PROSPECTIVE INTERPRETATION RISK: PREDICTING AND PREVENTING MISCOMMUNICATION BETWEEN LLM AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language model (LLM) based agentic systems increasingly rely on models communicating with one another, yet existing uncertainty and multi-agent methods rarely estimate how a particular receiver will interpret a message before it is sent. This matters in heterogeneous systems, where capable receivers can reconstruct different tasks from the same message. We model this as a sender–receiver problem with a latent receiver type and define *prospective interpretation risk* (PIR): the probability that a receiver reconstructs a task other than intended. Rather than model an LLM's full input–output behaviour, we use black-box probes relating messages, intended tasks, and receiver-specific reconstructions, yielding scalable supervision while separating interpretation from downstream capability failure. Offline, heterogeneous frozen receivers provide supervision for receiver-conditioned risk and the effects of predefined mutable message features. At deployment, history induces a posterior over receiver types, guiding message revision and selection. We further introduce *value of interpretation information* (VoII), querying for receiver information only when its expected communication benefit exceeds its cost. Our theory characterises when receiver information has decision value and bounds such queries. Empirically, interpretation-failure rates vary by 4–13$\times$ across receivers. Receiver information reduces PIR calibration error by 68% relative to a receiver-agnostic predictor, largely by correcting receiver-specific risk levels. PIR-guided revision reduces interpretation failure by 44% relative to the original message and 40% relative to a generic rewrite, mostly through a repair that helps every receiver. VoII outperforms information-gain and random querying at matched cost on the interpretation objective it optimises, lowering interpretation failure from 3.84% to 3.79% while querying 18.2% of episodes.

1 INTRODUCTION

Large language model (LLM) applications are increasingly built as systems rather than single models. Planners delegate to executors, specialist agents exchange intermediate results, and tool-using components pass natural-language instructions to one another. Frameworks such as CAMEL (Li et al., 2023), AutoGen (Wu et al., 2024), MetaGPT (Hong et al., 2024), and ChatDev (Qian et al., 2024) make this pattern explicit. In these systems, communication is part of the execution path: a downstream model acts on what an upstream model meant to convey.

This creates a reliability problem that is easy to miss when models are evaluated in isolation. LLM systems are heterogeneous by nature: receivers can differ in architecture, pre-training, fine-tuning and post-training (Ouyang et al., 2022), system prompts, tools, memory, and update history (Chen et al., 2024). A natural-language handoff is therefore less stable than a typed interface. Different receivers can read the same message as different tasks (Sclar et al., 2024; Mizrahi et al., 2024), and the mismatch can spread even when each component is capable on its own (Cemri et al., 2025). In a healthcare triage pipeline, for example, one model may read "prioritise urgent cases" as symptom severity and another as time sensitivity (Figure 1, panel 1). Experiment 1 confirms that this happens in practice: on the same test messages, interpretation-failure rates vary widely across real receiver LLMs, and their ordering changes across datasets (Figure 2).

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.

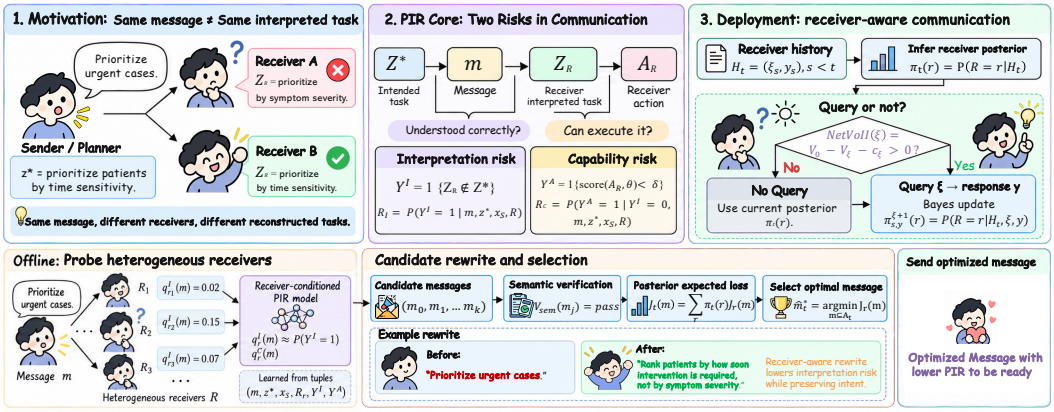


Figure 1: **Framework overview.** Top: receivers read one message as different tasks (1), PIR separates interpretation from capability risk (2), and the sender infers the receiver and queries only if worthwhile (3). Bottom: offline probes train the PIR model, which picks the verified rewrite to send.

