WHAT DOES TALKING ADD? THE VALUE OF FEEDBACK IN MULTI-AGENT INFERENCE

Anonymous authors
Paper under double-blind review

ABSTRACT

In multi-agent language model systems, a decision-maker combines reports from agents with different private evidence to answer a task question. Limited report space requires each agent to select which evidence to include. The information needed for this selection may be held by another agent, leaving the reporting agent uncertain about what to preserve. A follow-up protocol addresses this issue by allowing the decision-maker to ask questions after reading the initial reports. We compare the accuracy and cost of this protocol with stronger independent reporting and sequential reporting protocols. Stronger independent reporting improves report encoding and allows local revision, while sequential reporting sends selection information from one agent to another before the receiving agent prepares a report. Each agent's initial evidence access and the total resource limits remain unchanged across protocols. Our analysis separates accuracy loss caused by information missing from reports from failures to use information already received. Before the sender knows which evidence is needed, a report must support multiple possible needs. Earlier access to selection information can reduce what the report must preserve. Controlled routing and document-selection experiments show that stronger reports reduce the estimated accuracy advantage of follow-up. With the direction of information dependence known, sequential reporting achieves high accuracy with fewer generated tokens than follow-up. The experiments also reveal errors in preparing and reading reports, even when the report budget is sufficient to include all required information.

1 INTRODUCTION

Multi-agent language model systems divide work among agents that exchange messages to reach a joint decision. Frameworks such as AutoGen and MetaGPT organize these exchanges through programmable conversations and role-specific workflows (Wu et al., 2024; Hong et al., 2024). When agents hold different evidence, a useful local report may depend on information the sender does not possess. An agent can hold the decisive record while a collaborator holds the clue needed to identify that record. Reporting then requires choosing what to preserve before the current information need is fully visible. Follow-up questions can resolve this uncertainty, but consume resources that could support more informative reports or additional local checking. Their value therefore depends on what stronger reports or a fixed sequence of messages could already achieve.

Consider diagnosing a failed request in a software service. A log agent observes that the request followed path $r7$, while a configuration agent holds component and version records for many paths. Before receiving the path clue, the configuration agent must choose which records to include in a limited report. **Figure 1** compares three ways to obtain the needed evidence. Independent reports can encode records for several possible paths. A fixed exchange can first send the path clue to the configuration agent and the decision-maker, allowing the configuration report to focus on $r7$. Alternatively, the decision-maker can read both initial reports and ask for a missing detail. Both forms of interaction allow evidence selection after a clue arrives, whereas a sufficiently informative initial report may already preserve the required record.

Existing work motivates evaluating communication against improvements to the underlying inference procedure. Debate, independent sampling, local revision, and test-time compute allocation can improve decisions on evaluated tasks (Du et al., 2024; Wang et al., 2023; Li et al., 2024; Madaan

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.

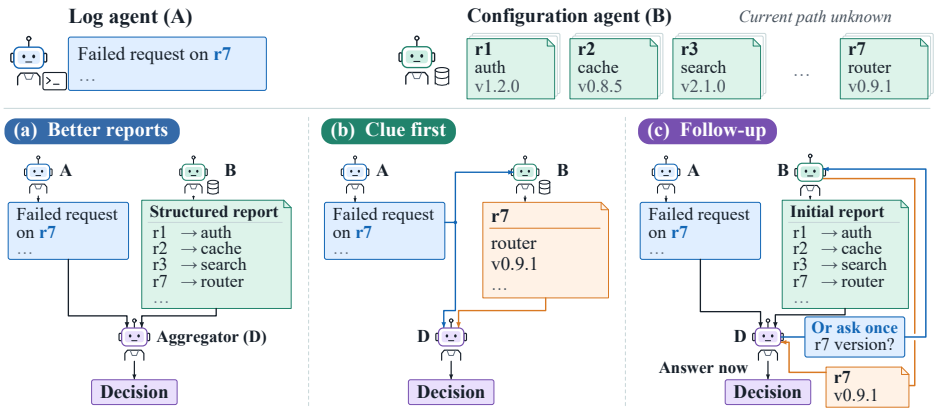


Figure 1. Three ways to use distributed evidence in fault diagnosis.

et al., 2023; Snell et al., 2025). Comparisons that match thinking budgets across methods, together with voting controls and interventions on diversity and confidence, show that interaction gains depend on computation and aggregation as well as information exchange (Choi et al., 2025; Tran & Kiela, 2026; Zhu et al., 2026). Efficiency methods reduce costs through pruning, sparse communication, and lower-cost replacements (Wang et al., 2025; Zeng et al., 2025; Chen et al., 2026). For distributed evidence, failure analyses and HiddenBench identify limitations in coordination and information exploration, with structured communication improving evidence integration (Cemri et al., 2025; Li et al., 2026). Information bottleneck analysis, prompt compression, and knowledge-aware orchestration connect message design to retained information and collaborators' needs (Yu et al., 2026; Pan et al., 2024; Zhang et al., 2025). These studies motivate comparing the evidence supplied by an exchange with what stronger reports could preserve. The remaining question is how much decision value feedback adds after improving reports or providing selection clues through a fixed exchange.

We examine this question through the timing of evidence selection. A report sent before the relevant need is known must serve multiple possible needs, while a later report can focus on the current need. Classical one-way communication reasoning characterizes the distinctions required for joint coverage (Roughgarden, 2015). Reconstruction error then bounds the residual reporting loss. Models may also fail to use available evidence (Liu et al., 2024), motivating a separate term for unused decision potential and a diagnostic based on record decoding. We connect these information conditions to protocol comparisons, clue-timing interventions, and fixed-report evidence-access tests. Record annotations distinguish omitted targets from failures after an entry has been transmitted, and joint reader outcomes identify recoverable answers missed by the model. Experiments use controlled routing and document selection with two language models, retaining a known direction of information dependence. Our contributions are as follows.

1. **Comparisons for the value of feedback.** We compare autonomous follow-up with stronger independent reporting and fixed-order communication under common evidence permissions and resource ceilings. Paired accuracy differences and separate transmission and generation costs characterize the resulting quality–cost trade-offs.
2. **Information conditions and record diagnostics.** We characterize the distinctions required for report coverage and bound reporting loss through reconstruction error. Decoding actual decision records provides a lower bound on unused decision potential, linking the analysis to evidence available at the decision stage.
3. **Evidence on reporting and clue timing.** Stronger reporting substantially reduces the estimated follow-up advantage on permutation routing. Earlier clues allow a fixed exchange to achieve high accuracy at lower generation cost. Fixed-report interventions and common-reader comparisons distinguish the benefit of renewed evidence access from improvements in using received information.

2 MODEL AND PROBLEM FORMULATION

Assessing feedback requires fixing the task and the evidence available to each participant before comparing communication procedures.

2.1 DISTRIBUTED EVIDENCE AND INFORMATION FLOW

The fault-diagnosis example separates the clue identifying a relevant record from the required record content. To preserve this distinction across protocols, we represent the task input, private evidence, and evaluation target jointly.

Definition 1 (Distributed-evidence task)**.** A task is a fixed distribution P over instances $Z = (X, E_A, E_B, Y)$ with a finite answer set $\mathcal{Y}$. Input X is available to A, B, and D. Agents A and B hold private evidence E_A and E_B, respectively, and the evaluation target is $Y \in \mathcal{Y}$. Let I_A, I_B, and I_D denote the initial inputs of agents A, B, and decision-maker D, respectively. The instance distribution and initial access are specified by

$$Z \sim P, \qquad I_A = (X, E_A), \quad I_B = (X, E_B), \quad I_D = X. \tag{1}$$

Starting from these initial inputs, agents exchange messages that form the decision-maker's evidence record. A protocol π comprises message-generation procedures, routing and stopping rules, and a terminal decision rule. For each role $i \in \{A, B, D\}$, let H_i denote the message sequence received by that role. The role generates messages from I_i, H_i, and locally retained state. Let T_D^π denote D's received message sequence and a_π the final answer. Selecting evidence using peer messages constitutes *peer-conditioned disclosure*.

Model parameters and prompts are fixed before evaluation, and internal random seeds are independent of the instance. Access to Y is restricted to the evaluator.

2.2 COMPARISON PROTOCOLS

We compare autonomous querying with two alternatives that obtain decision-relevant evidence by improving independent reports or providing selection clues before reporting. The four protocols retain the same initial access and decision interface while varying local processing and clue timing.

Independent reporting (**Report**) tests what independent preparation can achieve. Agents A, B select, compress, and organize evidence into task-aware reports for D. Reports are mutually unseen before transmission, with content determined from each sender's initial input.

Reporting with local revision (**Revise**) adds private checking and revision (Madaan et al., 2023), using only the sender's initial input and draft. **Revise** tests whether local refinement improves reporting, providing an additional reference for assessing the value of communication.

Sequential reporting (**Sequential**) tests whether a fixed message order supplies useful selection clues. The same message from A reaches both B and D, then B reports conditionally and D answers. The direction and round count are fixed during development, while B's content adapts to A's message. **Sequential** thus provides peer-conditioned disclosure through a predetermined schedule.

Autonomous follow-up (**Follow-up**) tests the flexibility of selecting a query after aggregation. After reading independent reports, D either answers directly or selects one agent and a question. The agent responds using the permitted initial input, local state, and query, then D answers. This single optional query makes the decision to seek further evidence, the recipient, and the content depend on the initial reports.

Appendix A.2 gives the execution sequences and their information-flow diagram in **Figure A1**.

2.3 RESOURCE CONSTRAINTS AND COMPARISON TARGETS

In **Sequential**, one generated message has two recipients. In **Revise**, private drafts consume generation without transmission. We therefore define separate communication and generation budgets.

Definition 2 (Resource-feasible protocol)**.** For communication budget b and generation budget c, let $\mathcal{E}_\pi$ be the multiset of actual message transmissions and $\mathcal{J}_\pi$ the model calls in an execution. For

transmission e, m_e is the complete message and $\ell(m_e)$ is the length under the deployed tokenizer. For call j, n_j counts generated tokens. The costs and feasibility bounds are given by

$$C_\pi = \sum_{e \in \mathcal{E}_\pi} \ell(m_e), \qquad G_\pi = \sum_{j \in \mathcal{J}_\pi} n_j, \qquad C_\pi \le b, \quad G_\pi \le c \quad \text{almost surely.} \tag{2}$$

A protocol is feasible at (b, c) when execution satisfies these bounds and the stated information permissions.

The two sums in **Definition 2** count transmission and generation separately. Communication includes every transmitted copy. Generation includes drafts, checking, compression, final answers, and observable thinking tokens. We report actual usage alongside budgets. Generated-token counts provide a proxy for computational work (**Appendix B**).

Under these budgets, we compare protocol accuracy, defined as $\mathrm{Acc}_\pi = \Pr(a_\pi = Y)$ over task and model randomness. For the reporting comparison, we select a **Report** or **Revise** candidate by a prespecified development rule. The selected protocol, denoted Report_0, is fixed before testing. The selection covers only the declared candidates and development procedure (**Appendix C.1**).

Using Follow and Seq for **Follow-up** and **Sequential**, the paired accuracy differences against the selected reporting protocol and **Sequential** are given by

$$\Delta_{\mathrm{report}} = \mathrm{Acc}_{\mathrm{Follow}} - \mathrm{Acc}_{\mathrm{Report}_0}, \qquad \Delta_{\mathrm{seq}} = \mathrm{Acc}_{\mathrm{Follow}} - \mathrm{Acc}_{\mathrm{Seq}}. \tag{3}$$

Both paired accuracy differences and actual resource use characterize the trade-off between quality and cost, while uncertainty intervals quantify the precision of the comparison (**Appendix C.2**). The acceptable trade-off depends on the application. Explaining either difference requires separating information lost during transmission from failures to use the resulting record.

3 Information boundaries and feedback opportunities

The protocols in **Section 2** can differ in both the evidence transmitted and the accuracy achieved from that evidence. We separate these sources of error before examining how clue timing changes report coverage.

3.1 Reporting loss and decision use

In fault diagnosis, the received record must preserve the relevant configuration state, and the decision-maker must link that state to the request's path. To distinguish information preservation from decision use, we compare the model with an ideal decision rule given the same record.

Let V_π be the complete decision record, including (I_D, T_D^π) and all retained queries or local state used to answer. For any observation U, choosing a most probable label maximizes accuracy. The Bayes reference averages this optimal accuracy over U and is given by

$$\mathcal{B}(U) = \mathbb{E}\left[\max_{y \in \mathcal{Y}} \Pr(Y = y \mid U)\right]. \tag{4}$$

The gap from this reference, $\mathcal{B}(V_\pi) - \mathrm{Acc}_\pi$, measures unused decision potential. The reference allows unrestricted computation, while the model operates under the stated budgets.

To measure the information preserved during transmission, we compare the received record with the team's original evidence $W = (X, E_A, E_B)$, whose Bayes reference is $\mathcal{B}(W)$. We assume $Y \perp V_\pi \mid W$, excluding external answer information, and $Y \perp a_\pi \mid V_\pi$, requiring a complete decision record. The two references then yield the following decomposition.

Proposition 1 (Reporting loss and unused decision potential)**.** *Under these assumptions, the accuracy gap relative to full evidence decomposes into two nonnegative terms given by*

$$\mathcal{B}(W) - \mathrm{Acc}_\pi = \underbrace{\mathcal{B}(W) - \mathcal{B}(V_\pi)}_{L_\pi \text{ (reporting loss)}} + \underbrace{\mathcal{B}(V_\pi) - \mathrm{Acc}_\pi}_{U_\pi \text{ (unused potential)}}. \tag{5}$$

*On the same task distribution and original evidence, the accuracy difference between **Follow-up** and a comparator π is given by*

$$\mathrm{Acc}_{\mathrm{Follow}} - \mathrm{Acc}_\pi = (L_\pi - L_{\mathrm{Follow}}) + (U_\pi - U_{\mathrm{Follow}}). \tag{6}$$

Equation (6) separates changes in available information from changes in decision use. A reduction in reporting loss can be offset by worse use of the received information, so a more informative record need not improve model accuracy. **Appendices D.2 and D.5** give the proof.

