds, and the profile converts them into a *critical-path* latency in which the devices work in parallel within a round and rounds are serialized (Appendix A).

3.3 Problem statement

Given two agents with disjoint shards under an unknown topology, a device profile and a deadline D, the problem is to choose the protocol that maximizes the probability of a correct consensus answer within D, and to characterize how the choice depends on the joint context size S, on the *evidence density* ρ, that is, the fraction of the context the task needs, and on D. S and D are observable before communicating and ρ is not, so ρ enters as a condition under which the choice must remain valid rather than as a variable the rule can consult.

4 The AsymComm Benchmark

Data, deal and the blind topology mixture. The questions come from HotpotQA (Yang et al., 2018) in its distractor setting (ten titled paragraphs, two supporting) and, for a second realization of the evidence-density axis, from MuSiQue (Trivedi et al., 2022). A deterministic $\texttt{deal}(\text{item}, \text{topology})$ takes a question with its ten paragraphs (an item) and assigns the supporting paragraphs to the two shards according to the topology and alternates the distractors. Because a deal only decides where the evidence is placed, every dealt episode is profiled again and kept only if its information profile matches its topology exactly (Section 3), which we call the strict filter. The *blind topology mixture* draws a fixed number of episodes of each topology without telling the agents which they face, since only a blind test can reveal whether a protocol handles sufficiency.

Scale axis. The same 200 strict complementary episodes are presented at eleven joint sizes from the natural size of the questions (ten original paragraphs, about 1.4K tokens) to the model's window, by dealing paragraphs from a disjoint pool evenly to both shards and shuffling them so the supporting paragraphs sit at random depth. Since the question, the answer and the supporting paragraphs are fixed, every comparison across sizes and protocols is paired by episode, which lets the thresholds of Section 6 carry confidence intervals. Sizes below the natural one remove the item's distractors.

Evidence-density axis. On the scale axis every episode needs exactly two paragraphs, so the evidence density falls automatically as S grows and a protocol that sends about two paragraphs per side is matched to the data by construction. To separate the amount of context from the fraction of it that is needed, $k \in \{1, 2, 4, 8\}$ questions share one pair of shards at a fixed size; the agents answer all k and are scored per question, and a *blind* set hides k from them.

Determinism and statistics. Model-driven coding requires the encoder and decoder, which run as separate processes, to traverse bit-identical computations, so the benchmark ships a determinism harness and treats any cross-process mismatch as a measurement to be reported (Appendix I). Accuracies carry Wilson 95% intervals (Wilson, 1927); differences between protocols are paired by episode with a McNemar-style standard error (McNemar, 1947); the threshold S^* at which handover's advantage over a protocol reaches zero is the first zero crossing of an isotonic fit to the paired difference, with a bootstrap interval over episodes.

5 Communication Protocols

In this section, we describe the protocols, the rule that chooses among them, and the codec that can be attached to any of them. Figure 2 walks one question through the protocols.

5.1 From sending everything to sending what the answer needs

Every protocol answers one question: how much evidence crosses the wire, and who decides. Since the scarce resources are the receiver's prefill, the sender's decode and the rounds (Section 3),

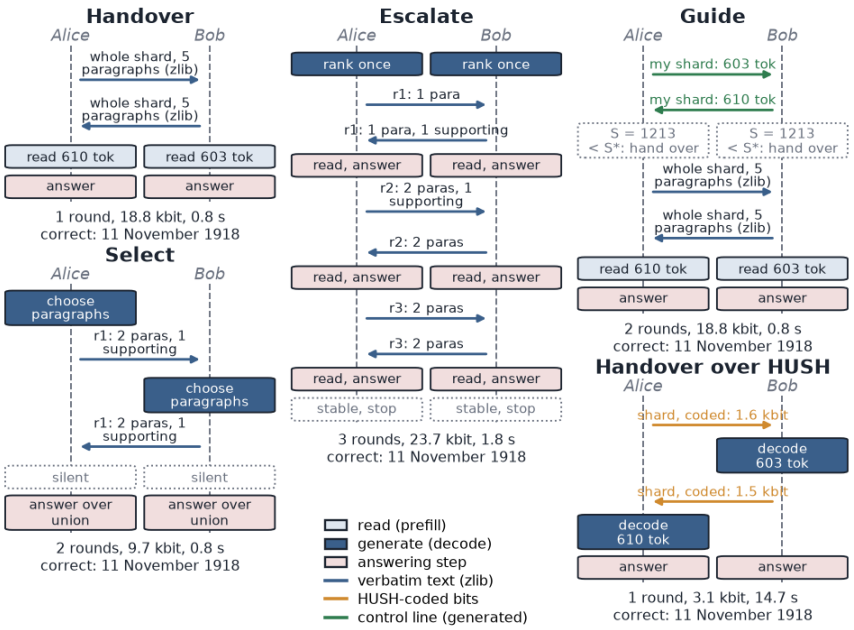


Figure 2: **One question, five protocols.** A real complementary episode with five paragraphs per shard, one of them supporting. For every protocol Alice's lifeline is on the left, Bob's on the right, and time flows downward; boxes are what each agent's model does (read, generate, answer), the arrows between the lifelines are the messages, and the metered totals are below. Guide first exchanges the two shard sizes and, because the joint size is below the threshold, hands over. Read-aloud, conversation, Request and Handshake on the same episode are shown in Appendix F (Figure 7).

a protocol should move content by prefill and never by decode, so that the model generates only decisions, and its stopping condition must be a signal the receiver can produce. The protocols below differ in how much they send and in who decides when to stop, from everything and nobody to as much as the receiver's answer needs.

Handover sends everything and decides nothing. *Context handover* (B0) swaps the shards as zlib-compressed text, that is, the original text under a general-purpose lossless compressor, which is also the default wire of every protocol below; it is the reference every protocol is compared with, since the receiver obtains the complete context. Two baselines send the same or similar content by generation instead of copying, and show what decode costs: *read-aloud* (B1) has the model generate the whole shard, and *conversation* (B2) negotiates in natural language until the agents agree, taking turns with one message per round.

Select sends a fixed amount, chosen by the sender. Each round, each agent generates a list of at most two paragraph numbers and the selected paragraphs are sent verbatim, with an optional line stating what it still needs; each agent must send one paragraph in the first round, afterwards an agent with nothing new stays silent, and the exchange ends when both are silent or after three rounds. Both then answer over the same canonical union. Because paragraphs per round and rounds are capped, the amount sent does not depend on how much the task needs, which Appendix D tests.

Escalate sends as much as the receiver's answer needs. Each agent ranks its paragraphs once, with a single short generation, and sends the next 1, 2, 4, 8 and 16 paragraphs of its ranking in successive rounds. After every round both agents answer from the paragraphs exchanged so far, and the protocol stops as soon as neither answer has changed, or after five rounds. The stopping signal is the stability of the receiver's own answer, which every receiver can produce, so Escalate needs to know neither the size of the context nor how much of it the task needs. We also measured two receiver-driven designs, *Request*, in which each agent asks the peer for what it believes it is missing, and *Handshake*, which first checks with a local answer and a short probe whether any evidence needs to move and escalates only when it does; both are dominated by Escalate except where nothing needs to move (Appendices C, D and E).

5.2 GUIDE: THE BENCHMARK'S MEASUREMENTS AS A RULE

Guide is not a new protocol but the choice among protocols, made by the agents themselves from what the benchmark measured. Section 6.2 shows that the boundary of handover has two thresholds, one set by the device's reading-to-writing ratio and one by the model's long-context behavior, and that beyond it the deadline selects between Escalate and Select. Guide applies that rule online. Before communicating, the two agents exchange the sizes of their shards, one 32-bit integer each way in one round trip, and compute the joint size S; each then runs handover if S is below both thresholds and the deadline admits it, Escalate otherwise, and Select when the deadline does not admit Escalate's rounds. Because the thresholds are derived from the benchmark's measurements, a new protocol measured on the same axes enters the rule at the sizes and deadlines where it wins, so Guide extends as the benchmark is reused.

5.3 HUSH: A CODEC FOR THE WIRE

The codec is a separate dimension of design: it changes how many bits each transmitted token costs, not which paragraphs are sent. Because the agents hold the same weights, the paragraphs a protocol sends can be arithmetic-coded (Witten et al., 1987) against the model's next-token distribution conditioned on the shared context, so that predictable tokens cost a fraction of a bit and the receiver reconstructs the text bit-exactly. We call this codec HUSH, for Hashed-weight Unified Source coding for Handover, since the identical weights act as the shared source model. Its price is that the receiver decodes token by token; a chunked mode decodes several segments in one batched pass. The codec can run handover on its own or be attached to any protocol (Select+HUSH, Escalate+HUSH); it saves bits rather than time, and Section 6.3 measures it separately.