Current approaches cover parts of this problem but not the quantity needed to control it. Uncertainty methods mainly estimate confidence in a model's own answer (Kadavath et al., 2022; Farquhar et al., 2024), ambiguity (Cole et al., 2023; Kim et al., 2024) and clarification (Zhang & Choi, 2025; Lin et al., 2026a; Dong et al., 2026; Suri et al., 2026) methods ask whether a message has multiple readings or needs more information, and multi-agent diagnostics study broader communication failures and how they spread (Cemri et al., 2025; Mguni et al., 2026; Lin et al., 2026b). Listener-aware methods show that generation can adapt to the listener (Fried et al., 2018; Zhu et al., 2021; Wang et al., 2021; Choi et al., 2025). What is largely missing is a calibrated estimate, available *before* sending, of whether *this receiver* will recover the intended task from *this message*, and a way to act on it.

An LLM's full text input–output space is too large to model in advance. We instead model a smaller, task-relevant object: the mapping from a message to the semantic task a receiver reconstructs. Each benchmark instance gives an intended task $z^\star$ and nearby alternative tasks. Independent black-box probes (Ribeiro et al., 2020) ask which task the receiver infers, and a separate rollout measures execution. This gives a scalable behavioural label without modelling free-form outputs or treating chain-of-thought (Wei et al., 2022) as ground truth (Turpin et al., 2023). It also lets us separate *interpretation failure*, which message revision can address, from downstream *capability failure*, which wording alone may not fix (Figure 1, panel 2). Experiment 1 confirms this separation. Experiment 2 shows that interpretation risk is predictable in advance, with strongly receiver-dependent calibration.

We call this quantity *prospective interpretation risk* (PIR). Offline, sending the same semantic handoffs to heterogeneous frozen LLM receivers gives privileged supervision (Vapnik & Vashist, 2009) on how interpretation and capability vary by receiver type (Figure 1, bottom left). We distil this response surface into receiver-conditioned heads and receiver-specific risk effects of predefined mutable message features (Hinton et al., 2015). At deployment, this population-wide supervision disappears: behavioural history gives a posterior over a closed set of receiver types (Sun et al., 2025; Pasquini et al., 2025), and risk is averaged over that posterior instead of conditioned on a known model identity. Experiment 3 shows that ordinary behavioural histories sharpen this posterior.

Beyond prediction, PIR changes how the sender communicates. The sender can rewrite a message (Zhou et al., 2023) to reduce posterior expected interpretation loss (Figure 1, bottom centre), or first gather receiver information and revise under the updated posterior. Querying, however, is itself costly: in larger agent networks, clarifying by default can cause *query flooding*, which has long limited the scalability of broadcast-style search in peer-to-peer systems (Lv et al., 2002). We therefore introduce *value of interpretation information* (VoII, Figure 1, panel 3) and query only when the expected improvement in the next message decision exceeds its cost (Howard, 1966; Rao & Daumé III, 2018).

This setting can be viewed as a Bayesian game (Harsanyi, 1967). The sender chooses a message under uncertainty about a latent receiver type, much as cheap-talk (Crawford & Sobel, 1982) and Bayesian persuasion (Kamenica & Gentzkow, 2011) models study communication under private information. Here, however, the main friction is receiver-dependent interpretation, not conflicting preferences. When receiver information is costly, the problem links to information theory and rational inattention

(Sims, 2003; Matějka & McKay, 2015): the sender should gather only the receiver information useful for its next message. Our two theoretical results make this precise. The receiver-conditioning theorem shows when heterogeneity can change the optimal message, and the information-to-communication frontier bounds the decision value of any query at a given receiver-information budget.