Further computation on an unchanged complete record acts on U_π while preserving the record's Bayes reference. **Revise** checks before transmission and can change report content. **Sequential** and **Follow-up** also change evidence selection through peer clues. Even when protocol budgets are equal, message allocations may differ, so one protocol's record need not contain another's (**Appendix D.3**). We therefore examine which required states each report preserves.

3.2 Clue timing and report coverage

Fix the task input X. Let q be the need determined by A's lawful observation and $g(q, E_B)$ the required state in B's evidence. In fault diagnosis, q identifies the request's path and g selects the corresponding configuration state. A deterministic decision rule f maps this state to the target, giving $Y = f(X, q, g(q, E_B))$. The need describes what must be recovered, and is distinct from **Follow-up**'s generated question.

Before receiving q, agent B uses a reporting rule r based only on E_B. The report must support recovery for every need compatible with the task. Write Λ for the declared lawful pairs (q, e) of needs and evidence states, suppressing the fixed X.

Definition 3 (Joint report coverage)**.** A report $r(E_B)$ jointly covers Λ if a fixed decoder can recover the required state for every lawful pair. The coverage condition is expressed as

$$\mathrm{decode}(q, r(e)) = g(q, e) \qquad \text{for all } (q, e) \in \Lambda. \tag{7}$$

If the decision record V_π retains the task input X, need q, and such a report, the target is determined, giving $\mathcal{B}(V_\pi) = \mathcal{B}(W) = 1$ and $L_\pi = 0$. Joint coverage is sufficient for zero reporting loss and stronger than preserving optimal accuracy under a particular distribution. Actual accuracy also depends on the model's use of the decoded state.

To determine what coverage demands of a representation, consider evidence states mapped to the same report. Two evidence states can yield the same report only if they require the same state for every need lawful for both.

Proposition 2 (Distinctions required for coverage)**.** *For finite nonempty need and evidence sets, a deterministic report r covers Λ if and only if, for every $(q, e), (q, e') \in \Lambda$, the equality $r(e) = r(e')$ implies $g(q, e) = g(q, e')$.*

Before learning q, a report must preserve distinctions relevant to any lawful need. After learning q, an encoder $r(e, q)$ needs only the distinctions relevant to that need. This change determines how many different messages an exact report must be able to express. **Appendix E.3** gives the proof.

Consider finite nonempty need and evidence sets $\mathcal{Q}, \mathcal{E}$ with every pair lawful. For each evidence state e, the profile $(g(q, e))_{q \in \mathcal{Q}}$ lists the required states across all needs. Let N count distinct profiles as e varies and N_q count distinct required states at a fixed need q. The decoder knows q in both cases. With unrestricted computation and a codebook fixed beforehand for encoding and decoding, the minimum numbers of report symbols before and after the encoder learns q are given by

$$M_{\text{before}} = N, \qquad M_{\text{after}} = \max_{q \in \mathcal{Q}} N_q. \tag{8}$$

Distinct profiles require distinct messages before the clue, whereas only required states at that need must be distinguished afterward. **Appendix E.4** proves this classical one-way communication count (Roughgarden, 2015).

For K routes and distinct components under a fixed active rule, each assignment specifies a profile of selected components. Cyclic shifts yield K profiles, whereas arbitrary permutations yield $K!$. Once the route is known, either family has K possible selected components. Substituting these counts into **Equation** (8) gives

$$(M_{\text{before}}, M_{\text{after}})_{\text{cyclic}} = (K, K), \qquad (M_{\text{before}}, M_{\text{after}})_{\text{permutation}} = (K!, K). \tag{9}$$

The two families have the same conditional state space but different requirements before the clue arrives. Both **Sequential** and **Follow-up** permit disclosure after the clue. These counts characterize

exact recovery for every lawful need. Average error additionally depends on the task distribution, while token-budget feasibility depends on complete serialized messages (**Appendix E.4**).

Model-generated reports may fail exact coverage. To relate such failures to reporting loss, let R_π be the report actually generated and fix a decoder $d(X, q, R_\pi)$ with a failure value for malformed inputs. Suppose (X, q, R_π) is recoverable from V_π and the deterministic target relation holds almost surely. Define the reconstruction error as $\epsilon_\pi = \Pr\{d(X, q, R_\pi) \neq g(q, E_B)\}$, with g understood conditionally on X.

Proposition 3 (Reconstruction and reporting loss)**.** *Under these conditions and **Proposition 1**'s assumptions for π, bounds on reporting loss and unused decision potential are given by*

$$L_\pi \leq \epsilon_\pi, \qquad U_\pi \geq \max\{0, 1 - \mathrm{Acc}_\pi - \epsilon_\pi\}. \tag{10}$$

Whenever reconstruction succeeds, applying f yields the correct answer. The decision record therefore supports accuracy of at least $1-\epsilon_\pi$. Combining $\mathcal{B}(V_\pi) \geq 1-\epsilon_\pi$ with $\mathcal{B}(W) = 1$ in **Equation** (5) gives both bounds. **Appendix E.5** provides the full proof.

An answer decoder provides a direct diagnostic without estimating the Bayes reference. Let d_{ans} read the complete decision record V_π, with accuracy $\mathrm{Acc}_{d,\pi} = \Pr\{d_{\mathrm{ans}}(V_\pi) = Y\}$. Since $\mathcal{B}(V_\pi) \geq \mathrm{Acc}_{d,\pi}$, a lower bound on unused decision potential is given by

$$U_\pi \geq \max\{0, \mathrm{Acc}_{d,\pi} - \mathrm{Acc}_\pi\}. \tag{11}$$

The decoder must recover the need from material available to D, with ground truth used only for scoring. Paired accuracy differences and intervals then provide estimates for this lower bound on unused decision potential. A decoder failure identifies a limitation of that decoder and leaves open recovery by another rule. Constructed encodings establish representation feasibility, while this diagnostic requires the reports actually transmitted (**Appendix E.1–E.2**).

3.3 Implications for protocol design and evaluation

A compact representation can reduce the value of later disclosure when the model can produce and read reports that cover the required needs. When joint coverage becomes costly, an earlier clue can instead guide the sender to the required state. If A's report identifies that need, **Sequential** supplies the clue through a fixed order, while **Follow-up** additionally chooses a recipient and question after aggregation.

Record decoding and evidence-access interventions examine different parts of this account. **Equation** (11) assesses decision potential left unused in a received record. Holding reports and queries fixed while changing private-evidence access examines the benefit of further disclosure. The following experiments use these diagnostics to explain the accuracy and cost differences between protocols.

4 Experiments

We examine how report quality and clue timing affect inference under limited communication. Protocol comparisons establish the accuracy and cost differences, while record diagnostics and evidence-access interventions examine where these differences arise.

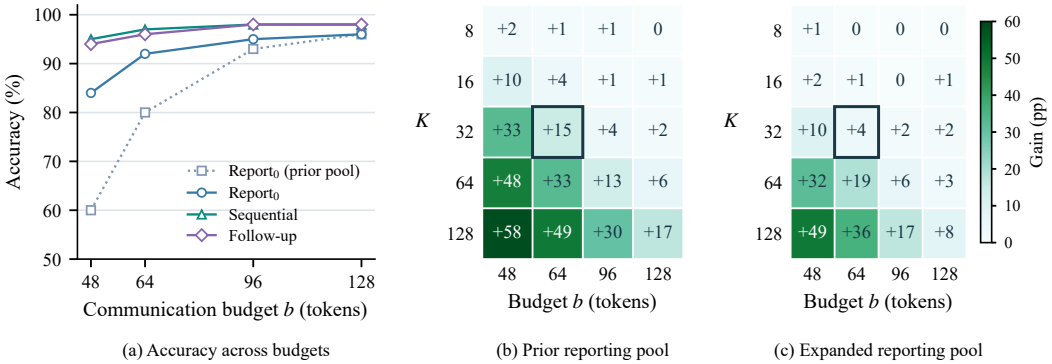
4.1 Experimental setup

Agent A resolves a request to one of K routes. Agent B holds the route-to-component assignment, an active failure rule, and component states. Decision-maker D identifies the selected component under that rule from transmitted messages. All candidate components fail the active check when selected. *Cyclic* assignments shift all route indices equally, while *Permutation* assignments use an arbitrary permutation. Both have uniform route selectors and share base worlds with targets recomputed for each assignment. The two structures instantiate **Equation** (9) under a known $A \to B$ information dependence.

The primary model is Qwen3-14B in thinking mode (Qwen Team, 2025). We compare the protocols in **Section 2** and select Report_0 from the expanded **Report/Revise** pool on 64 independent development worlds. Selection pools accuracy across both structures, with communication and generation

Table 1. Accuracy and mean costs at $K = 32$, given as Cyclic / Permutation. The $b = 64$ rows use 200 worlds and the $b = 128$ row uses 100. Bold and underlined accuracies mark the highest and second-highest values among the three expanded-pool procedures at $b = 64$. Bold costs mark the lowest usage.

Protocol	b	Accuracy (%)	Communication	Generation (k)
Report_0 (prior pool)	64	96 / 80	24 / 38	2.00 / 3.80
Report_0	64	97 / 92	**24** / 37	2.20 / 3.20
Sequential	64	**98** / **97**	38 / **36**	**1.25** / **1.20**
Follow-up	64	97 / 96	25 / 56	2.10 / 5.00
Report_0	128	97 / 96	28 / 94	1.80 / 3.50



(a) Accuracy across budgets (b) Prior reporting pool (c) Expanded reporting pool

Figure 2. Accuracy and clue timing on Permutation. (a) Protocol accuracy at $K = 32$. (b,c) Early-clue minus late-clue accuracy for the prior and expanded reporting pools, in percentage points. Outlined cells reuse the main comparison's 200 worlds. Other cells use 100 worlds.

costs resolving ties. The chosen configuration remains fixed across structures and clue-timing conditions. **Appendix F** gives candidate scores, model-specific configurations, and generation settings.

The main setting uses $K = 32$, communication budget $b = 64$, generation budget $c = 65{,}536$, and 200 base worlds with one model draw per condition. The grid spans $K \in \{8, 16, 32, 64, 128\}$ and $b \in \{48, 64, 96, 128\}$, using 100 worlds per cell except the main setting, which reuses the main-comparison sample.

Communication counts complete messages and every transmitted copy. Generation includes thinking, revision, repair, and final answers from all calls. Invalid answers remain errors. We pair condition outcomes by world to estimate accuracy differences and pointwise 95% intervals (**Appendix M**).

4.2 Stronger reports and fixed-order exchange

At $K = 32, b = 64$, the Qwen3 reporter selected from the expanded pool uses fixed-position character encoding with private revision. We compare this reporter with the baseline selected from the earlier pool and also evaluate independent reporting at $b = 128$.

Expanding the reporting pool raises Permutation accuracy from 80% to 92%, a paired improvement of 12 points with interval $[7.5, 17.0]$ (**Table 1**). The estimated **Follow-up** advantage consequently falls from 16 to four points, with the remaining difference in $[-0.5, 8.5]$. Much of the estimated follow-up advantage can therefore be recovered by improving independent reporting.

Sequential reaches 97% accuracy, compared with 96% for **Follow-up**, giving a paired **Follow-up**–**Sequential** difference of -1 point with interval $[-3.5, 1.5]$. Mean generation is 1.20k tokens for **Sequential** and 5.00k for **Follow-up**. With the dependency direction known, the fixed exchange supplies the clue before B selects evidence. High accuracy can therefore be achieved through conditional disclosure along a predetermined order.

The lower generation cost of **Sequential** persists across the tested initial-message allocations for **Follow-up**. Initial reporting and revision account for 1.86k of the reference 3.80k generation gap, or 49% (**Appendix J**). Much of the difference arises during report preparation, before a follow-up query is issued.

4.3 CLUE TIMING AND TARGET COVERAGE

We hold the reporter fixed and deliver the selection clue before or after report preparation, charging clue transmission in both conditions. An irrelevant control clue matches the relevant clue's transmitted length. **Figure 2** reports the accuracy changes across assignment sizes and communication budgets.

At $b = 64$, the expanded reporter's early-clue gain increases from one point at $K = 16$ to 19 at $K = 64$ and 36 at $K = 128$. At $K = 128$, increasing b from 48 to 128 reduces the gain from 49 to eight points. The need to retain evidence for multiple possible requests becomes more restrictive as the assignment grows, consistent with the coverage distinction in **Proposition 2**.

To examine whether these gains reflect better target coverage, we annotate the transmitted entries and apply a fixed decoder using only messages received by D. At $K = 128, b = 64$, late-clue reports permit correct recovery in 61 of 100 worlds. Of the 39 failures, 26 omit the target entry, nine retain the entry but fail recovery, and four have unclassifiable messages. Early access reduces target omission to one world and raises recovery to 96. Increasing the late-clue budget to 128 instead reduces omission to two worlds and raises recovery to 92 (**Appendix I**). Earlier clues improve target availability through selective disclosure, while larger reports permit broader coverage.

At $K = 32, b = 64$, a complete representation fits, and reports with recognizable formats contain the target entry. Model accuracy is 92% with late access, 91% with the irrelevant clue, and 96% with the relevant early clue. Entry coverage alone does not ensure correct content or interpretation. The following interventions distinguish additional disclosure from improved use of evidence already received.

4.4 ADDITIONAL EVIDENCE AND DECISION-RECORD USE

We first hold the initial reports fixed to compare further processing with renewed private-evidence access. A separate initial record is generated with ceilings of eight tokens for A and 24 for B. Direct answering uses this record alone. Additional local reasoning gives D a private reasoning call before the final decision.

For the two reply conditions, D generates one forced query on each of the 200 worlds. The report-only and fresh-evidence conditions use this query. The report-only condition gives the selected responder the previously sent report and the query. The fresh-evidence condition also supplies that agent's private evidence. Query and reply ceilings are 12 and 20 tokens. The initial record, query, and reply allowance thus remain fixed when private-evidence access changes (**Appendix I.1**).