6 EVALUATION

In this section, we evaluate the protocols on the AsymComm benchmark. **Setup.** Both agents run Qwen3-8B (Yang et al., 2025) in fp16 with greedy decoding and identical weights on RTX 6000, A100 and H100 GPUs; the GPU and precision do not affect the results (Appendix H). Latencies use the Jetson AGX Orin profile; the Jetson Orin NX profile gives the same orderings at about 2.4 times the latency (Appendix F). Sample sizes are given in each figure and table. Accuracy is consensus accuracy on strict sets.

6.1 THE BOUNDARY OF CONTEXT HANDOVER

Handover's accuracy falls and its latency grows with the context. Figure 1 shows the phenomenon: handover's accuracy is flat while the joint context is small, then declines steadily, while its latency, the receiver's reading of the peer shard, grows in proportion to the context. The decline is a dilution of the relevant paragraphs among many rather than an effect of where the evidence sits, since moving them to the edges of each shard recovers only part of the loss (Appendices H and G), and it follows the amount of context rather than the fraction that is needed (Appendix D).

Beyond the boundary Escalate is the most accurate protocol, and its decline is the decline of its ranking step. Figure 3 places the protocols on the same axis (full numbers in Table 4), and the four curves have four shapes. Handover starts highest and falls fastest. Select is flat in latency and declines later than handover in accuracy, from a ceiling well below handover's at small sizes. Escalate starts close to handover and declines far more slowly, because it keeps sending evidence until both answers stop changing, so what limits it is the model's ability to rank rather than the size of the context the receiver must read. Conversation is low everywhere. An ablation that keeps the doubling schedule and the stopping rule and changes only the order in which paragraphs are sent confirms this (Table 1): with an oracle order that sends the supporting paragraphs first, Escalate is as accurate at the largest size as handover is at the natural size, in fewer rounds and under two seconds, and letting the model choose each round in view of what it has received recovers about half of the gap between one-shot ranking and the oracle. The receiver can therefore use a well-ordered subset even at 32K tokens, and the whole decline of Escalate with context is the decline of the ranking step.

In numbers, below about 2.5K tokens handover is both the most accurate protocol and the fastest. Handover's accuracy advantage over Escalate reaches zero at $S^* = 2{,}453$ tokens (bootstrap inter-

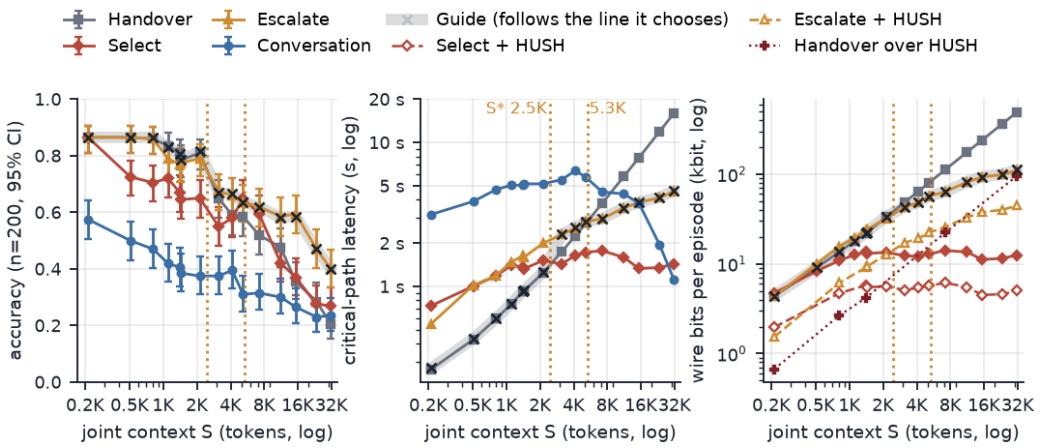


Figure 3: **Protocols on the scale axis** (n=200 per point, the same episodes at every size, Jetson AGX Orin profile). Left: consensus accuracy with Wilson intervals of handover (grey), Select (red), Escalate (orange) and conversation (blue). Middle: critical-path latency. Right: wire bits per episode, with HUSH attached to Select and Escalate as dashed lines and handover over HUSH as the dotted line. Guide, the rule of Section 6.2 applied online by the agents, is drawn in all three panels as a grey halo with cross markers: it follows handover up to the accuracy crossover and Escalate beyond it, and the line it follows stays visible underneath. The dotted vertical lines mark the accuracy crossover against Escalate (S^*, 2.5K) and the latency crossover of Escalate (5.3K), both interpolated between neighboring grid points; the points below 1.4K are the sub-natural sizes of Table 2, where the HUSH variants are measured at two sizes with n=100 for Select and Escalate and n=20 for handover.

	S=7.2K				S=31.6K			
Order of paragraphs	acc	latency	kbit	rounds	acc	latency	kbit	rounds
Oracle (supporting paragraphs first)	.845	1.4 s	31	2.4	.83	1.8 s	42	2.5
Model, chosen each round	**.75**	2.9 s	49	3.0	**.60**	4.3 s	84	3.5
Model, ranked once (Escalate)	.62	3.0 s	64	3.3	.40	4.6 s	111	3.8
Lexical overlap with the question	.57	2.8 s	76	3.6	.445	4.0 s	116	3.9
Handover (for reference)	.52	3.8 s	112	1	.20	16.0 s	493	1

Table 1: **Ranking ablation for Escalate** (n=200, Jetson AGX Orin profile). The doubling schedule and the stability stop are unchanged; only the order in which each agent sends its paragraphs differs. The oracle order uses the gold labels and is an upper bound, not a protocol; bold marks the best of the orders an agent can compute.

val 1,421 to 3,722), and from 7.2K on handover is less accurate at every grid point with paired intervals that exclude zero; Escalate's latency crossover is marked in Figure 3. Against Select and conversation the advantage lasts longer (Figure 3), so S^* is always quoted with its protocol.

Figure 4 shows where the time goes (per-agent token counts in Table 6): in handover nearly all of it is the receiver reading; read-aloud, which has the sender write the same content, would take about twenty times longer; conversation's time is almost entirely writing, and its accuracy is the lowest; Escalate spends a short ranking step and a few rounds of reading a few paragraphs, and has the highest accuracy.

6.2 Thresholds and the decision rule

The two thresholds have different sources. The latency threshold comes from the hardware: handover's latency is the receiver's prefill of the peer shard, Select's latency is a fixed amount of decode and prefill, and the size at which they are equal depends only on the device's prefill-to-decode ratio. The accuracy threshold comes from the model, because it is where handover's long-context decline crosses the protocol's curve. Handover decodes the 6 tokens of its answer and prefills half of S;

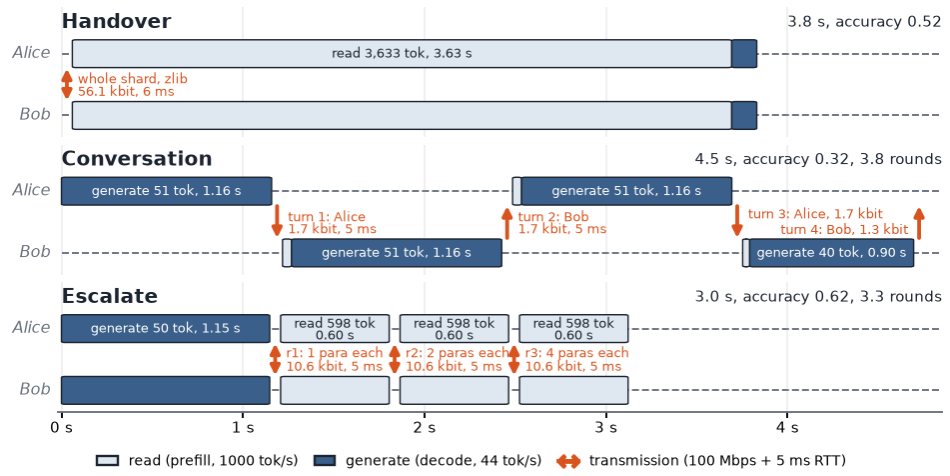
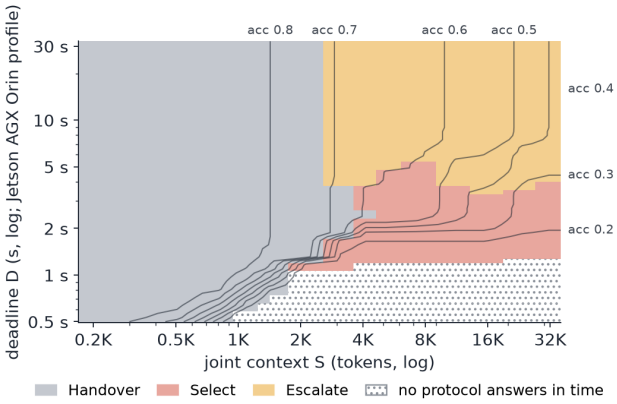
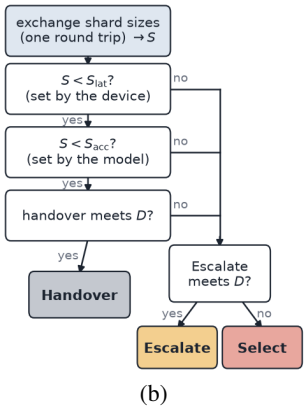


Figure 4: **Where the time goes.** Space-time diagrams of the average episode at 7.2K tokens under the Jetson AGX Orin profile, in the notation of Figure 2 but with time running rightward and the boxes drawn to scale in time: reading (prefill) and writing (decode) sit on each agent's lane and messages cross between the lanes. Handover and Escalate act in parallel, so only Alice's boxes are labeled; conversation alternates turns, so its generations are serial. The token counts are per-round means, scaled by a few percent so that each drawn total equals the metered mean latency printed at the top right of its panel, together with the accuracy and the mean number of rounds. Read-aloud is not drawn: it would take 85 seconds on this axis.