The framework therefore connects measurement, privileged receiver modelling, posterior inference, message revision, and selective querying. The experiments follow this logic. Experiment 1 shows that interpretation varies by receiver and differs from execution. Experiment 2 tests prospective prediction and the value of receiver conditioning. Experiment 3 tests how much ordinary history reveals about the latent receiver type. Experiment 4 tests whether posterior-guided revision reduces *measured* misinterpretation, and Experiment 5 tests whether VoII improves query allocation for the interpretation objective. Because interpretation failures are rare, any enrichment stays in training or analysis data and is tracked by provenance (King & Zeng, 2001; Saerens et al., 2002). Calibration (Guo et al., 2017) and headline evaluation use untouched natural-prevalence validation and test sets.

Section A surveys uncertainty, ambiguity, clarification, partner modelling, multi-agent communication, strategic communication, costly information, and privileged information.

2 Problem Setup and Measurement

Sender Intent, Interpretation, and Action

Let $\theta \sim p(\theta)$ denote an underlying task instance and x_S the task/environment information visible to the sender. The sender's intended semantic task is $z^\star = g(\theta, x_S) \in \mathcal{Z}$, where $\mathcal{Z}$ is the semantic task space and g maps the underlying task and sender-visible context to the semantic task the sender intends the receiver to perform. A sender S expresses this intent in a natural-language message m. Receiver R first reconstructs a task $Z_R \in \mathcal{Z}$ and then acts on it, producing action A_R:

$$z^\star \longrightarrow m \longrightarrow Z_R \longrightarrow A_R. \tag{1}$$

We model the intermediate task variable Z_R, not the receiver's full distribution over text outputs. This reduces a huge black-box input–output space to the semantic object that matters for communication reliability: which task the receiver believes it has been asked to do. Receiver type R is likewise defined by behaviour: it stands for a given frozen model and configuration and needs no access to the model's internals. Because several surface descriptions can express the same task, let $\mathcal{Z}^\star$ denote the semantic equivalence class of $z^\star$. We separate failure at the two stages of Equation (1):

$$Y^I = \mathbf{1}\{Z_R \notin \mathcal{Z}^\star\}, \quad Y^A = \mathbf{1}\{\text{score}(A_R, \theta) < \delta\}. \tag{2}$$

Thus $Y^I = 0, Y^A = 1$ means the receiver understood the task but its action scored below the pass threshold δ, whereas $Y^I = 1, Y^A = 0$ can occur when a wrong interpretation happens to pass this check. The distinction matters because only the former is a true capability failure.

Definition 2.1 (Prospective interpretation risk)**.** For a prospective receiver R, sender intent $z^\star$, context x_S, and candidate message m,

$$\mathcal{R}_I(m, z^\star, x_S, R) := \mathbb{P}(Y^I = 1 \mid m, z^\star, x_S, R). \tag{3}$$

To keep execution difficulty separate, we also define conditional capability risk

$$\mathcal{R}_C := \mathbb{P}(Y^A = 1 \mid Y^I = 0, m, z^\star, x_S, R). \tag{4}$$

Independent Interpretation Probe

Because Z_R is latent, we measure it through behaviour, using benchmark-defined task identities rather than self-reported reasoning. Each probe presents an independently worded description of $z^\star$, a small set of nearby but truly different tasks $z_1^-, \ldots, z_K^-$, and a NONE OF THESE/unclear option. The receiver picks the task that best matches the original message. This keeps supervision finite and black-box: it needs neither access to model internals nor a list of possible text outputs, and probe calls grow linearly with message–receiver pairs and can run in parallel across receivers. Probe wording is kept separate from the sender message, and option order is randomised (Zheng et al., 2024). Interpretation and execution are measured in independent rollouts from the same (m, x_S, R):