On Permutation, additional local reasoning raises accuracy from 75% to 80%, report-only replies reach 81%, and fresh-evidence replies reach 96% (**Table 2**). The paired fresh-evidence gain over report-only replies is 15 points with interval $[9.5, 21.0]$. Further processing improves decisions from the fixed record, while renewed private access supplies evidence that supports a larger gain. The reply-capacity comparison in **Figure A4** shows that this gain also depends on the space available to transmit that evidence.

We next examine how much model error can be corrected from the received record alone. We apply the same fixed decoder separately to each protocol's record, including the route clue available to D. A complete reference encoding provides a control for report construction (**Table 3**).

The reference encoding yields 100% decoder and 98% model accuracy, compared with 96% and 92% for generated Report_0 records. Errors remain both in constructing reports and in using them, even when a complete representation fits. The latter four-point difference estimates a lower bound on unused decision potential in **Equation** (11). With this decoder, **Sequential** and **Follow-up** each reach 99%, exceeding reporting by three points with intervals $[1.0, 5.5]$ and $[0.5, 6.0]$. Conditional disclosure improves recoverability, while terminal reading leaves part of that evidence unused. Joint reader outcomes are reported in **Appendix I**.

Table 2. Accuracy on separately generated fixed records over 200 worlds per structure, given as Cyclic / Permutation. Gains are relative to direct answering from the same record. The reply conditions share one forced query and the reply allowance. Bold and underlined accuracies mark the highest and second-highest values.

Intervention	Information access	Accuracy (%)	Gain (pp)
Direct answer	Initial record	95 / 75	+0 / +0
Additional local reasoning	Initial record	<u>96</u> / 80	+1 / +5
Report-only reply	Own report + query	<u>96</u> / <u>81</u>	+1 / +6
Fresh-evidence reply	Adds private evidence	**97 / 96**	+2 / +21

Table 3. Model and fixed-decoder accuracy (%) on 200 Permutation worlds at $K = 32, b = 64$. Both readers receive the same record. Among generated records, bold and underlined values mark the highest and second-highest accuracies.

Decision record	Model	Fixed decoder
Complete reference encoding	98	100
Report_0	92	<u>96</u>
Sequential	**97**	**99**
Follow-up	<u>96</u>	**99**

4.5 Model and document evidence

We repeat the routing comparison on 200 worlds with Qwen2.5-14B-Instruct and the native non-thinking template (Qwen et al., 2025), keeping the task and resource ceilings fixed. The **Follow-up–Sequential** difference is -2.5 points with interval $[-6.0, 1.0]$, while generation is 1.85k and 0.40k tokens, respectively. Fresh-evidence replies improve over report-only replies by 14 points with interval $[8.5, 19.5]$. The benefit of renewed evidence access and the lower generation cost of fixed-order disclosure both persist under this model (**Appendix L**).

To examine selection from document evidence, we give A a ticket and B 16 runbooks, using 200 worlds and $b = 128$. The decision requires the record matching the ticket despite conflicting versions. **Sequential** improves over reporting by 8.5 points with interval $[3.5, 14.0]$, using 2.90k rather than 4.10k generated tokens. Earlier access to the ticket clue supports focused disclosure when the relevant state must be identified within documents (**Appendix K**).

5 Conclusion

We studied the quality and cost of feedback relative to stronger independent reports and fixed-order communication. The analysis separates reporting loss from unused decision potential, characterizes the distinctions required for joint report coverage, and connects reconstruction error to the information available for a decision. These results explain how clue timing changes the evidence a report must preserve.

The experiments show that stronger reporting reduces the estimated follow-up advantage. Under known information dependence, a fixed exchange provides the selection clue before reporting and achieves high accuracy with lower generation cost. As the set of possible needs grows, earlier clues reduce target omissions. Fixed-report interventions and comparisons using the same decoder across protocols further distinguish gains from renewed evidence access from failures to use received information. Together, these findings support choosing communication procedures according to the coverage required before a clue arrives and the processing needed after evidence is received.

AI USE STATEMENT

We used generative AI tools to assist with language polishing, figure preparation, and checking mathematical proofs for potential errors or gaps. The authors reviewed all suggested edits and take full responsibility for the final manuscript.

REFERENCES

Mert Cemri, Melissa Z. Pan, Shuyi Yang, Lakshya A. Agrawal, Bhavya Chopra, Rishabh Tiwari, Kurt Keutzer, Aditya Parameswaran, Dan Klein, Kannan Ramchandran, Matei A. Zaharia, Joseph Gonzalez, and Ion Stoica. Why Do Multi-Agent LLM Systems Fail? In *Advances in Neural Information Processing Systems*, volume 38, pp. 135676–135729. Curran Associates, Inc., 2025.

Yulang Chen, Haoxuan Peng, Jinyan Liu, Zichen Wen, Dongrui Liu, and Linfeng Zhang. AgentSlimming: Towards Efficient and Cost-Aware Multi-Agent Systems. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 30064–30086. Association for Computational Linguistics, 2026.

Hyeong Kyu Choi, Xiaojin Zhu, and Sharon Li. Debate or Vote: Which Yields Better Decisions in Multi-Agent Large Language Models? In *Advances in Neural Information Processing Systems*, volume 38, pp. 101732–101764. Curran Associates, Inc., 2025.

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.

Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Zili Wang, Steven 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.

Junyou Li, Qin Zhang, Yangbin Yu, Qiang Fu, and Deheng Ye. More Agents Is All You Need. *Transactions on Machine Learning Research*, 2024.

Yuxuan Li, Aoi Naito, and Hirokazu Shirado. Systematic Failures in Collective Reasoning under Distributed Information in Multi-Agent LLMs. In *Forty-third International Conference on Machine Learning*, 2026.

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, 2024.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. Self-Refine: Iterative Refinement with Self-Feedback. In *Advances in Neural Information Processing Systems*, volume 36, pp. 46534–46594. Curran Associates, Inc., 2023.

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 *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 963–981. Association for Computational Linguistics, 2024.

Qwen, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, et al. Qwen2.5 Technical Report. *arXiv preprint arXiv:2412.15115, version 2*, 2025.

Qwen Team. Qwen3 Technical Report. *arXiv preprint arXiv:2505.09388*, 2025.

Tim Roughgarden. Communication Complexity (for Algorithm Designers), Lecture 2: Lower Bounds for One-Way Communication. Lecture notes, Stanford University, 2015.

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*, 2025.

Dat Tran and Douwe Kiela. Single-Agent LLMs Outperform Multi-Agent Systems on Multi-Hop Reasoning Under Equal Thinking Token Budgets. *arXiv preprint arXiv:2604.02460*, 2026.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-Consistency Improves Chain of Thought Reasoning in Language Models. In *International Conference on Learning Representations*, 2023.

Zhexuan Wang, Yutong Wang, Xuebo Liu, Liang Ding, Miao Zhang, Jie Liu, and Min Zhang. AgentDropout: Dynamic Agent Elimination for Token-Efficient and High-Performance LLM-Based Multi-Agent Collaboration. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 24013–24035. Association for Computational Linguistics, 2025.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, Ahmed Awadallah, Ryen W. White, Doug Burger, and Chi Wang. AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversations. In *Conference on Language Modeling*, 2024.

Wendi Yu, Lianhao Zhou, Xiangjue Dong, Sai Sudarshan Barath, Declan Staunton, Byung-Jun Yoon, Xiaoning Qian, James Caverlee, and Shuiwang Ji. When Do Multi-Agent Systems Help? An Information Bottleneck Perspective. *arXiv preprint arXiv:2607.16133*, 2026.

Yuting Zeng, Weizhe Huang, Lei Jiang, Tongxuan Liu, XiTai Jin, Chen Tianying Tiana, Jing Li, and Xiaohua Xu. S^2-MAD: Breaking the Token Barrier to Enhance Multi-Agent Debate Efficiency. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 9393–9408. Association for Computational Linguistics, 2025.

Jusheng Zhang, Yijia Fan, Kaitong Cai, Xiaofei Sun, and Keze Wang. OSC: Cognitive Orchestration through Dynamic Knowledge Alignment in Multi-Agent LLM Collaboration. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 6320–6337. Association for Computational Linguistics, 2025.

Xiaochen Zhu, Caiqi Zhang, Yizhou Chi, Tom Stafford, Nigel Collier, and Andreas Vlachos. Demystifying Multi-Agent Debate: The Role of Confidence and Diversity. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 33909–33930. Association for Computational Linguistics, 2026.

A INFORMATION ACCESS AND COMPARISON PROTOCOLS

This appendix describes the information access and communication order in **Section 2**. Protocols operate on the same task instance Z. The target Y is used only for evaluation. Message generation and evidence retrieval use permitted inputs and received information.

A.1 INITIAL CLUE VISIBILITY

In the basic setting, A and B access $I_A = (X, E_A)$ and $I_B = (X, E_B)$, while $I_D = X$. Selection clues originate in legitimate observations available to A and help identify which evidence held by B is relevant. Making the same clues available before reporting changes initial access while preserving the original evidence and target.

The private condition retains the clues in A's input. B-only additionally provides the same raw clues to B, leaving the initial inputs of A and D unchanged. Shared provides the clues to both B and D. Thus, private versus B-only tests earlier clue availability to the sender. Shared also changes the initial information available to the decision-maker.

These interventions share existing raw clues from permitted observations, with the evaluation target reserved for scoring. In deployment comparisons, actual transmission needed to provide clues to additional agents still incurs communication cost. Subsequent access to messages remains governed by the protocol.

A.2 COMMUNICATION PROCEDURES

The four protocols differ in when each agent receives information. **Report** and **Revise** use independent reporting. **Sequential** gives the first report to the second sender, while **Follow-up** lets D query an evidence agent after receiving the initial reports. Each responder retains access to the responder's initial evidence.

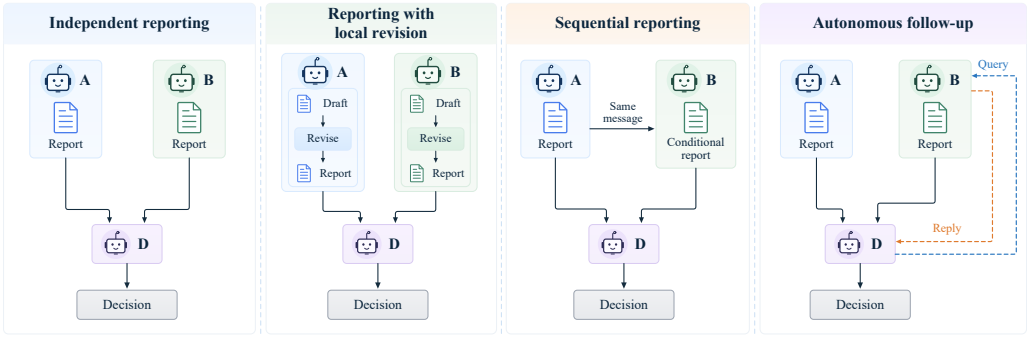


Figure A1. Information flow in the four comparison protocols. The follow-up panel shows a query to B, while either evidence agent may be selected.

Independent reporting with Report. Each evidence agent generates a report from the corresponding initial input, with task-aware evidence selection, compression, and organization permitted. Both reports are transmitted to D, which answers from the initial input I_D and received reports.

Private checking with Revise. Each evidence agent generates a draft, then checks and revises that draft using the permitted initial input. Only the final reports are transmitted to D. Drafting and revision both count toward generation cost. Checking uses only the agent's initial input and private draft.

Autonomous querying with Follow-up. After receiving independent reports, D either answers directly or queries one evidence agent. The selected agent responds using the initial evidence, local state, and query, after which D gives the final answer. The model chooses whether, whom, and what

to ask. Initial reports, the query, and the response count toward the total communication budget, with at most one query allowed.

Fixed-order reporting with Sequential. Using the $A \to B$ direction fixed during development, A generates a message and sends the same content to B and D. Agent B reports using the initial input I_B and this message. After receiving that report, D answers. The two transmissions of the message from A reuse a single generation but incur separate communication costs.

These allocations and dependencies define distinct feasible procedures whose quality and costs are compared jointly. The **Follow-up/Sequential** difference compares the tested complete procedures. Explaining these differences also requires clue interventions and report coverage checks.

A.3 FULL-EVIDENCE REFERENCE

FULL measures model accuracy with centralized access to X, E_A, E_B. The Bayes reference in **Section 3** evaluates the same information through an optimal decision rule.

B RESOURCE ACCOUNTING

We measure communication and generation separately to compare protocol accuracy with the resources used to achieve that accuracy.

B.1 COMMUNICATION AND GENERATION

Communication sums tokens in complete transmitted messages over actual transmission edges, including reports, queries, responses, and dynamic routing fields. Sending identical content to two recipients counts twice. Static public task instructions count toward input processing. Inter-agent communication counts the messages exchanged during execution.

Generation sums outputs from all model calls, including drafts, revision, reports, queries, responses, and final answers. Content generated once and sent to multiple recipients incurs generation cost once and communication cost for each copy. The model's tokenizer measures the complete transmitted messages.

The communication and generation limits are equal across protocols, but actual resource use may differ. Reporting actual usage shows how much of each budget was used and at what accuracy.

B.2 BUDGET CONSTRAINTS

Per-role allocations and total limits b and c are fixed before evaluation. Communication counts the transmitted text, while generation includes discarded drafts, compression, and repair calls. Invalid final answers remain in the accuracy denominator. **Appendix F** gives the message allocations and generation settings.

B.3 COST COMPARISONS

Independent reports permit parallel local processing. In **Follow-up**, querying and responding depend on initial reports, while **Sequential** conditions B's report on A's message. Costs are measured for each complete procedure. When an intervention reuses a report or query, the cost comparison includes the original preparation cost in every condition.

C BASELINE SELECTION AND INTERPRETATION OF COMPARISONS

We select reporting candidates on development data and compare their accuracy and costs on independent evaluation worlds. The component diagnostics hold selected parts of each procedure fixed to examine evidence access and decision use.

C.1 Development scope of the reporting baseline

Report candidates may use task-aware representations such as structured summaries, evidence indices, and conditional statements. A candidate specifies reporting prompts, checking procedures, and resource allocation. These choices and development effort define the scope of report optimization. Baseline Report_0 is selected from this **Report/Revise** candidate set by a prespecified development rule.