(a)



(b)

Figure 5: **From measurements to Guide.** (a) Best protocol by accuracy under a deadline, over context size S and deadline D on the Jetson AGX Orin profile. For each deadline D and protocol, the score at a size is the fraction of the 200 episodes answered correctly with latency at most D; along S, over the fifteen sizes including the four below 1.4K, each protocol's scores are replaced by their isotonic decreasing fit, the pool-adjacent-violators average that never increases with S and also locates S^*. The colour of a cell is the protocol with the highest fitted score, ties going to the lower mean latency; dotted cells are those whose best score is below 0.10, where no protocol answers in time; contours show the best fitted score. (b) The rule that Guide applies online: the agents exchange shard sizes, compare S with the two thresholds, and let the deadline decide the rest.

Select decodes about 50 and prefills about 390 whatever S is. Equating the two gives the latency threshold in closed form,

$$S_{\text{lat}} \;=\; 2\left(44\,\frac{r_p}{r_d} + 390\right), \tag{1}$$

2.8K tokens on the Jetson AGX Orin profile and 2.7K on the Jetson Orin NX profile; with the published prefill rate of 633 tok/s (NVIDIA, 2024) instead of our rounded 1000 it would be 2.0K.

The deadline turns the two thresholds into a rule. A deployed system optimizes the fraction of episodes answered correctly within its deadline, which Figure 5 evaluates for every protocol, size and deadline. The chart has three regions: below about 2.5K tokens handover is best at every deadline it can meet; above it Escalate is best whenever the deadline allows its rounds, about 4 to 5 seconds at every size; under tighter deadlines, down to about one second, only Select answers in time. With Equation 1 this reduces to the rule of Figure 5(b): hand over when S is below both thresholds and the deadline admits handover's prefill, otherwise escalate, and fall back to Select only when the deadline does not admit the escalation's rounds.

Guide attains the best protocol at every size. Guide applies this rule online, and its curve in Figure 3 lies on the upper envelope of handover and Escalate at every size: handover's 0.81 up to 2.1K tokens, where it hands over, and Escalate's accuracy and latency from 3.1K on, where it escalates, within 0.01 of the protocol it chooses.

Robustness to evidence density. Every episode on the scale axis needs two paragraphs, so a protocol that sends about two per side is matched to the data. We therefore repeated the comparison at fixed sizes with $k \in \{1, 2, 4, 8\}$ questions sharing one pair of shards, so that the fraction of needed evidence varies while the context does not (Appendix D). Handover is invariant to the density, so its decline follows the amount of context; Select and the request-based designs collapse as the need grows, because their volume is capped; and Escalate, which is not told k, matches or exceeds handover at every hidden density at the cost of about one round per doubling of the need.

Blind topology mixture. The axes above are complementary topologies throughout. On a blind mixture that hides solo, redundant and null episodes among them, Escalate stays accurate everywhere but pays its ranking step and first rounds even where nothing needs to move; Handshake removes that cost on those topologies and matches Escalate elsewhere, so it is a router in front of Escalate for mixed topologies (Appendix E).

6.3 The codec, measured separately

The codec saves bits and not time, and pays for the bits with decoding. Attaching HUSH changes nothing about which paragraphs are sent, so Select+HUSH and Escalate+HUSH have exactly the accuracy of Select and Escalate at every size. The right panel of Figure 3 shows their wire bits falling by a factor of two to three, and Figure 8 in Appendix F shows their latency rising by a similar factor, because the receiver now generates what it used to read; at the largest size Escalate+HUSH takes four times as long as Escalate. On a whole shard the codec reaches about three bits per token, a fifth of zlib's, and every two-process round trip was bit-exact (Table 10); the chunked mode decodes several times faster for a moderate increase in bits.

7 Conclusion

In this paper, we consider two on-device agents that share a model but not a view of the world, and that face an asymmetry of information when they must act together. The AsymComm benchmark measures what each way of communicating costs them, and the rule it yields, Guide, lets the agents choose the best protocol from the sizes of their contexts alone. Our benchmark and experiments also yield four findings. Handover has a boundary with two sources: the device sets where it stops being the fastest and the model sets where it stops being the most accurate. The cost of communication is the receiver's reading and the sender's writing, not the wire, so a protocol should move evidence by copying and generate only decisions. The right stopping signal is the receiver's own answer: sending until it stops changing needs neither the size of the context nor the fraction of it that is needed. The remaining accuracy lies in the ranking step, since a receiver that reads a well-ordered subset answers at the largest context as it does at the normal size. The benchmark and the code that reproduces every experiment are released.

Limitations. The boundary is charted with one model family at one size; the accuracy threshold will move with the model, the latency threshold will not. All latencies are computed from published device rates by declared formulas, and nothing is measured on a physical device. Two agents are the basic unit of cooperation rather than its whole.

AI USE STATEMENT

In this work, we used generative AI tools for the following tasks with required disclosure: providing feedback on research methodology and experiments, implementing methods, cleaning and reformatting datasets, helping to formulate a mathematical claim (the closed-form latency threshold), assisting with translation, interpreting results, and producing the figures (the plotting scripts and the schematic and flowchart drawings were generated with AI assistance from the measured records). We have not used generative AI tools to generate synthetic datasets, to develop theoretical models or conceptual frameworks, to prove mathematical claims or to write proofs, or to propose hypotheses, and qualitative or thematic data analysis is not applicable to this work. Additionally, we used generative AI tools for the following tasks with recommended disclosure: creating and editing software code, creating scientific figures, drafting and editing parts of the paper, suggesting the structure of the paper, identifying and summarizing related literature, formatting references, and proposing titles and keywords. We have reviewed all AI-assisted work. The research questions, the outline of every section and every design decision were set by the authors; AI-generated code was verified by tests and dry runs before any experiment was run; every number in the paper was regenerated from the experiment records; every reference was checked against its source and every statement about a cited paper was compared with that paper; and every paragraph of the text was read and revised by the authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

The benchmark is built from publicly released question-answering datasets, HotpotQA (Yang et al., 2018) and MuSiQue (Trivedi et al., 2022), under their original licenses, and involves no human subjects; what we release is a generator that reproduces the episodes from those sources rather than a redistribution of the data. The device profiles are computed from published device rates and no experiment involves a physical robot. We see no direct path from the results to harmful applications beyond those of the underlying language model; the protocols we recommend transmit less than handover does, which reduces rather than increases the exposure of an agent's context on the wire.

REPRODUCIBILITY STATEMENT

Every number in this paper can be regenerated from the released artifacts: the episode generator with fixed seeds, which produces the deals, the information profiles, the blind mixture and the scale and evidence-density sets (Appendix B); one JSONL record per run and episode with the full cost vector and transcript; the metering module and the device profiles with their constants and sources (Appendix A); the figure and table scripts, which read only the records; and the determinism harness with its two-process bit-exactness tests, together with the two hazards that broke determinism during this work (Appendix I). Appendix I also maps the protocol names used in the paper to the identifiers used in the code.

REFERENCES

Aviv Adler and Jennifer Tang. Synchronizing probabilities in model-driven lossless compression, 2026. URL https://arxiv.org/abs/2601.10678. International Conference on Learning Representations (ICLR) 2026; OpenReview nGzYkW4FSB.

Mayank Arya and Yogesh Simmhan. Understanding the performance and power of llm inferencing on edge accelerators, 2025. URL https://arxiv.org/abs/2506.09554.

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. LongBench: A bilingual, multitask benchmark for long context understanding. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3119–3137, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.172. URL https://aclanthology.org/2024.acl-long.172/.

Xavier Cadet, Edward Koh, and Peter Chin. Asympuzl: An asymmetric puzzle for multi-agent cooperation, 2025. URL https://arxiv.org/abs/2512.03466. NeurIPS 2025 Workshop on Multi-Turn Interactions in LLMs; OpenReview SXcJh8hoLz.