a task-identification probe gives $\widehat{Z}_R$, and task execution gives A_R. The execution branch never sees the probe alternatives. A specific wrong task and NONE OF THESE are kept as separate raw outcomes, but both count as $Y^I = 1$ for binary PIR. We repeat the probe $K_p = 6$ times per message–receiver pair, in separate calls with independently generated wording and option order. For message–receiver pair i, let $\mathcal{P}_i$ contain the indices of its parsed probe responses. For $|\mathcal{P}_i| > 0$, the soft training target is

$$\widetilde{y}_i^I = \frac{1}{|\mathcal{P}_i|} \sum_{k \in \mathcal{P}_i} \mathbf{1}\{\widehat{Z}_{R,i}^{(k)} \notin \mathcal{Z}_i^\star\}. \tag{5}$$

These repeats reduce measurement noise and are not treated as independent examples. An unparsed probe response is dropped from the average, and NONE OF THESE counts as a wrong task. Training uses the resulting soft target directly. Section D.2 describes the task bank and splits.

3 METHOD

The measurement protocol turns open-ended LLM behaviour into receiver-specific interpretation and execution outcomes. We use these compact labels in two stages: offline, to learn the decision-relevant response surface across receiver types, and at deployment, to infer the current receiver and choose a communication action without seeing its identity or predicting all of its future text.

Privileged Receiver Supervision.

Offline, each candidate message can be sent to several frozen receivers. Let r index receiver types and R_r denote the corresponding frozen receiver. For R_r we collect $(m, z^\star, x_S, R_r, \widetilde{y}_r^I, y_r^A)$, where $y_r^A \in \{0, 1\}$ is the observed action-failure label for that message–receiver pair. Hereon, subscripts r and posterior terms $\pi_t(r)$ refer to the receiver-type index. Sending one semantic handoff to several frozen LLMs gives privileged supervision on heterogeneity arising jointly from architecture, training, post-training, and configuration. Receiver identity indexes the behavioural type, so these sources need not be separated. The responses are distilled into two receiver-conditioned heads with parameters ϕ, $q_r^I(m) = q_\phi^I(m, z^\star, x_S, e_r)$ and $q_r^C(m) = q_\phi^C(m, z^\star, x_S, e_r)$, where e_r is a learned embedding of type r. These separate networks are trained with binary cross-entropy (BCE) losses $\mathcal{L}^I = \mathrm{BCE}(q^I, \widetilde{y}^I)$ and $\mathcal{L}^C = (1 - \widetilde{y}^I)\,\mathrm{BCE}(q^C, y^A)$, whose weight, the share of probes read correctly, softens the condition $Y^I = 0$ of Equation (4). Both are calibrated on natural-prevalence validation data. PIR is the estimand, not one fixed predictor: Experiment 2 tests the heads on natural messages, and Experiments 4–5 fit candidate-level estimators to measured rewrites whose distribution differs.

Closed-Set Receiver Inference.

Privileged supervision is available only offline. At deployment the true receiver is hidden but belongs to the same finite type set, $\mathcal{R} = \{R_1, \ldots, R_K\}$, $\mathcal{R}_{\text{train}} = \mathcal{R}_{\text{test}}$. The sender instead observes a history of calibration queries and responses, $H_t = \{(\xi_s, y_s)\}_{s<t}$, from which a small history encoder gives the calibrated posterior $\pi_t(r) = P(R = r \mid H_t)$. For one-shot evaluation, H_0 is built from genuine stored responses of the hidden receiver to separate calibration-bank items, and every receiver type uses the same sampling rule. The type model need only separate receiver profiles with different communication risks, not predict the receiver's future text. Rather than a maximum a posteriori (MAP) guess, we carry receiver uncertainty into the decision. For any message, $\bar{q}_t^I(m) = \sum_r \pi_t(r) q_r^I(m)$ and $\bar{q}_t^C(m) = \sum_r \pi_t(r) q_r^C(m)$, and define per-receiver loss

$$J_r(m) = c_M(m) + L_I q_r^I(m) + L_C(1 - q_r^I(m)) q_r^C(m), \tag{6}$$

where $L_I, L_C \geq 0$ weight interpretation and capability failure, with $L_C = 0$ when we isolate interpretation risk, and $c_M(m)$ the cost of sending message m (the subscript M denotes message cost, e.g. token or latency cost). The controller acts on the posterior expected loss

$$\bar{J}_t(m) = \sum_r \pi_t(r) J_r(m). \tag{7}$$

Posterior-Guided Message Revision.