Selection uses development data. The selected protocol, model version, decoding configuration, and mapping from budgets to configurations are fixed before formal evaluation. Clue conditions use the same frozen baseline, separating the intervention from gains due to reselection. Intervals on development samples inform candidate selection and planning. Independent test samples provide the subsequent comparison.

C.2 Paired differences and substitution

Correctness is recorded for each protocol on the same instance, yielding within-instance **Follow-up**/Report_0 and **Follow-up/Sequential** differences. Interval estimation preserves this pairing. If an instance includes multiple clue or budget conditions, resampling retains them together. Mechanism comparisons estimate within-instance changes in feedback gains across conditions.

We compare reporting and **Sequential** with **Follow-up** using paired accuracy differences, uncertainty intervals, and actual costs. Resampling retains all compared outcomes from each world together. These estimates describe the quality–cost trade-off under the tested conditions.

An advantage of **Follow-up** over Report_0 supports gains from the tested feedback procedure over the selected reporting baseline. An advantage over **Sequential** measures the gain over a fixed message sequence. The size and interval of each difference, reported alongside costs, indicate the precision of these comparisons. Wide intervals leave the quality trade-off unresolved.

C.3 Scope of component diagnostics

Querying, disclosure, and final decisions may each limit feedback performance. Component replacements on the same instance preserve other inputs and generated content while changing a particular stage. For example, replacing the response procedure while keeping the model query fixed examines disclosure. Restricting the responder to already transmitted reports examines the contribution of renewed access to private evidence. Reference queries also use only their declared permitted information.

Such replacements measure system differences under specific interventions, each of which may affect both information preservation and use. Interpretation therefore accounts for the changed resources and information access.

D Conditions and proofs for the information boundary

We first state the information conditions for a complete decision record, then prove the accuracy decomposition and examine how computation and additional messages affect the two terms.

D.1 Objects and complete decision records

We consider accuracy over a finite answer set $\mathcal{Y}$. Instances (X, E_A, E_B, Y) follow a fixed population distribution. The protocol, model parameters, and prompts are fixed before evaluation on that distribution. Let $W = (X, E_A, E_B)$ denote the original evidence. Clues provided before reporting in the visibility interventions must be obtained from lawful evidence in W. If retrieval, tool observations, or other inputs provide additional answer information, the original evidence must be expanded to include those inputs.

The record V_π includes all instance-dependent information available for the final decision, including the initial input, received messages, and retained queries, drafts, random choices, or other computational state. Fresh randomness used to answer is conditionally independent of Y given this record.

The required conditional independence relations are expressed as

$$Y \perp V_\pi \mid W, \qquad Y \perp a_\pi \mid V_\pi. \tag{12}$$

The first condition excludes answer information acquired outside W while permitting randomized, history-dependent protocols and dependent evidence across agents. The second requires the record to include all answer-relevant information used in answering. Completeness also requires retained state that affects the interpretation of received messages. For example, an internal random choice may determine a query whose response can only be decoded using that choice. Applying the bound to a text log therefore requires recovering this state from the log.

All posteriors refer to the population distribution. Finiteness of $\mathcal{Y}$ ensures the maximum exists. A fixed label order resolves ties. Invalid outputs lie outside $\mathcal{Y}$ and count as incorrect in the upper-bound argument.

D.2 Accuracy decomposition in Proposition 1

Proof of Proposition 1 (accuracy decomposition). For any observation U, consider a decision a based on that observation and conditionally independent randomness. The decision's conditional probability of correctness is bounded as follows:

$$\Pr(a = Y \mid U) = \sum_{y \in \mathcal{Y}} \Pr(a = y \mid U) \Pr(Y = y \mid U) \le \max_{y \in \mathcal{Y}} \Pr(Y = y \mid U). \tag{13}$$

Taking expectations gives $\Pr(a = Y) \le \mathcal{B}(U)$. Choosing a posterior maximizer pointwise attains equality. The optimal accuracy supported by the observation is therefore $\mathcal{B}(U)$. Setting $U = V_\pi$ gives $\mathrm{Acc}_\pi \le \mathcal{B}(V_\pi)$.

By conditional independence and iterated expectation, the posterior probability of each label y can be written as

$$\Pr(Y = y \mid V_\pi) = \mathbb{E}[\Pr(Y = y \mid W) \mid V_\pi]. \tag{14}$$

Convexity of the maximum bounds the decision-record Bayes accuracy by the full-evidence reference as follows:

$$\begin{aligned} \mathcal{B}(V_\pi) &= \mathbb{E}\left[\max_y \mathbb{E}[\Pr(Y = y \mid W) \mid V_\pi]\right] \\ &\le \mathbb{E}\left[\mathbb{E}[\max_y \Pr(Y = y \mid W) \mid V_\pi]\right] = \mathcal{B}(W). \end{aligned} \tag{15}$$

Both gaps are therefore nonnegative. Adding and subtracting $\mathcal{B}(V_\pi)$ from $\mathcal{B}(W) - \mathrm{Acc}_\pi$ proves **Eq.** (5). □

This identity separates reporting loss from unused decision potential for one protocol. A component replacement may change both gaps.

The reporting-loss term is specific to accuracy. Equality $\mathcal{B}(V_\pi) = \mathcal{B}(W)$ can hold even when a report changes the posterior or omits original evidence. For example, let $Y \in \{0, 1\}$ and let W have two equally likely values with $\Pr(Y = 1 \mid W)$ equal to 0.6 and 0.9. A constant report supports always choosing label 1, matching the full-evidence Bayes accuracy of 0.75 despite losing confidence information.

D.3 Processing fixed inputs and disclosing more evidence

Let C denote a computational result obtained from a complete input U. Under the conditional independence relation $Y \perp C \mid U$, the posterior and Bayes accuracy are preserved as follows:

$$\Pr(Y = y \mid U, C) = \Pr(Y = y \mid U), \qquad \mathcal{B}(U, C) = \mathcal{B}(U). \tag{16}$$

If only the computational result C is retained and the original input U discarded, the same condition guarantees only $\mathcal{B}(C) \le \mathcal{B}(U)$. Preserving the bound and approaching that bound with an actual model are separate matters. Increased generation, reasoning rounds, or reformulation need not improve accuracy.

Appending a message M while retaining the input U instead gives the following Bayes-accuracy bound:

$$\mathcal{B}(U, M) \geq \mathcal{B}(U), \tag{17}$$

because a decision based on (U, M) can ignore M and reuse an optimal rule for U. Equality holds when a message is redundant or changes the posterior while preserving optimal decision quality. The additional message can also distract an actual model.

This weak monotonicity applies only to nested observations. With the same communication limit as reporting, **Follow-up** may shorten initial reports to reserve capacity for a query and response. **Sequential** changes initial recipients and content. These changes produce potentially nonnested final records. Comparing them requires evaluating the two complete-protocol accuracy differences in **Section 2**.

D.4 Bayes references and empirical diagnostics

If the task input X determines the answer through the almost-sure relation $Y = f(X)$, every complete record containing X has $\mathcal{B}(V_\pi) = \mathcal{B}(W) = 1$. The reporting loss is then zero, $L_\pi = 0$, and the difficulty of evaluating f contributes to unused decision potential. The decision record must retain the task input actually available to the model. An informational account of distributed evidence requires a generating distribution in which local evidence and answers remain uncertain given task input X. Computational difficulty is assessed through system experiments.

FULL measures one model given all evidence. FULL's accuracy difference from a restricted system is not $\mathcal{B}(W) - \mathcal{B}(V_\pi)$, because both systems may use their inputs imperfectly, and input length and computation conditions also differ. Improving the decision procedure while holding reports fixed demonstrates recovery of some unused potential, with further recoverable gains potentially remaining. A checked representation that recovers answers for all declared lawful needs shows that the required information can be preserved within that scope. A model's failure to decode that representation does not establish information loss.

The decomposition directs empirical checks toward information preservation and evidence use. Estimating the ideal Bayes terms or assigning individual errors to them requires additional assumptions. Component comparisons instead report their specific effects, information permissions, and costs.

D.5 Protocol comparison in Proposition 1

Proof of Proposition 1 (protocol comparison). **Proposition 1** applies to two protocols on the same population distribution and the same original evidence W. Each protocol must satisfy **Equation** (12) with the protocol's complete decision record. The preceding proof gives $L_\pi, U_\pi, L_{\mathrm{Follow}}, U_{\mathrm{Follow}} \geq 0$. Substituting $\mathrm{Acc}_\rho = \mathcal{B}(W) - L_\rho - U_\rho$ for $\rho \in \{\pi, \mathrm{Follow}\}$ and cancelling the two occurrences of $\mathcal{B}(W)$ proves **Equation** (6). □

The identity applies to potentially nonnested decision records and independently randomized protocols sharing the same task distribution and original evidence. Applying the identity to a protocol with additional external observations requires expanding the evidence reference. Component interventions can affect both terms, so their observed effects cannot be assigned to the terms without further identification assumptions.

E Coverage references and recovery proofs

We specify the scope of coverage checks and separate representation validity from budget feasibility. The final subsections prove the exact and approximate recovery results used in **Section 3.2**.

E.1 Objects and scope of a coverage check

Section 3.2 fixes X and represents the diagnostic need determined by A's observation as q. The required state $g(q, E_B)$ is determined by the need q and B's evidence. The task must first specify lawful evidence states, needs, and the required-state function g. Deriving and communicating q

from A's observation remain explicit protocol operations, with their input and transmission costs included.

For a small illustration, suppose routes 1, 2, and 4 have components C_2, C_4, C_8 and rule τ, while route 3 has C_7 and rule τ'. A report can list all four entries or encode the rule that route j maps to C_{2j} and τ, with route 3 as an exception. Both representations cover all four needs. If the clue identifies route 3 before reporting, only the component and rule assigned to route 3 are required. Transmitting either representation requires including the relevant instance parameters and exceptions. Tokenizer accounting determines the resulting cost.

The candidate encoder r and decoder are fixed before checking. Encoding accesses only B's permitted input. Task input X is omitted from function arguments because X is fixed. Each checked evidence state is tested against all declared lawful needs, without selecting needs by model success. Decoding may use declared public task semantics, but instance-specific parameters, identifiers, exceptions, and mappings must be included in transmitted messages.

If two lawful evidence states produce the same report and a need lawful for both requires different states, that representation cannot cover both. The collision invalidates that representation for the specified scope. Both passing and failing checks concern the declared encoder, decoder, and lawful pairs. Length requirements across representations need a separate argument.

Within the declared task scope, suppose $Y = f(X, q, g(q, E_B))$ and the decision record V_π retains the task input X, need q, and a report satisfying **Equation** (7). Decoding and applying f recovers Y, so $\mathcal{B}(V_\pi) = 1$. Because the need q comes from agent A's lawful observation, the original evidence W also determines the target Y, giving $\mathcal{B}(W) = 1$ and zero information loss in **Equation** (5). Sufficiency and coverage must hold on the task distribution's support. Finite development checks validate only the cases checked.

Joint coverage requires exact recovery for all declared needs and is stronger than preserving optimal decision accuracy. Failed reconstruction may leave the optimal answer unchanged or concern a need outside the distribution's support. The effect of failed reconstruction on accuracy depends on how the unrecovered state affects the optimal answer. **Appendix D.2** gives an example where optimal accuracy is preserved but the posterior changes.

E.2 Feasibility, generation, and reading

Budget feasibility requires counting complete transmitted messages, including instance parameters, mappings, routing fields, and formatting. Candidate length must fit both the role allocation and the total budget with other messages. **Sequential** still charges both transmissions of the clue message. A checked representation's length supplies a feasible upper bound on the communication needed for that construction.

Validity of a structural intervention is assessed from how required states depend on evidence and from coverage checks before model answers are observed. When rules that apply to multiple needs require only a few parameters, exceptions must be counted. When claiming dependence on separate details, those details must vary lawfully and affect the required states. This check separates changes in distinctions among required states from changes in wording or input redundancy.

Model experiments subsequently test whether an effective report is generated and correctly used by the receiver. Reference encoders support validation or diagnosis. Reports, queries, and responses in the main comparison remain model-generated under declared permissions.

E.3 Proof of Proposition 2

Fix the task input X, finite need and evidence sets $\mathcal{Q}, \mathcal{E}$, and a nonempty lawful scope $\Lambda \subseteq \mathcal{Q} \times \mathcal{E}$. The reporting rule $r\colon \mathcal{E} \to \mathcal{M}$ is deterministic and fixed before the instance. The required-state function g is defined on Λ. **Proposition 2** characterizes a particular report without requiring every need–evidence pair to be lawful.

Proof of Proposition 2. If a decoder satisfies **Equation** (7), two lawful pairs $(q, e), (q, e')$ with $r(e) = r(e')$ give the same decoder input. Their required states must therefore agree. Conversely, for each (q, m) with a nonempty set of lawful preimages $\{e\colon (q, e) \in \Lambda,\ r(e) = m\}$, the collision

condition makes $g(q, e)$ constant on that set. Define the decoder to be that constant. On pairs with no lawful preimage, define the decoder arbitrarily. The constructed decoder recovers the required state on every pair in Λ. □

The criterion is about recovering the declared state. If different states yield the same final answer, failure of this stronger recovery requirement need not reduce optimal answer accuracy. If the required state is constant, a constant report suffices. A non-Cartesian lawful scope may reveal evidence through q, and the criterion automatically accounts for this by considering only needs lawful for both evidence states.

E.4 Proof of the report-alphabet counts

The collision criterion yields a classical counting interpretation under additional assumptions. Fix the task input X and finite nonempty need and evidence sets $\mathcal{Q}, \mathcal{E}$, and assume every pair is lawful. A state profile lists all states that might be requested from an evidence state. We define the profile, the number of distinct profiles, and the number of distinct required states at each need as follows:

$$h(e) = (g(q, e))_{q \in \mathcal{Q}}, \qquad N = |\{h(e) \colon e \in \mathcal{E}\}|, \qquad N_q = |\{g(q, e) \colon e \in \mathcal{E}\}|. \tag{18}$$

Only the number of possible report symbols is charged in this ideal model. Encoders and decoders have unrestricted computation and arbitrary codebooks fixed from the task family before the instance. The decoder knows q in both cases. Let M_{before} be the smallest report-alphabet size permitting exact recovery when the encoder does not know q, and M_{after} the corresponding size when the encoder also knows the need. Their minimum sizes are given by **Equation** (8). The proof uses the classical deterministic distinguishability argument for one-way communication and Index (Roughgarden, 2015).