Grégoire Delétang, Anian Ruoss, Paul-Ambroise Duquenne, Elliot Catt, Tim Genewein, Christopher Mattern, Jordi Grau-Moya, Li Kevin Wenliang, Matthew Aitchison, Laurent Orseau, Marcus Hutter, and Joel Veness. Language modeling is compression. In *International Conference on Learning Representations*, pp. 14165–14181, 2024.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. Improving factuality and reasoning in language models through multiagent debate. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 11733–11763. PMLR, 2024.

Ozgur Ercetin and Mohaned Chraiti. Predictive-state communication: Innovation coding and reconciliation under delay, 2026. URL https://arxiv.org/abs/2602.10542.

Jakob Foerster, Ioannis Alexandros Assael, Nando de Freitas, and Shimon Whiteson. Learning to communicate with deep multi-agent reinforcement learning. In *Advances in Neural Information Processing Systems 29*, 2016.

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*, pp. 23247–23275, 2024.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, Yang Zhang, and Boris Ginsburg. RULER: What's the real context size of your long-context language models? In *First Conference on Language Modeling*, 2024.

Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. LLMLingua: Compressing prompts for accelerated inference of large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 13358–13376. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.emnlp-main.825.

Vishnu Teja Kunde, Jean-Francois Chamberland, Krishna R. Narayanan, and Jamison Ebert. Real-time text transmission via llm-based entropy coding over fixed-rate channels, 2026. URL https://arxiv.org/abs/2605.01991.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with PagedAttention, 2023. URL https://arxiv.org/abs/2309.06180.

Yaniv Leviathan, Matan Kalman, and Yossi Matias. Fast inference from transformers via speculative decoding. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 19274–19286. PMLR, 2023.

Mosh Levy, Alon Jacoby, and Yoav Goldberg. Same task, more tokens: the impact of input length on the reasoning performance of large language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15339–15353, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.818. URL https://aclanthology.org/2024.acl-long.818/.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems 33*, pp. 9459–9474, 2020.

Guohao Li, Hasan Abed Al Kader Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. CAMEL: Communicative agents for "mind" exploration of large language model society. In *Advances in Neural Information Processing Systems 36*, pp. 51991–52008, 2023. doi: 10.52202/075280-2264.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173, 2024a. doi: 10.1162/tacl_a_00638. URL https://aclanthology.org/2024.tacl-1.9/.

Wei Liu, Chenxi Wang, Yifei Wang, Zihao Xie, Rennai Qiu, Yufan Dang, Zhuoyun Du, Weize Chen, Cheng Yang, and Chen Qian. Autonomous agents for collaborative task under information asymmetry. In *Advances in Neural Information Processing Systems 37*, pp. 2734–2765, 2024b.

Yuhan Liu, Yuyang Huang, Jiayi Yao, Shaoting Feng, Zhuohan Gu, Kuntai Du, Hanchen Li, Yihua Cheng, Junchen Jiang, Shan Lu, Madan Musuvathi, and Esha Choukse. DroidSpeak: KV cache sharing across fine-tuned model variants. In *23rd USENIX Symposium on Networked Systems Design and Implementation (NSDI 26)*, pp. 319–338. USENIX Association, 2026.

Zechun Liu, Changsheng Zhao, Forrest Iandola, Chen Lai, Yuandong Tian, Igor Fedorov, Yunyang Xiong, Ernie Chang, Yangyang Shi, Raghuraman Krishnamoorthi, Liangzhen Lai, and Vikas Chandra. MobileLLM: Optimizing sub-billion parameter language models for on-device use cases. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 32431–32454. PMLR, 2024c.

Quinn McNemar. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153–157, 1947. doi: 10.1007/BF02295996.

Jesse Mu, Xiang Lisa Li, and Noah Goodman. Learning to compress prompts with gist tokens. In *Advances in Neural Information Processing Systems 36*, pp. 19327–19352, 2023. doi: 10.52202/075280-0848.

Neousys Technology. Flyc-300: Nvidia jetson orin nx-based mission computer for drones and unmanned systems. https://www.neousys-tech.com/en/product/product-lines/edge-ai-gpu-computing/nvidia-jetson/flyc-300, 2025. Accessed 2026-09-24.

NVIDIA. Jetson ai lab: Benchmarks (llm text generation rate, mlc, int4). https://www.jetson-ai-lab.com/archive/benchmarks.html, 2024. Accessed 2026-09-24. Per-model rates are produced by the jetson-containers MLC benchmark, https://github.com/dusty-nv/jetson-containers/tree/master/packages/llm/mlc.

NVIDIA. Jetson agx orin for next-gen robotics. https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/jetson-orin/, 2025. Accessed 2026-09-24.

Hamed Omidvar and Vahideh Akhlaghi. A communication-theoretic framework for llm agents: Cost-aware adaptive reliability, 2026. URL https://arxiv.org/abs/2605.09121.

Joon Sung Park, Joseph C. O'Brien, Carrie J. Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. Generative agents: Interactive simulacra of human behavior, 2023. URL https://arxiv.org/abs/2304.03442.

Chau Pham, Boyi Liu, Yingxiang Yang, Zhengyu Chen, Tianyi Liu, Jianbo Yuan, Bryan A. Plummer, Zhaoran Wang, and Hongxia Yang. Let models speak ciphers: Multiagent debate through embeddings. In *International Conference on Learning Representations*, pp. 51899–51928, 2024.

Reiner Pope, Sholto Douglas, Aakanksha Chowdhery, Jacob Devlin, James Bradbury, Anselm Levskaya, Jonathan Heek, Kefan Xiao, Shivani Agrawal, and Jeff Dean. Efficiently scaling transformer inference, 2022. URL https://arxiv.org/abs/2211.05102.

Claude E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27 (3):379–423, 1948. doi: 10.1002/j.1538-7305.1948.tb01338.x.

Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed H. Chi, Nathanael Schärli, and Denny Zhou. Large language models can be easily distracted by irrelevant context. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 31210–31227. PMLR, 2023.

David Slepian and Jack Wolf. Noiseless coding of correlated information sources. *IEEE Transactions on Information Theory*, 19(4):471–480, 1973. doi: 10.1109/TIT.1973.1055037.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling LLM test-time compute optimally can be more effective than scaling parameters for reasoning. In *International Conference on Learning Representations*, pp. 10131–10165, 2025.

Sebastian Trimpe and Raffaello D'Andrea. Event-based state estimation with variance-based triggering. *IEEE Transactions on Automatic Control*, 59(12):3266–3281, December 2014. doi: 10.1109/tac.2014.2351951.

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/.

Chandra Shekhara Kaushik Valmeekam, Krishna Narayanan, Dileep Kalathil, Jean-Francois Chamberland, and Srinivas Shakkottai. Llmzip: Lossless text compression using large language models, 2023. URL https://arxiv.org/abs/2306.04050.

Edwin B. Wilson. Probable inference, the law of succession, and statistical inference. *Journal of the American Statistical Association*, 22(158):209–212, 1927. doi: 10.1080/01621459.1927.10502953.

Ian H. Witten, Radford M. Neal, and John G. Cleary. Arithmetic coding for data compression. *Communications of the ACM*, 30(6):520–540, 1987. doi: 10.1145/214762.214771.

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.

Aaron Wyner and Jacob Ziv. The rate-distortion function for source coding with side information at the decoder. *IEEE Transactions on Information Theory*, 22(1):1–10, 1976. doi: 10.1109/TIT.1976.1055508.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. In *International Conference on Learning Representations*, pp. 21875–21895, 2024.

Huiqiang Xie, Zhijin Qin, Geoffrey Ye Li, and Biing-Hwang Juang. Deep learning enabled semantic communication systems, 2021. URL https://arxiv.org/abs/2006.10685. IEEE Transactions on Signal Processing.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. 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 (eds.), *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2369–2380, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1259. URL https://aclanthology.org/D18-1259/.

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 36*, pp. 34661–34710, 2023. doi: 10.52202/075280-1506.

A METERING CONSTITUTION: FULL SPECIFICATION

Clock rule and boundary cases. Pre-encounter (free): each agent's own shard, the question, its standing instructions and their prefilled KV cache. Post-encounter (metered): everything else. Three conventions are applied uniformly. (i) Recomputation over an agent's own content for the purpose of communicating is charged; the HUSH encoder's teacher-forced pass, the KV encode of the latent baseline and Handshake's local answer are all billed even though they read only the sender's own data. (ii) Reorganization of content an agent already holds is free; assembling the canonical union after an exchange reuses each side's own half without charge. (iii) Injected state, that is, a spliced peer KV cache, counts toward memory and never toward latency.

Wire pricing. Text: 8 bits per UTF-8 byte, or the exact zlib(9) bitstream. Coded streams: the exact arithmetic-coder bitstream plus declared headers (48 bits in sequential mode; 8 plus 40 per chunk in chunked mode). Learned codes: $K \times (d \cdot b + 16)$ for b-bit quantization with fp16 scales. KV state: the exact quantized payload plus scales and a 64-bit header. Every message in both directions counts.

Latency. Sum convention: $\mathrm{dec}/r_d + \mathrm{pre}/r_p + \mathrm{bits}/\mathrm{link} + \mathrm{rounds} \cdot \mathrm{RTT}$. Critical-path convention, used throughout the paper: costs are bucketed per round and per agent; within a round the wall clock is the slower agent; rounds are serialized; wire time and round-trip time are added once. Batched decode of C independent streams: $\lceil n/C \rceil \, (1 + 0.05 \, (C-1))/r_d$ with $C \leq 8$, which is conservative against published batched-decode scaling on Jetson-class devices. Energy: $(\mathrm{dec} + \mathrm{dec}_{\mathrm{batched}})\, e_d + \mathrm{pre}\, e_p + \mathrm{bits}\, e_{\mathrm{bit}}$, never overlapped.

Device profiles. Both profiles are defined in this paper. The Jetson AGX Orin constants are rounded from published measurements of 7B and 8B models on that module (NVIDIA, 2024; Arya & Simmhan, 2025); the Jetson Orin NX constants are not measured but scaled down from them by about the ratio of the two modules' published compute, 275 against 100 TOPS (NVIDIA, 2025). Jetson AGX Orin profile (running an 8B model): $r_d = 44$ and $r_p = 1000$ tokens per second, 100 Mbps link, 5 ms round trip, 30 W. Jetson Orin NX profile: $r_d = 18$ and $r_p = 400$ tokens per second, 50 Mbps link, 5 ms round trip, 15 W. The published MLC benchmark for a 7B model on the AGX Orin gives 46.9 tokens per second decode and 633 prefill; the sensitivity of the latency threshold to the prefill rate is given in Section 6.2. KV memory: 0.14 GB per 1K tokens (Qwen3-8B in fp16: 36 layers, 8 KV heads, head dimension 128).

Record schema. One JSONL record per episode and device profile: identifiers, topology, question type, gold answer, both agents' answers, per-agent exact match, agreement, abstention flag, bits per direction, decoded, prefilled and injected tokens per agent, batched-decode tuples, rounds, KV growth in GB, latency under both conventions, energy, and the full transcript with route labels, probe contents, verdicts and wire bits per message. Bundled episodes also carry the per-question answers and correctness.

B BENCHMARK CONSTRUCTION AND DATASHEET

Source and deal. HotpotQA dev-distractor items (ten titled paragraphs, two supporting, bridge and comparison labels). $\texttt{deal}(\mathrm{item}, \mathrm{topology})$ assigns the supporting paragraphs per topology (solo: both to A; complementary: one to each; redundant: both to both; null: neither, with the gold answer set to *unknown* and the original answer kept in the metadata) and alternates the distractors deterministically by item hash.

Strict filters and yields. From one pool of 506 items, re-profiled per deal under the abstention-capable prompt: solo $(0, 1, 0, 1)$: 333 of 506; complementary $(0, 0, 0, 1)$: 400 of 506; redundant

$(0, 1, 1, 1)$: 295 of 506; null (all abstain): 316 of 506. Natural memorization ($p_\varnothing = 1$) excludes 16.4% of dev items, and the null filter excludes a further 24.1% of null deals on which the model answers correctly with the evidence absent. The blind mixture samples 100 episodes per topology with a fixed seed; it is bridge-heavy because the strict filter removes comparison items preferentially. The abstention instruction costs 18 points of full-context ceiling on answerable items.

Scale sets. The same 200 strict complementary episodes (set v3) at every size; padding paragraphs come from a pool of 66K dev paragraphs with the supporting titles excluded, are dealt evenly, and are shuffled per shard so that the supporting paragraphs sit at random depth. Realized mean joint sizes: 1,403, 1,436, 2,123, 3,121, 4,137, 5,169, 7,191, 11,255, 15,326, 23,455 and 31,584 tokens, all below the model's 32,768-token window. The 1,436 cell duplicates the natural cell, because natural episodes already exceed its budget, and is omitted from the tables. Sizes below the natural one keep the two supporting paragraphs and $d \in \{0, 2, 4, 6, 8\}$ of the item's own distractors; Table 2 gives the results. Ceilings are order-sensitive (canonical order 1.00, original document order 0.84, shuffled 0.90 at natural size), which is why the shuffled scale axis starts at 0.81 rather than 1.00.

d	realized S	Handover	Select	Escalate
0	212	.865	.865	.865
2	508	.865	.725	.865
4	808	.860	.705	.860
6	1,100	.835	.720	.790
8	1,403	.800	.660	.780

Table 2: **Below the natural size** (n=200, same items, shuffled order). The plateau extends down to the bare pair of supporting paragraphs; handover and Escalate coincide below 1K tokens because Escalate ships the whole shard in one or two rounds; Select loses accuracy as soon as there is anything to select.

Evidence-density sets. For $k \in \{1, 2, 4, 8\}$, k strict complementary items are dealt into one pair of shards (each item's two supporting paragraphs split, two hard negatives per item kept) and padded to the target size (7K or 32K); 200 bundles per cell. The blind set draws k per bundle without telling the agents (57, 45, 57 and 41 bundles for $k = 1, 2, 4, 8$). Each agent answers every question with one decode per question over the same notes; a bundle's accuracy is the mean per-question consensus. An interference control answers each question over a context holding only its own item's four paragraphs, with the removed paragraphs replaced by random paragraphs so that the context keeps the bundle's size: 0.590 (800 questions) at k=4 against 0.60 for handover on the same bundles, and 0.589 (1,600) at k=8 against 0.57. Sharing a context between questions therefore costs no accuracy by itself.

MuSiQue realization. Answerable MuSiQue validation items with 2, 3 or 4 supporting paragraphs; supporting paragraphs alternate between the agents, four of the item's own distractors are kept, and HotpotQA dev paragraphs pad the pair to 7.2K tokens; 200 items per hop count. No model-based strict filter was applied, so full-context handover is the ceiling and the absolute level is the model's MuSiQue accuracy at 7K. Results are in Table 9.

C ADDITIONAL PROTOCOLS

The two receiver-driven protocols summarized in Section 5 are specified here, together with the two baselines that appear only in the appendix tables. Their full results on the scale axis, with wire cost, are in Table 7.

Request. Each round, each agent states its current answer, a request for what it believes it is missing, and one paragraph it discloses; the peer's model must name at least one paragraph in reply. Two weaker versions are kept for comparison: a pure request without disclosure (v1), and a request whose reply is optional (v2); the version with a mandatory reply is v3, the one reported as Request.

Handshake. Each agent answers from its own shard, exchanges a one-line status, and probes the peer with two paragraphs and a short verdict; an answer is committed only if both agents reproduce

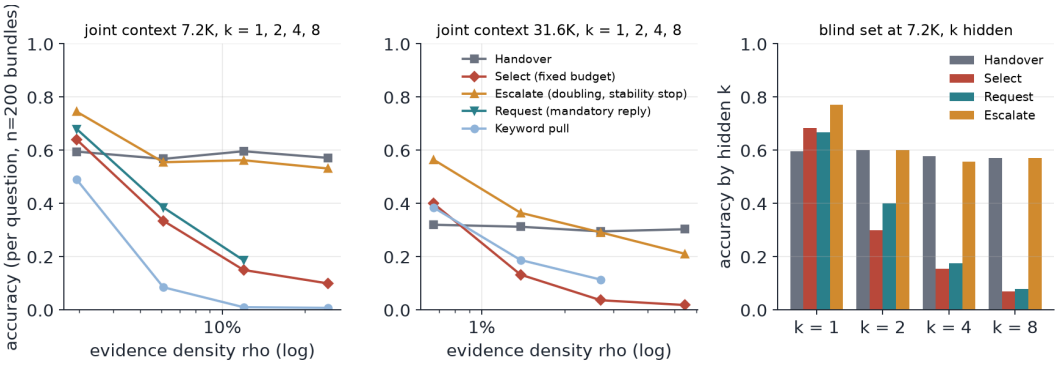


Figure 6: **The evidence-density axis** at 7.2K and 31.6K tokens (n=200 bundles per cell; accuracy is the mean per-question consensus). Right: the blind set at 7.2K, on which the agents are not told k.

it from their own shard plus the peer's probe, otherwise the protocol escalates to Escalate, and if both remain silent it certifies *unknown*. Handshake avoids moving evidence where nothing needs to move.

Keyword pull (B3). The peer is treated as a passive store: an agent decodes a short query, and the peer's device answers it by string matching over its shard without running its model, for up to four rounds. It is the plainest form of pull-based communication and fails on bridge questions, where the asker cannot name the entity it lacks.

Latent KV transfer (B4). The sender ships its KV cache instead of text; the variants and their pre-registered measurement are described under Appendix F.