Proof of Equation (8)**.** Two distinct profiles differ at some need q. Because every pair is lawful, that need is possible for both evidence states, so **Proposition 2** requires distinct messages. Thus every exact encoder without q needs at least N symbols. Assigning a distinct symbol to each profile attains N, since the decoder selects the profile's q coordinate. When the encoder knows q, the N_q distinct required states require at least N_q symbols for that need, giving the lower bound $\max_q N_q$. A separate injective assignment of those values to the same alphabet for each q attains the bound. The decoder uses q to select the assignment. These constructions also cover $N = 1$ and $N_q = 1$. □

For a routing family with $K \geq 1$ routes, K distinct component indices, and a fixed active rule, consider two declared assignment scopes. All cyclic shifts give K profiles, whereas all permutations give $K!$ profiles. For a fixed route, every component can occur in both scopes. This gives **Equation** (9). The strict difference $K! > K$ starts at $K = 3$. These full-family counts hold with the other task fields fixed so that the selected component and rule form the required state. They concern the entire declared lawful family. The criterion is exact recovery for every need. Average-error performance and natural-language efficiency require separate analyses.

Fixed-length binary labels could represent these alphabets using $\lceil \log_2 M \rceil$ bits in the ideal model. This statement excludes the cost of delivering q, routing, and implementing or communicating a codebook. For deployment, explicit message and tokenizer costs determine feasibility within **Report-64**. **Sequential** and **Follow-up** both provide the clue before conditional disclosure and share the corresponding ideal count. Their practical difference concerns execution and cost.

E.5 Proof of Proposition 3 and an additional gain bound

For **Proposition 3**, allow the task input X to vary across the population. For almost every value of X, the need $q = q(X, E_A)$ is determined by lawful evidence and the target satisfies $Y = f(X, q, g_X(q, E_B))$. The notation g in the main text suppresses this dependence on X. The actual report R_π may be randomized. The triple (X, q, R_π) is recoverable from the complete record V_π. Fix a measurable decoder $d(X, q, R_\pi)$, defined on every possible report with a failure value for malformed inputs. Extend the answer map to that value by an arbitrary fixed answer. All protocol parameters and decoder rules are fixed for the population being analyzed.

Proof of Proposition 3. Apply the decoder and answer map to construct an answer $\widehat{Y}$ from V_π. Whenever $d(X, q, R_\pi) = g_X(q, E_B)$, the deterministic target relation gives $\widehat{Y} = Y$. The constructed answer and the Bayes reference therefore satisfy the following accuracy bounds:

$$\Pr(\widehat{Y} = Y) \geq 1 - \epsilon_\pi, \qquad \mathcal{B}(V_\pi) \geq \Pr(\widehat{Y} = Y) \geq 1 - \epsilon_\pi. \tag{19}$$

Because the need q is determined by the lawful input (X, E_A), the original evidence W determines the answer. The full-evidence Bayes accuracy is therefore $\mathcal{B}(W) = 1$, which gives the reporting-loss bound $L_\pi \leq \epsilon_\pi$. The unused potential consequently satisfies $U_\pi = 1 - \mathrm{Acc}_\pi - L_\pi \geq 1 - \mathrm{Acc}_\pi - \epsilon_\pi$. Combining this lower bound with nonnegativity yields the second inequality in **Proposition 3**. □

Additional bound on feedback gains. For comparison with **Follow-up**, apply the assumptions in **Equation** (12) to **Follow-up** using the original evidence W from protocol π, giving $L_{\mathrm{Follow}}, U_{\mathrm{Follow}} \geq 0$. The gain decomposition in **Equation** (6) then bounds the accuracy improvement as follows:

$$\mathrm{Acc}_{\mathrm{Follow}} - \mathrm{Acc}_\pi = L_\pi - L_{\mathrm{Follow}} + U_\pi - U_{\mathrm{Follow}} \leq L_\pi + U_\pi \leq \epsilon_\pi + U_\pi. \tag{20}$$

The result covers random reports and both endpoints $\epsilon_\pi = 0, 1$. Reconstruction error can exceed reporting loss when distinct states share an answer or another decoder succeeds. Since $\epsilon_\pi + U_\pi \geq L_\pi + U_\pi = 1 - \mathrm{Acc}_\pi$, the total-gain inequality is not sharper than the elementary accuracy ceiling. The total-gain inequality separates reporting loss constrained by reconstruction from possible improvements in evidence use. Exact coverage with an incorrect terminal decision can still permit an accuracy gain under **Follow-up**. A smaller reconstruction error need not reduce measured gains if model use deteriorates. A reference encoder does not establish ϵ_π for model-generated reports, and unrecognized syntax is a failure of the declared total decoder rather than evidence of information omission. Recognition rates must be reported separately. A diagnostic that supplies the evaluator's selected route when that route is unavailable from the decision record does not meet the proposition's conditions.

F Reporting candidates and development outcomes

F.1 Expanded reporting pool

The expanded pool uses 64 independent development worlds, separate from evaluation. Each routing world has Cyclic and Permutation assignments, giving 128 paired condition outcomes. Each document world has clean, version-conflict, and dense-collision variants, giving 192 outcomes with 64 independent sampling units. All candidates receive the same worlds within a task. We select the highest pooled answer accuracy, resolving ties by lower mean communication, then generation, then candidate identifier.

Routing candidates combine **Report** or **Revise** with decimal lists (LIST), fixed-width indices (FIXED), variable-width indices (VAR64), or validated shift summaries with a decimal-list fallback (STRUCT). Document candidates use compact keyed records (KEY), JSON rows, or labeled text records (TEXT), each preserving the service, region, release, rule, and component fields.

Table A1. Routing development scores for the expanded pool. Each Cyclic / Permutation count is out of 64. Pooled accuracy combines the two assignments, with bold values marking the selected reporting candidate for each model.

Candidate	Qwen3 C / P	Pooled (%)	Qwen2.5 C / P	Pooled (%)
Report / LIST	61 / 51	87.5	60 / 55	89.8
Revise / LIST	62 / 55	91.4	60 / 57	**91.4**
Report / FIXED	62 / 58	93.8	59 / 53	87.5
Revise / FIXED	63 / 60	**96.1**	60 / 55	89.8
Report / VAR64	61 / 56	91.4	59 / 51	85.9
Revise / VAR64	62 / 59	94.5	60 / 54	89.1
Report / STRUCT	60 / 47	83.6	58 / 45	80.5
Revise / STRUCT	62 / 50	87.5	59 / 48	83.6

Table A2. Document development scores. Each count is out of 64 base worlds reused across the three distractor conditions. Bold marks the selected reporting candidate.

Candidate	Clean	Version	Dense	Pooled (%)
Report / KEY	58	52	48	82.3
Revise / KEY	59	54	50	**84.9**
Report / JSON	59	50	46	80.7
Revise / JSON	59	53	48	83.3
Report / TEXT	58	49	44	78.6
Revise / TEXT	59	50	46	80.7

Table A3. Selected reporting configurations for the main comparisons.

Config ID	Task / model	Reporting candidate	Development score
Q3-R32	Routing / Qwen3	Revise / FIXED	123/128 (96.1%)
Q25-R32	Routing / Qwen2.5	Revise / LIST	117/128 (91.4%)
Q3-D16	Documents / Qwen3	Revise / KEY	163/192 (84.9%)

The core selections are **Revise**/FIXED for Qwen3 routing, **Revise**/LIST for Qwen2.5 routing, and **Revise**/KEY for documents. The selected format is also used by the corresponding complete protocols. **Sequential** uses conditional **Report**, while the initial **Follow-up** reports use **Revise**. The candidate scores above select the independent-reporting comparator. The routing grid assignments appear in **Table A5** and remain fixed within each clue-timing comparison.

F.2 Earlier candidate development

Candidate development compares **Report**, **Revise**, **Sequential**, and **Follow-up** under two sets of reporting instructions, denoted original and compact. The compact instructions change reporting guidance while preserving the initial information available to each role. The reporting comparator is selected by pooling accuracy across both assignment structures, with communication and generation used to resolve ties. The selection rule is applied separately for each model, assignment size, and budget. The character-encoding candidates subsequently extend this pool.

Table A4 compares 12 configurations on eight development worlds, each evaluated under Cyclic and Permutation. The two structures and all configurations are paired by world. Candidate selection uses these development outcomes, while the main comparison uses independent evaluation worlds.

Table A4. Candidate-development accuracy and mean token costs. Accuracy is the number correct out of eight worlds per structure. Invalid counts are Cyclic / Permutation. Communication and generation are averaged over both structures, with generation in thousands of tokens. Bold entries mark the selected reporting, sequential, and follow-up configurations within the preceding pool.

Protocol	b	Candidate	Cyclic	Perm.	Invalid	Comm.	Gen. (k)
Report	64	original	6/8	5/8	0 / 2	42.6	5.67
Report	64	compact	4/8	5/8	1 / 2	39.6	6.08
Revise	64	original	7/8	4/8	0 / 1	38.1	6.42
Revise	**64**	**compact**	**8/8**	**4/8**	**0 / 3**	**37.2**	**7.80**
Sequential	64	original	8/8	7/8	0 / 0	38.2	1.18
Sequential	**64**	**compact**	**8/8**	**8/8**	**0 / 0**	**35.1**	**1.21**
Follow-up	**64**	**original**	**7/8**	**8/8**	**1 / 0**	**39.4**	**5.87**
Follow-up	64	compact	6/8	7/8	2 / 0	35.7	4.31
Report	128	original	8/8	7/8	0 / 0	71.6	2.39
Report	128	compact	8/8	7/8	0 / 0	66.5	2.63
Revise	128	original	8/8	8/8	0 / 0	67.9	3.74
Revise	**128**	**compact**	**8/8**	**8/8**	**0 / 0**	**59.3**	**3.96**

At $b = 64$, **Revise** with compact instructions has the highest pooled reporting accuracy, solving all eight Cyclic worlds and four Permutation worlds. **Sequential** with compact instructions solves all eight worlds in both structures. The selected **Follow-up** configuration uses the original instructions and solves seven Cyclic worlds and eight Permutation worlds. These candidates define the earlier comparison pool. The expanded reporting results in the main text additionally include fixed-position character encoding, and their percentages are kept separate from these development counts.

F.3 Representation feasibility

A public alphabet of 32 characters represents component indices by position in the report. The encoding uses only the reporting agent's permitted evidence. Constructive checks on eight new worlds under both structures recover all 32 required states in every case, giving 512 successful checks. Complete communication, including the route message and routing prefixes, costs 21–26 tokens for Cyclic and 34–39 for Permutation, within the allocations at $b = 64$. The Qwen3-14B and Qwen2.5-14B-Instruct tokenizers (Qwen Team, 2025; Qwen et al., 2025) give the same counts on these messages.

These checks establish that a complete representation fits the budget on the eight-world feasibility set. Model-generated reports additionally require correct construction and interpretation. The reference profiles below extend the representation check to other assignment sizes.

F.4 Reporting formats and configuration assignments

Table A5 gives the reporting procedure and encoding selected at each assignment size and budget. Selection follows the pooled development criterion in **Appendix F.1**. The selected reporter remains fixed across clue-timing conditions.

Table A5. Reporting configurations for the routing comparisons. FIXED uses fixed-width component indices, and VAR64 uses a prefix-free variable-width code. These assignments specify the reporting procedure, separately from full-reference fit profiles.

Configuration	K	b	Workflow	Codec
C008-048	8	48	Revise	FIXED
C008-064	8	64	Revise	FIXED
C008-096	8	96	Report	FIXED
C008-128	8	128	Report	FIXED
C016-048	16	48	Revise	FIXED
C016-064	16	64	Revise	FIXED
C016-096	16	96	Report	FIXED
C016-128	16	128	Report	FIXED
C032-048	32	48	Revise	FIXED
C032-064	32	64	Revise	FIXED
C032-096	32	96	Report	FIXED
C032-128	32	128	Report	FIXED
C064-048	64	48	Revise	FIXED
C064-064	64	64	Revise	FIXED
C064-096	64	96	Report	FIXED
C064-128	64	128	Report	FIXED
C128-048	128	48	Revise	VAR64
C128-064	128	64	Revise	VAR64
C128-096	128	96	Revise	VAR64
C128-128	128	128	Revise	FIXED

FIXED represents component indices using a public 64-symbol alphabet, with one symbol for $K \leq 64$ and two base-64 symbols for $K = 128$. VAR64 uses one symbol for indices 0–63 and an escape followed by one symbol for indices 64–127. Reports include the active rule and encode either the full assignment, a prefix in source-route order, or the requested entry after a clue arrives. STRUCT represents assignments by a single cyclic shift when possible and otherwise uses a decimal vector.

Before clue delivery, entries remain in source-route order. An overlong report permits one charged compression attempt, followed by retention of the longest complete prefix that fits the message budget. After clue delivery, B may report the requested entry directly. **Revise** adds a private checking call before transmission.

Table A6. Full-reference communication profiles over 200 reference worlds per assignment size. The totals include both initial messages and routing overhead. Fit counts are evaluated for the named reference format and budget on this separate reference sample.

K	Format	Chars/index	Wire min / med / max	Fits at 48 / 64 / 96 / 128
8	Index8	1	12 / 14 / 18	200 / 200 / 200 / 200
16	Index16	1	20 / 23 / 29	200 / 200 / 200 / 200
32	Char32	1	34 / 37 / 43	200 / 200 / 200 / 200
64	Index64	1	58 / 65 / 79	0 / 93 / 200 / 200
128	Pair64	2	119 / 134 / 158	0 / 0 / 0 / 64

Pair64 represents a complete mapping, whereas the budgeted VAR64 procedure may retain only a prefix. The reference profiles measure whether complete representations fit. Record recovery is evaluated separately on the generated messages.