D Robustness to evidence density

The scale axis has the bias noted in Section 4: every episode needs two paragraphs, so a protocol that sends about two per side is matched to the data. Figure 6 varies the evidence density at a fixed size by bundling k questions on one pair of shards (Table 5), and three results follow one another. First, handover is invariant to the density (0.57 to 0.60 at every k at 7K), so the collapse on the scale axis follows the amount of context and not the proportion that is needed. Second, every protocol with a fixed volume collapses as the need grows, because its cap does not move and a receiver cannot request what it does not know exists: Select falls from 0.64 to 0.10, and every request-based design falls with it, including Request with a mandatory reply (Tables 5 and 8). Third, Escalate adapts without being told k: on the blind set it matches or exceeds handover at every hidden density (0.77, 0.60, 0.56 and 0.57 against 0.60, 0.60, 0.58 and 0.57) while its latency grows by about one round per doubling of the need, and at 32K and k=1 it is 0.56 against handover's 0.32 at a quarter of the latency. An agent can always report that more evidence has stopped changing its answer; it cannot request what it does not know exists.

Since ρ is not observable, it enters the rule as a condition: with m needed paragraphs per side Escalate costs about $\log_2 m + 1$ rounds whereas handover costs $0.5\,S/r_p$ regardless of m, and the two meet between k=4 and k=8 at 7K. The complete rule is therefore: hand over when the context is small or most of it is needed; escalate when the context is large and the need is sparse; fall back to Select only when the deadline is under about three seconds.

E The blind topology mixture

Every fixed recipe fails on some topology, and Handshake helps only where nothing needs to move. The two axes above are complementary topologies throughout. On the blind mixture, which hides solo, redundant and null episodes among them, every fixed recipe fails somewhere (Table 3): conversation spends the most where there is nothing to learn, Select is not enough where cooperation is needed, and Escalate is accurate everywhere but pays its ranking step and first rounds on

redundant and null episodes too. Handshake removes that cost, committing or certifying *unknown* after its probes in about half of Escalate's time on those topologies and matching Escalate on complementary ones, for the same mixture accuracy of 0.96 at 1.37 against 1.55 seconds; it is a router in front of Escalate for mixed topologies. Identical models also make identical mistakes (the same wrong answer on 7.1% of redundant deals), so agreement between two copies of one model is not verification.

	per-topology accuracy (latency)				mixture mean	
Protocol	solo	complementary	redundant	null	acc	lat
Handover	1.00 (0.9 s)	1.00 (0.9 s)	1.00 (1.1 s)	1.00 (0.7 s)	1.00	0.9 s
Conversation	.82 (5.0 s)	.40 (5.3 s)	.84 (2.8 s)	.89 (9.1 s)	.74	5.6 s
Select	.72 (1.2 s)	.73 (1.5 s)	.88 (1.0 s)	1.00 (0.9 s)	.83	1.2 s
Handover over HUSH (chunked)	1.00 (3.8 s)	1.00 (3.7 s)	1.00 (4.7 s)	1.00 (2.9 s)	1.00	3.8 s
Escalate	.96 (1.7 s)	.90 (1.5 s)	.98 (1.4 s)	1.00 (1.6 s)	.96	1.55 s
Handshake	.97 (2.2 s)	.86 (2.0 s)	1.00 (**0.70 s**)	1.00 (**0.54 s**)	.96	**1.37 s**

Table 3: **The blind topology mixture** at natural size (n=100 per topology; handover over HUSH n=25; Jetson AGX Orin profile). Wire bits per episode: handover 22 kbit, Select 12, Escalate 24, Handshake 21, conversation 8, handover over HUSH 6. Handshake commits at the first round on 46 of 48 redundant episodes, certifies *unknown* on 45 of 47 null episodes, and escalates on 47 of 50 complementary episodes.

F FULL RESULT TABLES

S (tokens)	Handover	Select	Conversation	Escalate	Δ(handover − Escalate), paired
212	**.87** / 0.3 s	**.87** / 0.7 s	.58 / 3.2 s	**.87** / 0.6 s	0 (identical answers)
508	**.87** / 0.4 s	.73 / 1.0 s	.50 / 3.9 s	**.87** / 1.0 s	0 (identical answers)
808	**.86** / 0.6 s	.71 / 1.2 s	.47 / 4.7 s	**.86** / 1.2 s	0 (identical answers)
1,100	**.84** / 0.8 s	.72 / 1.4 s	.42 / 5.1 s	.79 / 1.5 s	+0.05 [+0.01, +0.08]
1,403	**.81** / 0.9 s	.67 / 1.4 s	.41 / 5.1 s	.77 / 1.6 s	+0.04 [+0.00, +0.08]
2,123	**.81** / 1.2 s	.65 / 1.5 s	.38 / 5.2 s	.79 / 2.0 s	+0.02 [−0.02, +0.06]
3,121	.65 / 1.7 s	.55 / 1.4 s	.38 / 5.5 s	**.68** / 2.3 s	−0.03 [−0.08, +0.03]
4,137	.59 / 2.3 s	.58 / 1.6 s	.40 / 6.4 s	**.66** / 2.5 s	−0.07 [−0.14, −0.00]
5,169	.58 / 2.8 s	.65 / 1.7 s	.31 / 5.7 s	**.64** / 2.8 s	−0.05 [−0.11, +0.01]
7,191	.52 / 3.8 s	.59 / 1.8 s	.32 / 4.5 s	**.62** / 3.0 s	−0.10 [−0.17, −0.03]
11,255	.47 / 5.8 s	.42 / 1.6 s	.30 / 4.4 s	**.58** / 3.5 s	−0.11 [−0.19, −0.03]
15,326	.35 / 7.8 s	.37 / 1.3 s	.27 / 3.8 s	**.59** / 3.8 s	−0.24 [−0.32, −0.16]
23,455	.28 / 11.9 s	.28 / 1.4 s	.23 / 2.0 s	**.47** / 4.1 s	−0.18 [−0.26, −0.11]
31,584	.20 / 16.0 s	.27 / 1.4 s	.23 / 1.1 s	**.40** / 4.6 s	−0.20 [−0.27, −0.12]

Table 4: **Accuracy and critical-path latency on the scale axis** (200 strict complementary episodes, identical at every size; Jetson AGX Orin profile); the last column is the paired difference between handover and Escalate with its 95% interval. Sizes above 1,403 add padding paragraphs to the natural episodes and sizes below it remove distractors from them (Table 2); at the three smallest, Escalate ships each shard whole and both agents produce exactly handover's answers.

	S=7.2K, accuracy / latency					S=31.6K, accuracy			
Protocol	k=1 (ρ 2.9%)	k=2 (6.1%)	k=4 (12%)	k=8 (24%)	blind	k=1	k=2	k=4	k=8
Handover	.59 / 3.8 s	.57 / 3.9 s	**.60** / 4.1 s	**.57** / 4.5 s	.59 / 4.0 s	.32	.31	**.29**	**.30**
Select	.64 / 1.6 s	.34 / 2.0 s	.15 / 2.3 s	.10 / 2.5 s	.32 / 2.0 s	.40	.13	.04	.02
Request	.68 / 1.6 s	.39	.19		.35	.37			
Escalate	**.74** / 2.4 s	**.56** / 4.7 s	.56 / 6.6 s	.53 / 9.0 s	**.63** / 5.6 s	**.56**	**.36**	.29	.21†

Table 5: **Accuracy against evidence density at a fixed context size** (Jetson AGX Orin profile; $^{\dagger}n$=169, wall-time limit). Blind set by hidden k = 1, 2, 4, 8: handover .60 / .60 / .58 / .57; Select .68 / .30 / .15 / .07; Escalate .77 / .60 / .56 / .57.

S	Handover read	Select read	Select written	Select rounds	Escalate read	Escalate written	Escalate rounds
1,403	702	340	32	2.6	655	30	2.7
2,123	1,062	342	37	2.7	877	33	3.0
3,121	1,560	325	36	2.5	1,155	34	3.2
4,137	2,068	312	45	2.7	1,303	37	3.2
5,169	2,584	327	48	2.8	1,530	38	3.4
7,191	3,596	351	49	2.8	1,706	36	3.3
11,255	5,627	353	40	2.6	2,177	37	3.5
15,326	7,663	304	33	2.5	2,478	37	3.6
23,455	11,727	320	35	2.4	2,652	39	3.7
31,584	15,792	337	38	2.5	2,971	43	3.8

Table 6: **Tokens read and written per agent on the full scale axis** . Read counts include about 16 tokens of instruction continuation per answering step. A padded paragraph is about 135 tokens; Select receives 2.4 to 2.5 paragraphs per agent at every size, Escalate 4.3, 11.5 and 17.3 at 1.4K, 7.2K and 31.6K.