F.5 Message allocations

Let $a \leq 8$ be the complete cost of A's route message and let m_B be the cost of B's initial report. Independent reporting reserves eight tokens for A, giving B an allowance of $b-8$ without exposing A's content or message length. **Sequential** transmits two copies of A's message and gives B the remaining $b-2a$. **Follow-up** allocates at most $\lfloor (b-8)/2 \rfloor$ tokens to the initial report from B, at most 12 to the query, and the remaining $b-a-m_B-\ell_Q$ to the reply, where ℓ_Q is the complete query cost. These allowances include message fields and routing text.

At $b = 64$, the reference follow-up rule gives B an initial allowance of 28 tokens. The allocation comparison in **Appendix J.1** also evaluates initial allowances of 20 and 36 tokens, retaining the ceilings for A and the query and assigning the actual remainder to the reply.

The fixed-report interventions generate separate initial reports with ceilings of eight and 24 tokens, followed by a forced query with ceiling 12 and reply ceilings from eight to 20 (**Appendix I.1**). Document protocols reserve 24 tokens for A's ticket message. The document follow-up allocation gives B at most 52 initial tokens and the query at most 24, with the reply using the remaining communication budget. Public syntax is included in the serialized model prompt and charged as input processing.

F.6 Generation settings

Both models use temperature 0.6 and top-p 0.95, with top-k 20 for Qwen3 and disabled for Qwen2.5. Qwen3 uses the native thinking template, while Qwen2.5 uses the native instruction template. Repetition penalty is 1.0, and presence and frequency penalties are zero. We retain one sampled output per world and condition. The total generation ceiling is 65,536 tokens, with per-call limits in **Table A7**.

Table A7. Per-call generation ceilings. Thinking and visible output share the stated limit, and each call is also bounded by the remaining total allowance.

Call	Qwen3	Qwen2.5
Initial report	8192	4096
Local revision	8192	4096
Syntax or overflow repair	2048	1024
Query decision or forced query	2048	1024
Reply	4096	2048
Additional local reasoning	8192	4096
Final decision	4096	2048

Inputs are limited to 16,384 tokens within a 32,768-token context. Each report, query, or reply permits at most one repair call. Generation costs include discarded drafts and repairs, and input overflow and invalid final answers count as errors.

For each world, the report-only and fresh-evidence conditions use the same responder seed. Terminal-reader seeds are matched across the fixed-record interventions.

G PAIRED DEVELOPMENT OUTCOMES

Figure A2 retains the outcome of every development world and distinguishes invalid answers from valid but incorrect answers. This view shows whether configurations with similar aggregate accuracy solve the same instances. Mean generation includes both correct and incorrect outcomes.

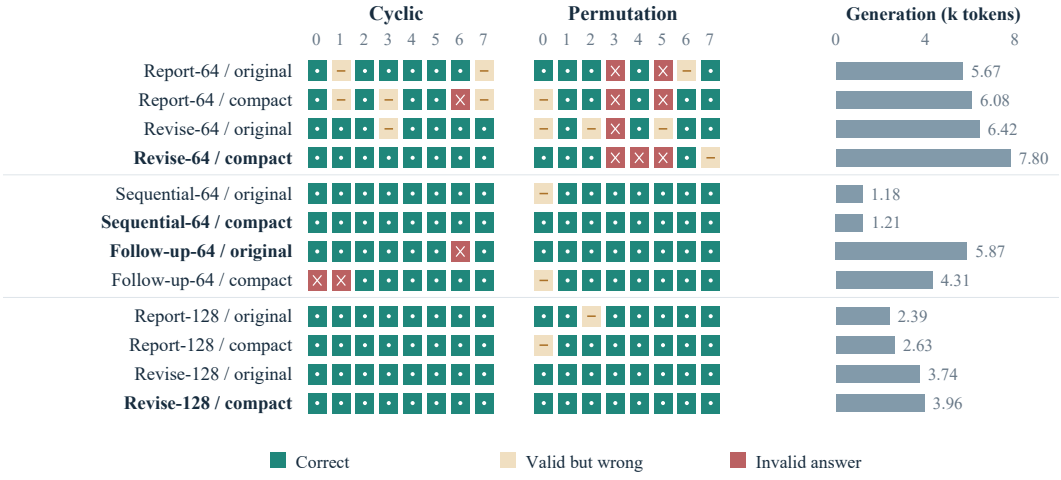


Figure A2. Development outcomes for the same eight worlds under each assignment structure. Bold labels mark the selected configurations within the preceding candidate pool. Generation costs are means over both structures.

Table A8. Changes from original to compact reporting instructions on the same development worlds. Gained / lost counts refer to newly correct / newly incorrect answers. Generation change is compact minus original on Permutation, in thousands of tokens.

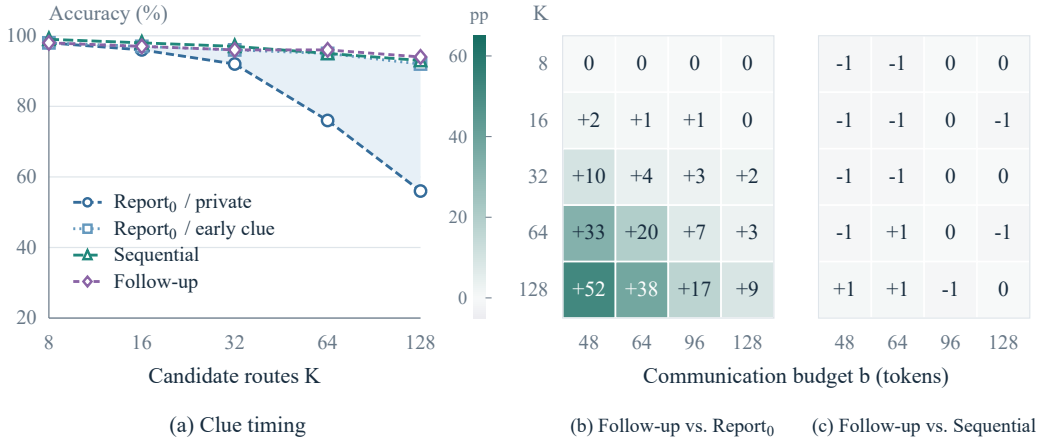
Protocol	b	Cyclic	Permutation	Gen. change (k)
Report	64	0 / 2	1 / 1	-0.62
Revise	64	1 / 0	2 / 2	+2.30
Sequential	64	0 / 0	1 / 0	-0.01
Follow-up	64	1 / 2	0 / 1	-2.72
Report	128	0 / 0	1 / 1	+0.35
Revise	128	0 / 0	0 / 0	+0.14

Table A8 makes these changes explicit. For **Revise** at $b = 64$ on Permutation, both configurations solve four worlds. However, the compact configuration gains two correct answers and loses two others, while increasing mean generation by 2.30k tokens. Equal aggregate accuracy therefore conceals differences in both the instances solved and the resources consumed. For **Follow-up**, compact instructions reduce generation by 2.72k tokens but lose one correct Permutation answer without gaining another. The paired outcomes show how the lower generation cost accompanies changes in the instances solved.

H Assignment size, communication, and clue access

H.1 Performance across assignment sizes and budgets

The extended comparison varies $K \in \{8, 16, 32, 64, 128\}$ and $b \in \{48, 64, 96, 128\}$. **Figure A3** reports the selected reporting comparator with late and early clue access, and compares **Follow-up** with both the reporting comparator and **Sequential**. Each cell compares procedures on the same task instances under the indicated budget. Cells use 100 base worlds except the $K = 32, b = 64$ setting, which shares the 200 worlds in the main comparison.



(a) Clue timing (b) Follow-up vs. Report$_0$ (c) Follow-up vs. Sequential

Figure A3. Extended results on Permutation. (a) Accuracy versus assignment size at $b = 64$. Shading shows the difference between early and late clue access. (b,c) Accuracy of **Follow-up** minus the selected reporting comparator and **Sequential**, respectively, in percentage points.

The difference between **Follow-up** and the reporting comparator grows as the assignment space expands under a small communication budget. At $K = 128$ and $b = 48$, the difference is 52 points, decreasing to nine points at $b = 128$. By comparison, differences between **Follow-up** and **Sequential** remain between minus one and one point across the displayed conditions. Both procedures disclose evidence conditionally on a selection clue, although one uses a fixed order and the other allows an autonomous query. These estimates concern the known dependency structure in **Section 4.3**.

H.2 Relevant and irrelevant clues

The clue-access control compares late access, an early irrelevant clue, and an early task-relevant clue. The irrelevant clue matches the delivery length of the relevant clue but contains no information about the target route. Clue transmission is included in the communication budget.

Table A9. Accuracy under different clue-access conditions at $K = 32$ and $b = 64$. Entries give Cyclic / Permutation. The reporter is fixed within each candidate pool.

Reporting pool	Late clue	Irrelevant clue	Early clue
Prior pool	96 / 80	96 / 81	97 / 95
Expanded pool	97 / 92	97 / 91	98 / 96

In Permutation, relevant early access improves the preceding reporting comparator from 80% to 95%, whereas an irrelevant clue yields 81%. For the expanded comparator, the corresponding accuracies are 92%, 96%, and 91%. The useful clue guides evidence selection, and the early-clue gain decreases after improving independent reporting. Within each pool, the comparison fixes the reporter, instructions, and computation allocation. Across pools, both the available representations and reporting procedures change.

H.3 Report recovery at selected assignment sizes

Table A10 compares record recognition, fixed-decoder recovery, and model decisions. Reporting uses **Revise** with FIXED at $K = 32, b = 64$ and $K = 128, b = 128$, and VAR64 at $K = 128, b = 64$. Each setting retains the same reporter across late and early clue access. Early access permits a targeted entry.

Table A10. Record diagnostics at selected routing settings. Late- and early-clue entries give recognized records / correct fixed-decoder answers / correct model answers out of the stated n. Full-reference fit counts have a separate denominator of 200 and refer to the formats in Table A6.

(K, b)	n	Full reference fits / 200	Late clue	Early clue
(32, 64)	200	200	196 / 192 / 184	199 / 198 / 192
(128, 64)	100	0	96 / 61 / 56	98 / 96 / 92
(128, 128)	100	64	98 / 92 / 87	99 / 98 / 95

At $K = 32, b = 64$, a full reference fits, but generated records and model answers do not attain the reference decoder's accuracy. At $K = 128, b = 64$, a recognizable prefix may omit the requested assignment. Early access allows selection of that assignment before reporting. Raising the budget to 128 increases late-clue recovery to 92% and model accuracy to 87%, while the early-clue values reach 98% and 95%. **Appendix I** separates target omissions from covered recovery failures and gives the joint outcomes of the two readers.

I Target coverage, record recovery, and evidence access

I.1 Fixed-report intervention

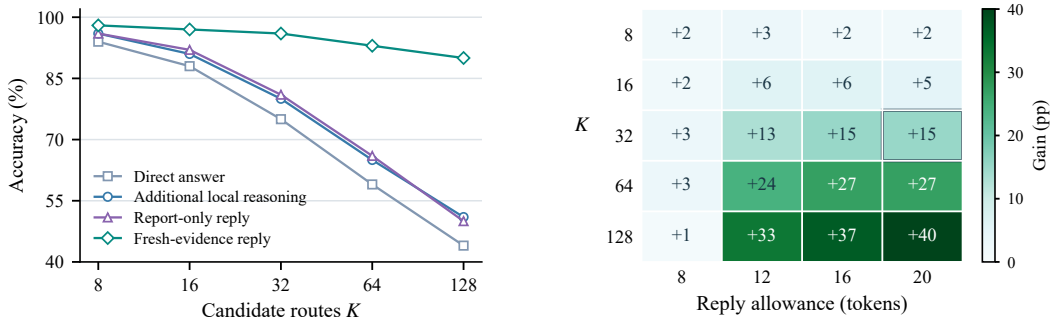
For each evaluation world, we generate a separate initial record with message ceilings of eight tokens for A and 24 for B, using the selected reporting family. These records are held fixed across interventions. The complete **Follow-up** comparison instead uses initial ceilings of eight and 28 tokens at the reference setting.

Decision-maker D generates one forced query per world. Both reply conditions and all reply allowances use the same recipient and question. Every world enters both conditions. By comparison,

the complete protocol queries on 78% of Permutation worlds, comprising 156 queried and 44 directly answered worlds.

Both reply conditions give the selected responder the initial report and the query. The fresh-evidence condition also provides the agent's private evidence. Direct answering uses the initial record alone, while additional local reasoning gives D a private reasoning call before the final answer. Each condition produces a separate final answer, and costs include the preparation of reused reports and queries.

Both reply conditions use the same repaired query when a repair is needed. Failed queries count as incorrect in both conditions. The second-model diagnostic uses the same intervention design with that model's selected reporting family.



(a) Fixed-report interventions

(b) Value of private-evidence access

Figure A4. Fixed-report interventions on Permutation with one forced query per world. (a) Accuracy versus assignment size with a 20-token reply allowance. (b) Fresh-evidence minus report-only accuracy across reply allowances. Initial-record and query ceilings are 32 and 12 tokens. The $K = 32$ condition uses 200 worlds and other sizes use 100.

With an eight-token reply allowance, fresh-evidence access improves accuracy by one to three points across assignment sizes. With 20 tokens, the gain reaches 15 points at $K = 32$ and 40 at $K = 128$. Renewed access benefits the decision when the reply can transmit the needed evidence.

I.2 Decoding actual decision records

The fixed decoder reads the public task and the messages received by D. The decoder identifies the route from A's message and recovers the corresponding component under the rule transmitted by B. Missing, malformed, or contradictory information results in abstention. Ground truth is used only for scoring and coverage annotation. Unrecognized records are classified separately from target omissions.

I.3 Target coverage under late and early clues

For the settings in **Table A10**, each record is classified as successful recovery, target absent, covered failure, or unclassifiable. Target absence means that the required source entry was omitted. A covered failure retains the entry but does not yield the correct decoder answer, for example because the value or received selector is incorrect. Unclassifiable records do not support reliable coverage annotation. These annotations are used only for evaluation.

Table A11. Record categories as percentages of all n worlds in each condition. The four outcome categories partition each row and sum to 100. Bold values mark the higher recovery rate within each late–early pair. The reference-encoding sample in Table A10 is separate.

(K, b)	Clue	n	Unclassifiable	Target absent	Covered failure	Recovered
(32, 64)	Late	200	2	0	2	96
	Early	200	0.5	0	0.5	**99**
(128, 64)	Late	100	4	26	9	61
	Early	100	2	1	1	**96**
(128, 128)	Late	100	2	2	4	92
	Early	100	1	0	1	**98**

At $K = 128, b = 64$ with late clues, target absence accounts for 26 of the 39 recovery failures. Nine failures retain the target entry, while four messages are unclassifiable. Early clues reduce omission to one world and raise recovery from 61 to 96. Increasing the budget to 128 instead leaves two omissions and raises late-clue recovery to 92. The larger report preserves more entries before the need is known, whereas the earlier clue permits selection of the requested entry within the smaller budget.

At $K = 32, b = 64$, the recognized reports cover the target in both timing conditions. The late-clue condition still includes four unclassifiable messages and four covered failures among 200 worlds. Earlier clues reduce each count to one. Target coverage and correct recovery thus remain distinct even when the complete representation can fit.

I.4 Joint outcomes of the model and fixed decoder

The joint reader comparison applies both readers to the identical received record. A world can be answered correctly by both, by the decoder alone, by the model alone, or by neither. **Table A12** reports these outcomes for the late-clue conditions.

Table A12. Joint reader outcomes as percentages of all n late-clue worlds. Each row sums to 100. Both-correct plus decoder-only gives fixed-decoder accuracy, while both-correct plus model-only gives model accuracy.

(K, b)	n	Both correct	Decoder only	Model only	Both wrong
(32, 64)	200	91	5	1	3
(128, 64)	100	52	9	4	35
(128, 128)	100	85	7	2	6

At $K = 128, b = 64$, 52 worlds are solved by both readers, nine only by the decoder, four only by the model, and 35 by neither. The nine decoder-only cases identify recoverable answers missed by the model. The four model-only cases offset part of this difference, leaving a five-point net recovery advantage. At $K = 32, b = 64$, the corresponding difference is five minus one percentage points, matching the four-point gap in the main comparison. Coverage categories describe transmitted content, while these joint outcomes describe decisions on that content.

I.5 Recovery across complete protocols

Table A13. Model / decoder accuracy (%) using the same decision record. The decoder uses only information available to the decision-maker. Bold values give decoder accuracy.

Decision record	Cyclic	Permutation
Report_0 (prior pool)	96 / **98**	80 / **88**
Report_0	97 / **99**	92 / **96**
Sequential	98 / **99**	97 / **99**
Follow-up	97 / **99**	96 / **99**

Table A13 shows that the decoder exceeds model accuracy under every displayed configuration. On Permutation, the difference is eight points for the preceding reporting comparator and four for the expanded comparator. The corresponding differences are two points for **Sequential** and three for **Follow-up**. By **Equation** (11), these differences estimate lower bounds on unused decision potential, with the fixed decoder providing an attainable reference for each record.

The expanded reporting comparator's decoder accuracy of 96% equals **Follow-up**'s model accuracy in aggregate. The independent reporting record therefore supports a higher accuracy than the model realizes. Recovery by another decoder remains possible on unresolved records, so the remaining decoder errors do not identify reporting loss.

I.6 Corrections and newly introduced errors

For the fixed-report intervention, **Table A14** separates corrections of initially incorrect answers from errors introduced on initially correct answers. All four outcome categories remain in the denominator. The direct-answer accuracy is 95% for Cyclic and 75% for Permutation.

Table A14. Joint outcome shares (%) relative to direct answering from the same initial reports. Each row sums to 100. Correct-to-correct plus wrong-to-correct gives the corresponding final accuracy in Table 2.

Structure	Intervention	Correct to correct	Correct to wrong	Wrong to correct	Wrong to wrong
Cyclic	Additional local reasoning	94	1	2	3
Cyclic	Report-only reply	93	2	3	2
Cyclic	Fresh-evidence reply	93	2	4	1
Permutation	Additional local reasoning	73	2	7	18
Permutation	Report-only reply	72	3	9	16
Permutation	Fresh-evidence reply	73	2	23	2

On Permutation, fresh-evidence replies correct 23% of all instances while changing 2% from correct to incorrect, producing a net gain of 21 points over direct answering. Report-only replies correct 9% and introduce errors on 3%, producing a six-point net gain. The benefit of additional evidence therefore coexists with occasional degradation.

I.7 Reference reports and generated reports

The reference encoder uses the reporter's permitted evidence to produce a complete valid message. Reading the same reference with a model and the fixed decoder examines interpretation without requiring the model to construct the report. The generated-report condition instead uses the reports produced by Report_0 on the same worlds.

Table A15. Correct answers from reference and model-generated reports over 200 worlds per structure. Each entry gives model / fixed-decoder counts from the same decision record.

Report source	Cyclic model / fixed	Permutation model / fixed
Reference encoding	198 / 200	196 / 200
Model-generated report	194 / 198	184 / 192

On Permutation, the model solves 196 reference records and 184 generated records, while the fixed decoder solves 200 and 192, respectively. The reference condition confirms that the declared record can specify the answer. The generated condition permits both imperfect construction and failures to use what was transmitted.

Table A16. Attributes of the expanded reporting comparator's records over 200 worlds per structure. Full-mapping fidelity is evaluated against local evidence. Target recovery uses only the decision record. A partially incorrect mapping can still contain the correct requested entry.

Structure	Within budget	Recognized record	Faithful full mapping	Correct target recovery
Cyclic	200	199	196	198
Permutation	200	196	188	192

The diagnostic distinguishes a recognized record from a faithful full mapping. In Permutation, 196 records are recognized, 188 preserve the full mapping, and 192 permit correct target recovery.

I.8 Protocol comparisons using the same decoder

The same fixed decoder is applied separately to each protocol's received record, giving the decoder accuracies in **Table A13**. On Permutation, **Sequential** and **Follow-up** each reach 99%, compared with 96% for Report_0. Their paired differences from reporting are three points, with intervals $[1.0, 5.5]$ and $[0.5, 6.0]$, respectively. This remaining difference concerns recoverability under the specified reader. The accuracy difference under this decoder is not a direct estimate of the reporting-loss term in **Proposition 1**.

Table A17. Paired accuracy comparisons on 200 Permutation worlds. The upper block compares reporting pools using model decisions. The lower block applies the same fixed decoder to every protocol. Differences and intervals are in percentage points.

First − second	First only	Second only	Δ (pp)	95% interval
Reporting-pool comparison, model decisions				
Report_0 − prior pool	26	2	+12.0	$[+7.5, +17.0]$
Follow-up − prior pool	38	6	+16.0	$[+10.0, +22.0]$
All protocols with the fixed decoder				
Follow-up − Report_0	7	1	+3.0	$[+0.5, +6.0]$
Sequential − Report_0	6	0	+3.0	$[+1.0, +5.5]$
Follow-up − Sequential	1	1	0.0	$[-1.5, +1.5]$

The decoder diagnostic evaluates the received records without changing the costs of the original model procedures.

I.9 Direct comparison of reply access

Table A18 provides the joint outcomes of the two reply procedures. Unlike separate comparisons against direct answering, this pairing directly identifies their accuracy difference. In Permutation, fresh-evidence replies correct 34 cases that report-only replies miss and miss four cases that report-only replies solve. The net improvement is 30 of 200 worlds, or 15 percentage points.

Table A18. Joint reply outcomes under fixed reports, queries, and reply allowances. Fresh denotes fresh-evidence replies and record denotes report-only replies. Each row sums to 200. The difference is fresh minus record in percentage points.

Structure	Both correct	Fresh only	Record only	Both wrong	Δ (pp)	95% interval
Cyclic	190	4	2	4	$+1.0$	$[-1.5, +3.5]$
Permutation	158	34	4	4	$+15.0$	$[+9.5, +21.0]$

J Follow-up allocation and generation costs

J.1 Allocation between initial reports and replies

The initial report and the subsequent reply share the communication allowance. We compare three initial allowances for B on Permutation at $K = 32, b = 64$, retaining the reporting candidate families, the eight-token ceiling for A, and the 12-token ceiling for the query. The reply uses the actual remainder after all preceding transmissions and routing fields. The 28-token condition is the reference allocation in **Table 1**.

Table A19. Follow-up performance under three initial-report allowances for B. Query rates use all worlds, including those answered directly. Bold and underlined accuracies mark the highest and second-highest point estimates. Bold resource values and query rates mark the lowest values among the three allocations.

B initial allowance	Accuracy (%)	Query rate (%)	Communication	Generation (k)
20	95	88	54	5.60
28	**96**	78	56	5.00
36	<u>95.5</u>	**56**	**52**	**4.55**

Increasing the initial allowance from 20 to 36 tokens lowers query frequency from 88% to 56% and mean generation from 5.60k to 4.55k. Accuracy varies within one percentage point and is highest at the intermediate allowance. Additional initial space can reduce the need for a query while restricting what a later reply can transmit.

Sequential reaches 97% accuracy with 36 transmitted tokens and 1.20k generated tokens in the same setting. Generation under **Sequential** is 73.6–78.6% lower than the three follow-up means. The cost difference therefore persists across the tested allocations, although the allocations change the frequency and cost of later exchange.

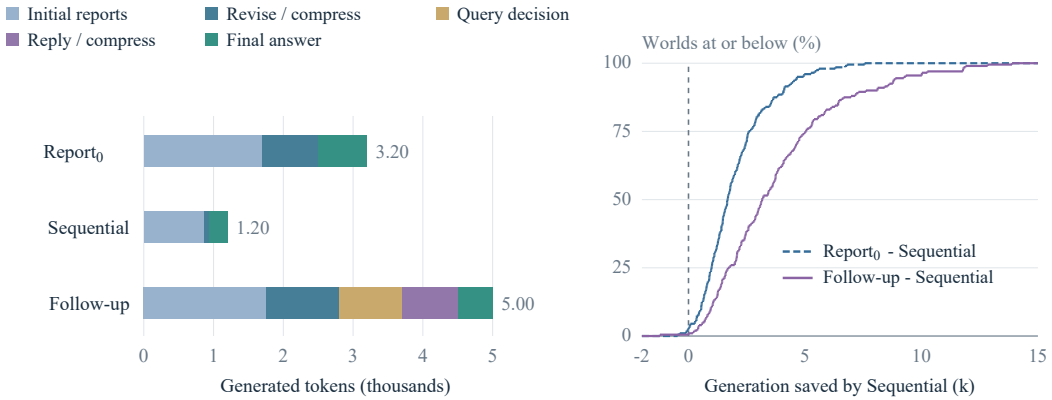
J.2 Costs for the main protocol comparison

The main Permutation comparison separates initial reporting, private revision or compression, query generation, reply generation or compression, and final answering. **Table A20** gives the stage means corresponding to the totals in **Table 1**.

Table A20. Mean generated tokens by stage for the main Permutation comparison, in thousands. All calls remain charged.

Stage	Report_0	Sequential	Follow-up
Initial reports	1.70	0.87	1.75
Revision and compression	0.80	0.07	1.05
Query decisions	0.00	0.00	0.90
Replies and compression	0.00	0.00	0.80
Final answers	0.70	0.26	0.50
Total	3.20	**1.20**	5.00

Initial reporting and revision account for 2.50k tokens under Report_0, 0.94k under **Sequential**, and 2.80k under **Follow-up**. The 1.86k difference between **Follow-up** and **Sequential** at these stages is 48.9% of their total 3.80k generation gap. The follow-up procedure additionally uses 0.90k for query decisions and 0.80k for replies and their compression. Final answers account for 0.70k, 0.26k, and 0.50k, respectively. This accounting locates the gap across stages of the tested implementations.



(a) Core-scenario generation by stage

(b) Distribution of paired savings

Figure A5. Generation costs for the main Permutation comparison. (a) Mean generation by stage. (b) Cumulative distributions over 200 paired worlds. Positive savings favor Sequential.

Table A21. Generation means, medians, and 90th percentiles, with mean communication, over 200 paired worlds. Generation is in thousands of tokens.

Protocol	Mean gen. (k)	Median (k)	90th pct. (k)	Mean wire
Report_0	3.20	2.84	5.63	37
Sequential	1.20	1.08	1.89	36
Follow-up	5.00	4.41	8.85	56

Table A22. Paired cost differences over 200 Permutation worlds. Negative values indicate lower resource use by Sequential. Generation is in thousands of tokens and communication is in tokens. Intervals use paired resampling of worlds.

First − second	Gen. Δ (k)	95% interval	Wire Δ	95% interval
Sequential − Follow-up	-3.8	$[-4.2, -3.4]$	-20.0	$[-20.4, -19.6]$
Sequential − Report_0	-2.0	$[-2.2, -1.8]$	-1.0	$[-1.4, -0.6]$

The main means give a 3.80k generation reduction for **Sequential** relative to **Follow-up** and a 2.00k reduction relative to reporting. The communication differences are 20 and one token, respectively. The paired analysis retains each world's protocol costs together during resampling.

J.3 Earlier development measurements

The earlier development comparison uses eight Permutation worlds. **Figure A6** and **Table A23** separate mean generation into reporting, checking or compression, query, reply, and final-answer stages. Costs include unsuccessful calls. This sample is separate from the main comparison and the candidate-development batch.