S	Handover	Select	Escalate	Handshake	Guide
1,403	.81 / 0.9 s / 22	.67 / 1.4 s / 14	.77 / 1.6 s / 24	.71 / 2.0 s / 28	.81 / 0.9 s / 22
2,123	.81 / 1.2 s / 33	.65 / 1.5 s / 13	.79 / 2.0 s / 32	.70 / 2.3 s / 35	.82 / 1.3 s / 34
3,121	.65 / 1.7 s / 49	.55 / 1.4 s / 12	.68 / 2.3 s / 43	.64 / 2.6 s / 46	.67 / 2.3 s / 43
4,137	.59 / 2.3 s / 65	.58 / 1.6 s / 12	.66 / 2.5 s / 49	.61 / 2.8 s / 50	.67 / 2.6 s / 49
5,169	.58 / 2.8 s / 81	.65 / 1.7 s / 13	.64 / 2.8 s / 57	.59 / 3.0 s / 57	.64 / 2.8 s / 57
7,191	.52 / 3.8 s / 112	.59 / 1.8 s / 14	.62 / 3.0 s / 64	.58 / 3.2 s / 64	.62 / 3.0 s / 64
11,255	.47 / 5.8 s / 176	.42 / 1.6 s / 14	.58 / 3.5 s / 82	.55 / 3.6 s / 76	.58 / 3.5 s / 81
15,326	.35 / 7.8 s / 239	.37 / 1.3 s / 11	.59 / 3.8 s / 93	.54 / 4.0 s / 89	.59 / 3.9 s / 93
23,455	.28 / 11.9 s / 366	.28 / 1.4 s / 12	.47 / 4.1 s / 99	.45 / 4.2 s / 93	.47 / 4.2 s / 100
31,584	.20 / 16.0 s / 493	.27 / 1.4 s / 13	.40 / 4.6 s / 111	.36 / 4.3 s / 95	.40 / 4.6 s / 112

Table 7: **The full scale axis with wire cost** (accuracy / Jetson AGX Orin latency / kbit, n=200). Handshake escalates almost every complementary episode and therefore adds its probe overhead to Escalate at every size.

Codec on the selective protocols. Attaching HUSH to the wire leaves accuracy unchanged and changes bits and latency as follows (Jetson AGX Orin profile, n=200): Select 13.6 to 5.8 kbit and 1.4 to 3.0 s at 1.4K; 14.1 to 6.1 kbit and 1.8 to 3.5 s at 7.2K; 12.6 to 5.1 kbit and 1.4 to 3.0 s at 31.6K. Escalate 23.6 to 9.1 kbit and 1.6 to 4.5 s at 1.4K; 63.6 to 25.7 kbit and 3.0 to 10.6 s at 7.2K; 110.5 to 44.6 kbit and 4.6 to 17.3 s at 31.6K.

Drone profile. Accuracies are device independent, so the paired differences and S^* are identical on the Jetson Orin NX profile; latencies scale by about 2.4 and the latency crossovers follow Equation 1 (2.7K against Select). Evaluating the decision chart across device speeds and deadlines gives the same three regions: at every deadline the device can meet, the first grid size at which handover is beaten is 3.1K whatever the device speed, because that boundary is set by accuracy; under tighter deadlines handover stops meeting the deadline before any protocol has to beat it, at 2.1K on the Jetson AGX Orin profile with a one-second deadline and at about 1K on a device four times slower.

Latent KV transfer (B4, pre-registered). Exchanging KV caches (Liu et al., 2026) or embedding vectors (Pham et al., 2024) assumes the receiver can consume the sender's representation directly, which requires identical architectures; on a wireless link a cache is orders of magnitude larger than the information it carries, which is why serving systems compress or evict it (Kwon et al., 2023; Zhang et al., 2023). Before measuring it we pre-registered where the meter's arithmetic places a protocol that ships the sender's KV cache, and the measurements landed within 5 to 20% of the prediction. A naive splice of the peer's cache reaches 0.25 to 0.27 on strict complementary episodes and 0.27 with conditioning and canonical ordering, indistinguishable from a null control; a bilateral re-encode in which each side prepends the peer's cache and recomputes its own shard reaches 0.60 at natural size and 0.48, 0.40 and 0.12 at 1.4K, 7.2K and 31.6K, and the same at int4 as at fp16. The

Protocol	S=7.2K k=1	k=2	k=4	k=8	blind	S=31.6K k=1	k=2	k=4	k=8
Keyword pull (B3)	.49 / 1.7 s	.09 / 1.0 s	.01 / 0.6 s	.01 / 0.8 s	.20 / 1.0 s	.39			
Request v1 (no disclosure)	.07 / 1.2 s	.09 / 2.1 s	.03 / 2.1 s	.02	.07 / 1.9 s	.05			
Request v2 (optional reply)	.50 / 1.4 s	.29	.14	.07	.27				
Request v3 (mandatory reply)	.68 / 1.6 s	.39	.19		.35	.37			

Table 8: **Keyword pull and the three Request versions on the evidence-density axis** (Jetson AGX Orin profile). Under v2, 92% of requests were answered with nothing; v3 forces a reply. Blind set by hidden k: keyword pull .60 / .09 / .01 / .02; Request v3 .67 / .18 / . . .

hops (ρ)	Handover	Select	Escalate	Request
2 (3.4%)	.23 / 3.8 s	**.28** / 1.5 s	.25 / 3.1 s	.24 / 1.9 s
3 (7.0%)	.17 / 3.7 s	.18 / 1.8 s	**.24** / 4.1 s	.21 / 2.5 s
4 (7.4%)	.12 / 3.7 s	.10 / 1.9 s	.11 / 4.2 s	**.13** / 2.1 s

Table 9: **MuSiQue realization of the evidence-density axis** at $S \approx 7.2$K (n=200 per hop count). The absolute level is set by the model (abstention rises from 0.2 to 0.5 with hops). Every communication protocol matches or exceeds handover at a fraction of its latency; Select loses its lead as the needed paragraphs increase. The four-hop cell has no higher ρ than the three-hop cell because MuSiQue's supporting paragraphs are short.

cache is 3×10^4 times the bits of the coded text for the same content, so this is the only protocol whose latency is dominated by airtime (break-even at 1.2 Gbps raw or 18 Mbps compressed); cross-attention over foreign cache entries, not bandwidth, is its accuracy bottleneck. A cache carries computation, not information.

Learned gist (negative result). A projection, in the spirit of gist tokens (Mu et al., 2023), trained from frozen shard features to K soft tokens injected into the peer's context appeared to help: B answering with its true peer gist reached 0.470, but with the wrong episode's gist 0.480 and with matched-norm random noise 0.050 (n=100, strict complementary), and accuracy was flat in capacity ($K = 1, 4, 8, 16$: 0.632, 0.660, 0.667, 0.670). The apparent gain carried zero bits. Training used 21,339 episodes under a lighter two-condition filter with role-swap augmentation. Every latent channel should be audited with shuffled and random controls before it is believed.

G WHY HANDOVER FAILS AND WHY LANGUAGE PROTOCOLS FAIL

The decline of Figure 1 is the dilution and distraction reported in the literature (Liu et al., 2024a; Shi et al., 2023), and it follows volume rather than the useful fraction (Appendix D); the same model scores above 90 on RULER (Hsieh et al., 2024) at 32K in non-thinking mode (Yang et al., 2025), but RULER retrieves and traces items inserted into unrelated text, whereas a two-hop question over two hundred distractors requires finding and combining two paragraphs among topical competitors, the regime that Levy et al. (2024) show degrades far inside the window. The bottleneck of the language protocols is consensus rather than knowledge: naive summary exchange reaches 46% either-agent accuracy and 6% consensus, whereas protocols that reconstruct identical contexts on both devices agree by construction. Identical models also make identical mistakes (the same wrong answer on 7.1% of redundant deals), so agreement between two copies of one model is not verification. The ranking ablation (Table 1) locates the remaining accuracy: a receiver that reads a well-ordered subset answers at 32K tokens as it does at the natural size, so the next gains lie in the ranking step, and letting the model choose each round in view of what it has received already recovers about half of the gap to the oracle.

H AUDIT OF THE HANDOVER COLLAPSE

Because the paper rests on the curve in Figure 1, eight explanations other than the model were tested and excluded before building on it (Table 11).

S	Protocol	n	acc	bits	KV (GB)	lat, AGX Orin (s)	lat, Orin NX (s)
1.4K	Handover	50	0.90	21.8k	0.11	0.91	2.26
1.4K	Handover over HUSH (chunked)	25	0.96	5.9k	0.19	3.68	9.03
1.4K	Handover over HUSH (sequential)	25	0.96	4.2k	0.19	18.1	44.2
7.2K	Handover	50	0.56	110k	0.51	3.77	9.40
7.2K	Handover over HUSH (chunked)	25	0.64	24.9k	1.01	17.7	43.5
31.6K	Handover	50	0.30	446k	2.22	16.0	39.9
31.6K	Handover over HUSH (chunked)	12	0.17	100k	4.42	76.6	188
31.6K	Handover over HUSH (sequential)	12	0.17	95.8k	4.42	375	919

Table 10: **The codec on whole shards** (the original three-point scale set, 50 complementary episodes, both device profiles). The coded modes reproduce handover's answers byte for byte on shared episodes (62 of 62), so their accuracy equals the bound's; the small-n cells are consistency checks.