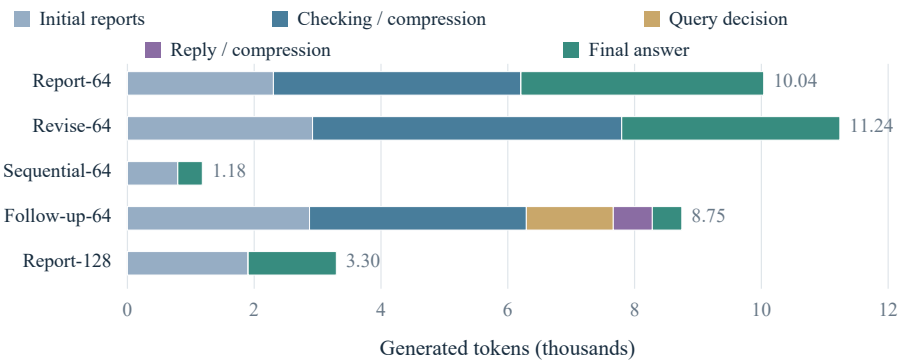


Figure A6. Mean generated tokens by execution stage on Permutation in the earlier development comparison. Initial reports and their checking or compression are shown separately. Reply costs include reply compression.

Table A23. Generation by execution stage, in thousands of tokens. Values include unsuccessful calls. Displayed components may differ from totals because of rounding.

Protocol	b	Initial	Check / comp.	Query	Reply / comp.	Final	Total
Report	64	2.30	3.91	0.00	0.00	3.83	10.04
Revise	64	2.92	4.88	0.00	0.00	3.44	11.24
Sequential	64	0.80	0.00	0.00	0.00	0.39	1.18
Follow-up	64	2.87	3.42	1.37	0.62	0.47	8.75
Report	128	1.90	0.00	0.00	0.00	1.39	3.30

Follow-up uses 8.75k generated tokens per instance, compared with 1.18k for **Sequential**. Initial reports consume 2.87k tokens and their checking or compression another 3.42k. More specifically, B's initial report and subsequent compression account for 5.9475k of the unrounded 8.7451k total, or 68.0%. Query generation accounts for 1.37k, and reply generation with compression accounts for 0.6159k. The large total therefore arises substantially before the reply is generated.

These costs compare complete procedures with different reporting instructions, message allocations, and communication sequences. The fixed-report intervention examines the contribution of renewed evidence access separately.

K Evaluation with document evidence

K.1 Task construction and information access

The controlled document task uses template-generated tickets and runbooks. Each of 200 base worlds contains 16 runbooks of 160–240 words and a ticket of 60–100 words. The selector is uniform, while component assignments and document order are shuffled. Agent A sees the ticket's service, region, release, and incident rule. Agent B sees the runbooks and component records. Decision-maker D receives only protocol messages and must identify the component in the unique record matching the ticket and active rule. Development and evaluation use separate base worlds.

A valid match must satisfy the service, region, release, and incident rule in the ticket. A superseded version remains a distractor even when the document marks that version as active.

All protocols use $b = 128$ and $c = 65{,}536$, with every transmitted copy and generated call charged. **Sequential** sends the ticket-derived clue to B and D before B reports. The primary condition contains four runbooks with the same service and region, of which only one matches the release and rule. The remaining three are superseded versions.

K.2 PRIMARY COMPARISON

Table A24. Accuracy and mean resource use on 200 document worlds with version distractors. All protocols use $b = 128$ and $c = 65{,}536$. Generation is in thousands of tokens. Bold and underlined accuracies mark the highest and second-highest values, and bold costs mark the lowest usage.

Protocol	Accuracy (%)	Communication	Generation (k)
Report_0 from prior pool	73.0	88	4.40
Report_0	82.5	78	4.10
Sequential	**91.0**	**76**	**2.90**
Follow-up	<u>90.5</u>	117	6.50

The expanded reporting pool raises accuracy from 73% to 82.5%, with mean communication decreasing from 88 to 78 tokens and generation from 4.40k to 4.10k. **Sequential** reaches 91% accuracy with 76 transmitted tokens and 2.90k generated tokens. **Follow-up** reaches 90.5% with 117 transmitted tokens and 6.50k generated tokens. The paired **Follow-up–Sequential** difference is -0.5 points with interval $[-5.0, 3.5]$. The fixed exchange obtains the ticket's selection information before reporting, reducing the need to preserve facts for other possible tickets.

Table A25. Paired comparisons in the primary document condition. First-only and second-only counts identify worlds solved by one procedure alone. All comparisons use the same 200 worlds. Differences and intervals are in percentage points.

First − second	First only	Second only	Δ (pp)	95% interval
Follow-up − Report_0	24	8	+8.0	$[+2.5, +13.5]$
Follow-up − Sequential	9	10	-0.5	$[-5.0, +3.5]$
Sequential − Report_0	23	6	+8.5	$[+3.5, +14.0]$

K.3 CHANGING DISTRACTORS WHILE RETAINING TARGETS

The clean, version-distractor, and dense-collision conditions reuse the same 200 base worlds, targets, and valid evidence. All three conditions retain 16 runbooks, with document lengths matched across conditions. Only distractor keys and version cues change. Clean distractors differ in service or region. In the primary condition, four records match the ticket's service and region. Dense collisions contain eight such records, including stale active flags, while retaining exactly one valid full-key match. The three conditions therefore preserve the dependency direction and the answer while changing the evidence-selection problem.

Table A26. Model / fixed-decoder correct counts out of 200 base worlds reused across the three distractor conditions. The same fixed-reader procedure is applied to every protocol's decision record.

Distractor condition	Report_0	Sequential	Follow-up
Clean keys	180 / 188	190 / 196	189 / 195
Version distractors	165 / 173	182 / 188	181 / 187
Dense collisions	152 / 163	172 / 180	171 / 179

Model accuracy decreases across these conditions for all procedures. The fixed decoder also loses accuracy, indicating reduced recoverability under the same decoder. The **Sequential**–Report_0 model difference is five points with clean keys, 8.5 with version distractors, and ten with dense collisions. **Appendix M** compares these gains directly.

K.4 TRANSMITTED EVIDENCE AND RESOURCE USE

Table A27. Attributes of transmitted records in the primary document condition. The three content categories sum to 200 per protocol. Evaluator annotations are not available to either the model or the decoder.

Record attribute	Report_0	Sequential	Follow-up
Correct version evidence present	177	191	190
Stale version only	16	6	7
Neither category	7	3	3

The correct-version evidence appears explicitly in 177 reporting records, 191 sequential records, and 190 follow-up records. A stale version alone appears in 16, six, and seven records, respectively. The fixed decoder correctly resolves 173, 188, and 187 records, while the models resolve 165, 182, and 181. These counts distinguish selecting and transmitting the requested version from using the received record.

Table A28. Mean generation for the three distractor conditions evaluated on the same base worlds, in thousands of tokens. All reporting, revision, query, reply, and final-answer calls remain charged.

Distractor condition	Report_0	Sequential	Follow-up
Clean keys	3.70	2.30	5.40
Version distractors	4.10	2.90	6.50
Dense collisions	4.90	3.90	7.90

The relative generation reduction of **Sequential** over reporting is 37.8% with clean keys, 29.3% with version distractors, and 20.4% with dense collisions. This pattern is consistent with continued work to select and use the correct version after the clue arrives. A larger accuracy difference can coexist with a smaller fractional generation reduction. These controlled conditions extend the evidence representation while retaining the known $A \to B$ dependence.

L SECOND-MODEL AND REFERENCE-FORMAT COMPARISONS

Table A29. Summary of the model and document comparisons on 200 worlds. Qwen2.5 routing uses Permutation at $K = 32, b = 64$, and the document comparison uses version distractors at $K = 16, b = 128$. Bold and underlined accuracies mark the highest and second-highest values. Bold costs mark the lowest usage.

Task	Protocol	Accuracy (%)	Communication	Generation (k)
Qwen2.5 routing	Report_0	87.0	40	1.10
	Sequential	**94.0**	**37**	**0.40**
	Follow-up	<u>91.5</u>	59	1.85
Document evidence	Report_0	82.5	78	4.10
	Sequential	**91.0**	**76**	**2.90**
	Follow-up	<u>90.5</u>	117	6.50

L.1 PROTOCOLS AND FIXED-REPORT INTERVENTIONS

Qwen2.5-14B-Instruct uses the native non-thinking template (Qwen et al., 2025). The comparison retains the routing task, world pairing, role permissions, and resource ceilings of the primary model. Reporting candidates are specified separately for each model. Each structure uses 200 worlds at $K = 32, b = 64, c = 65{,}536$. The two models belong to different generations of the Qwen series.

Table A30. Qwen2.5-14B-Instruct correct counts and mean costs at the main routing setting. C / P denotes Cyclic / Permutation. Generation is in thousands of tokens.

Protocol	Cyclic correct	Permutation correct	Wire (C / P)	Gen. k (C / P)
Report_0	188/200	174/200	26 / 40	0.55 / 1.10
Sequential	193/200	188/200	39 / 37	0.30 / 0.40
Follow-up	192/200	183/200	30 / 59	0.74 / 1.85

In Permutation, the paired gain of **Sequential** over reporting is seven points with interval $[3.0, 11.0]$. The **Follow-up**–reporting difference is 4.5 points with interval $[0.0, 9.0]$, and the **Follow-up**–**Sequential** difference is -2.5 points with interval $[-6.0, 1.0]$. Applying the same fixed decoder separately to each protocol gives 191, 196, and 195 correct answers for reporting, sequential exchange, and follow-up, respectively, corresponding to 95.5%, 98%, and 97.5%. In Cyclic, the corresponding decoder counts are 196, 198, and 197. As with Qwen3, correct use of received information remains relevant to the protocol comparison.

Table A31. Fixed-report interventions with Qwen2.5-14B-Instruct. Each structure uses 200 worlds. The reply procedures retain the same query and reply allowance, while report-only replies start without private evidence or cached private reasoning.

Intervention	Cyclic correct	Permutation correct
Direct answer	180/200	136/200
Additional local reasoning	184/200	144/200
Report-only reply	186/200	148/200
Fresh-evidence reply	190/200	176/200

The Permutation comparison gives 74% accuracy for report-only replies and 88% for fresh-evidence replies. The paired difference is 14 points with interval $[8.5, 19.5]$. This intervention changes evidence access within the second model. Generation costs are reported separately for the two model modes.

L.2 Reading identical facts in different formats

The reference-format control presents the same complete facts in a compact encoding and a delimited encoding on 200 paired Permutation worlds. Both representations fit the 64-token ceiling, while their actual communication differs. The fixed decoder recovers all 200 answers from either representation. Model accuracy therefore assesses the use of these complete reference records, separately from the construction of model-generated reports.

Table A32. Model correctness under two complete reference formats on 200 paired worlds. Communication entries give compact / delimited means. The fixed decoder solves all records in every condition.

Model	Compact	Delimited	Wire means
Qwen3-14B thinking	196/200	199/200	37 / 60
Qwen2.5-14B-Instruct	190/200	196/200	38 / 61

Delimited reporting changes Qwen3 correctness from 196 to 199 and Qwen2.5 correctness from 190 to 196. These are gains of 1.5 and three percentage points. The difference between gains is 1.5 points with interval $[-1.5, 4.5]$ in the paired comparison. The control holds facts fixed while allowing serialization and actual message cost to vary.

M SAMPLE ALLOCATION AND PAIRED COMPARISONS

M.1 SAMPLING UNITS AND INTERVAL CONSTRUCTION

The main routing comparison, fixed-record diagnostics, second-model comparison, and document task each use 200 base worlds per reported structure or task. Each world contributes one model draw per condition. The expanded routing grid uses 100 worlds per cell except the 200-world $K = 32, b = 64$ setting reused from the main comparison. Full-reference encoding profiles use 200 reference worlds per assignment size and remain separate from the 100-world large-assignment diagnostics. The earlier candidate-development comparison has eight independent worlds, each evaluated under paired structures and multiple configurations.

For paired procedures P and Q, let $Y_{i,P}$ and $Y_{i,Q}$ indicate whether world i is answered correctly. Their percentage-point difference is $100n^{-1}\sum_i(Y_{i,P} - Y_{i,Q})$. We compute pointwise 95% percentile-bootstrap intervals by resampling worlds and retaining all compared outcomes from each world together. Main-protocol and fixed-reply comparisons use 200,000 replicates. Fixed-decoder, model, document, and cross-condition comparisons use 100,000 replicates. Cost intervals use 30,000 replicates. Invalid answers remain zero-valued outcomes.

Table A33. Joint accuracy comparisons at $K = 32, b = 64$. Each structure uses 200 paired worlds. First-only and second-only counts identify worlds solved by exactly one of the compared procedures. Differences and intervals are in percentage points.

First − second	First only	Second only	Δ	95% interval
Cyclic				
Sequential − Report_0	4	2	+1.0	[−1.5, +3.5]
Follow-up − Report_0	2	2	0.0	[−2.0, +2.0]
Follow-up − Sequential	1	3	−1.0	[−3.0, +1.0]
Permutation				
Sequential − Report_0	14	4	+5.0	[+1.0, +9.0]
Follow-up − Report_0	14	6	+4.0	[−0.5, +8.5]
Follow-up − Sequential	2	4	−1.0	[−3.5, +1.5]

M.2 CONTRASTS ACROSS FORMATS AND DOCUMENT CONDITIONS

For the format control, define $Z_{i,m}$ as delimited-minus-compact correctness for model m. Model-dependent format sensitivity is assessed through $100\,\overline{Z_{i,\text{Qwen2.5}} - Z_{i,\text{Qwen3}}}$. For documents, let $G_{i,c} = Y_{i,\text{Sequential},c} - Y_{i,\text{Report}_0,c}$ denote the sequential gain under condition c. A change in this gain is $100\,\overline{G_{i,c_2} - G_{i,c_1}}$.

Table A34. Within-model format gains and cross-condition contrasts over 200 paired worlds per task. Intervals use 100,000 whole-world bootstrap replicates and are in percentage points.

Comparison	Difference (pp)	95% interval
Syntax gains from delimited reporting		
Qwen3 delimited − compact	+1.5	[−0.5, +4.0]
Qwen2.5 delimited − compact	+3.0	[+0.5, +6.0]
Qwen2.5 gain - Qwen3 gain	+1.5	[−1.5, +4.5]
Document gains from Sequential over reporting		
Versions - clean gain	+3.5	[−3.0, +10.0]
Dense - clean gain	+5.0	[−1.5, +12.0]
Dense - versions gain	+1.5	[−5.0, +8.5]

All contrast intervals include zero. The data do not establish differences in format sensitivity between models or changes in the sequential advantage across document conditions.