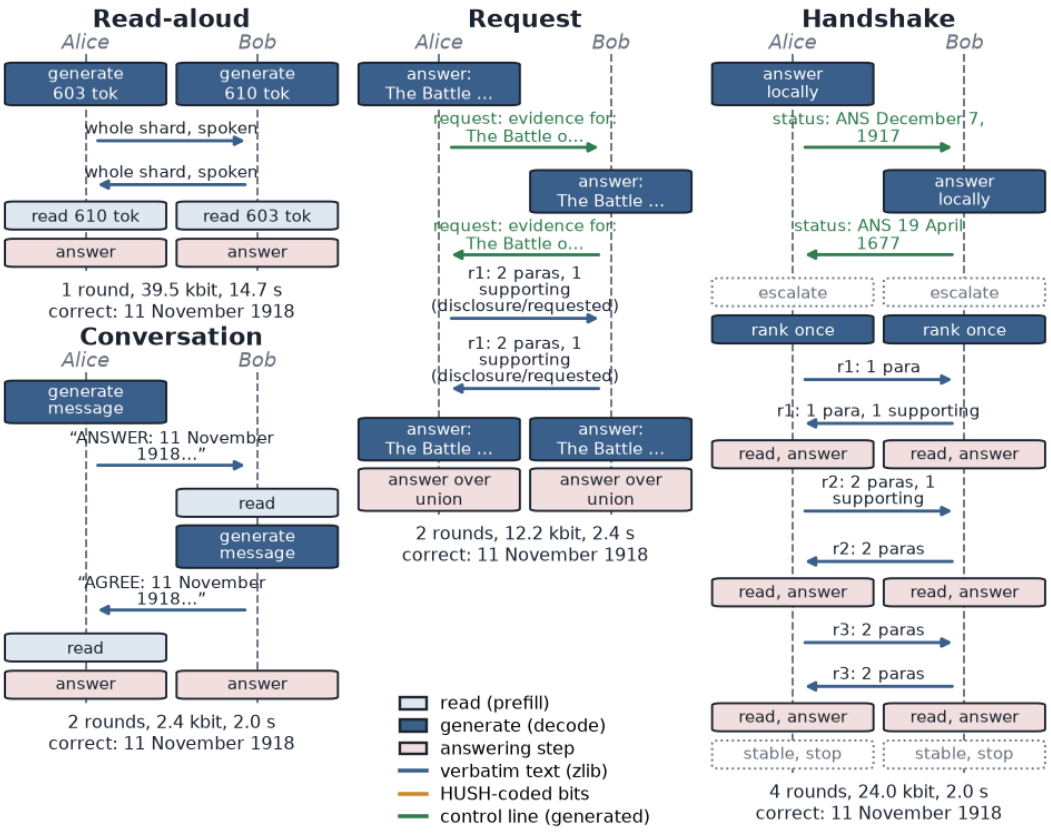


Figure 7: **The episode of Figure 2 under the four remaining protocols**, drawn from the run records in the same notation. All nine protocols answer this episode correctly; they differ in what they make each agent read and write and in how many rounds they take.

I Reproducibility Notes

Names in the paper and in the code. Handover is `b0_swap_gzip`; read-aloud is `b1_readaloud`; conversation is `b2_conv_simplex`; keyword pull is `b3_request`; latent transfer is `b4_kv`; Select is `hush_select`; Escalate is `hush_double`; Request v1, v2 and v3 are `hush_pull`, `hush_pullv2` and `hush_pullv3`; Handshake is `hush_adaptive_v2`; HUSH in sequential and chunked mode is `hush_lossless` and `hush_parallel`, and the coded wire of Select and Escalate is the option `wire: coded`; the humanoid and Jetson Orin NX profiles

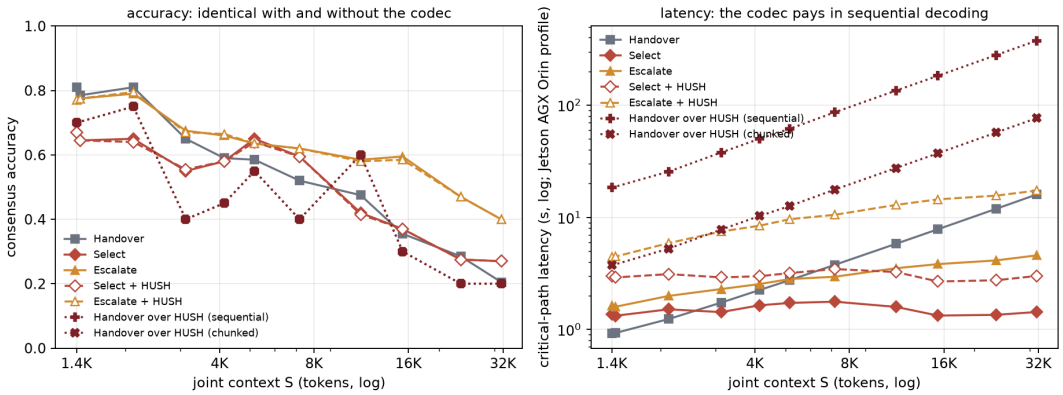


Figure 8: **Protocols with and without the HUSH codec on the wire.** Left: consensus accuracy; right: critical-path latency on the Jetson AGX Orin profile. Select+HUSH and Escalate+HUSH send exactly the paragraphs that Select and Escalate send, so their accuracy is identical; their latency is higher because the receiver decodes the paragraphs token by token. Select+HUSH and Escalate+HUSH are measured on the same 200 episodes at every size. Handover over HUSH, whose sequential decoding is slow, is measured on subsets of those episodes: 20 at each size up to 7.2K, 10 at 11K and 15K, and 5 at 23K and 31.6K, which is why its accuracy curve is noisy; the sequential and chunked modes agree on every episode.

#	Suspect	Test and verdict
1	Figure values not traceable	Values matched against the per-episode records and the merged scale table.
2	Context silently truncated	Joint size 31,184 to 31,708 tokens plus a 768-token margin, below the 32,768 window; `max_position_embeddings` is 40,960.
3	Degenerate output (overflow, repetition)	All 35 wrong answers at 32K read: coherent wrong entities, 4 abstentions, 5 sentence copies.
4	Prefill code path (single-shot below 4K, chunked above)	Both paths forced on every episode: 50 of 50 identical at 7K; 50 of 50 identical at 32K at three depths, 48 of 50 at the fourth (two fp16 tie flips). The dense grid shows its largest drop (0.81 to 0.65) between 2.1K and 3.1K, inside the single-shot path, and no discontinuity at the 4K boundary.
5	fp16 numerics	32K, n=50, H100: fp16 0.28, bf16 0.26, fp32 0.28.
6	Environment or kernel drift	Re-run under a fresh environment on an RTX 6000: 50 of 50 answers byte-identical to the earlier records. H100 against RTX 6000: 49 of 50 identical.
7	Position artefact	Supporting paragraphs moved to the start, middle or end of each shard at 32K: natural 0.28, middle 0.32, start 0.46 to 0.50, end 0.48. Position matters by about a factor of two; the best placement remains far below 0.81.
8	Small sample	n raised from 50 to 200: 32K falls from 0.30 to 0.20 (interval 0.14 to 0.26); 23.5K is 0.28.

Table 11: **Eight suspects for the handover collapse**, each tested and excluded.

are `TIER_H_ORIN` and `TIER_D_ORIN_NX`. The code identifiers predate the names used in the paper and are kept for the sake of the released records.

Hazard 1: silent quadratic attention on deployed-class GPUs. Recent transformers versions pass grouped-query K and V unexpanded to the fused attention kernels; on Turing (sm_75) every fused kernel rejects that shape and attention falls back silently to the unfused path, which is quadratic in memory and runs out of memory from about 7K tokens on a 24 GB card while being invisible at short contexts. The fix is backend-local: restore the K/V expansion so the memory-efficient kernel engages, and use chunked prefill with a manual greedy loop for prompts above 4K. The short path

is byte-identical to all earlier runs, and cross-process logit hashes are identical under the fix at 7.2K, 15.6K and 31.5K, so coder synchrony survives.

Hazard 2: train-mode generation reads an empty cache. With gradient checkpointing enabled in train mode, the library disables the KV cache silently; any generation-based validation then decodes from an empty cache and returns garbage. This zeroed all in-training validation of the gist study (reported 0.03 against a true 0.75). The fix is an eval-mode guard around any validation that generates.

Planning constants. Sequential coding on an RTX 6000 class GPU takes about 18 minutes per episode at 7.2K joint context and 81 to 92 minutes at 31.6K, both directions; Escalate over the coded wire at 31.6K takes about 5.6 minutes per episode on an A100. Long-context KV splicing needs the shipped caches held off the GPU (2.2 GB fp16 per 16K-token shard). Cross-process bit exactness: 566 of 566 round trips; encoder and decoder must share kernels and device class.