Receiver awareness matters only if it changes what the sender does. The interpretation/capability split tells the sender what to change: better wording can often reduce interpretation risk, but paraphrasing

cannot fix a receiver's execution capability. Our first intervention therefore targets *mutable* message features linked to interpretation failure: the feature set is predefined, and what is learned is each receiver's risk effect s_{rk} of feature k. Let m_0 be the initial message. Task-inherent features may predict risk but are not valid rewrite targets. Under the current posterior we aggregate $\bar{s}_{tk} = \sum_r \pi_t(r) s_{rk}$ and give m_0, $z^\star$, x_S, and the highest-risk mutable features to the rewrite model, which produces a fixed set of $K_m = 4$ candidates. Reducing predicted PIR is useful only if the rewrite preserves the task, so every candidate m'_j first passes a semantic-equivalence filter V_{sem}. This filter is a separate, stateless, batched call to the rewriter's model version. The verifier sees the intended task/context, original message, and candidates, but not the receiver posterior, PIR scores, feature weights, or candidate preference. A candidate passes only if it preserves the operation, target/referent, scope, constraints, conditions, and required output. The valid action set is $\mathcal{A}_t = \{m_0\} \cup \{m'_j : V_{\text{sem}}(m'_j) = \textsc{Pass}\}$. We then choose $\widehat{m}_t^\star = \arg\min_{m \in \mathcal{A}_t} \bar{J}_t(m)$. The original message is always kept, and the fixed generation budget rules out a "rewrite until the score falls" loop.

Receiver Queries and VoII.

Revision adapts the message to the current receiver belief. The second intervention asks whether that belief is worth refining. Before sending, the sender may therefore ask one costly query ξ from a fixed calibration bank $\mathcal{Q}$, kept separate from the test tasks. Repeated offline responses give a smoothed empirical likelihood $\widehat{P}(Y_\xi = y \mid R = r, \xi)$, where Y_ξ denotes the observable response signature elicited by query ξ (binary execution success in Experiment 5). Under π_t, the predictive response distribution is $\widehat{P}(y \mid H_t, \xi) = \sum_r \pi_t(r) \widehat{P}(y \mid R = r, \xi)$, and observing a reply y gives the update

$$\pi_{t+1}^{\xi,y}(r) = \frac{\widehat{P}(y \mid R = r, \xi)\pi_t(r)}{\sum_{r'} \widehat{P}(y \mid R = r', \xi)\pi_t(r')}. \tag{8}$$

Expected entropy reduction, or information gain, $\text{IG}(\xi) = H(\pi_t) - \mathbb{E}_y H(\pi_{t+1}^{\xi,y})$ measures how informative a query is and can shortlist the bank. Being informative, however, does not imply decision value: identifying the receiver better does not help if the preferred message stays the same. We therefore compare the best available action before and after a hypothetical query. Holding $\mathcal{A}_t$ fixed, the no-query value is $V_0 = \min_{m \in \mathcal{A}_t} \sum_r \pi_t(r) J_r(m)$, and the expected post-query value is

$$V_\xi = \sum_y \widehat{P}(y \mid H_t, \xi) \left[\min_{m \in \mathcal{A}_t} \sum_r \pi_{t+1}^{\xi,y}(r) J_r(m) \right]. \tag{9}$$

The corresponding gross VoII is $\text{VoII}_t(\xi) = V_0 - V_\xi$. Let $c_\xi \geq 0$ denote the decision cost of issuing query ξ (e.g. token, latency, or compute cost). The net value is

$$\text{NetVoII}(\xi) = \text{VoII}_t(\xi) - c_\xi = V_0 - V_\xi - c_\xi. \tag{10}$$

The sender asks $\xi^\star = \arg\max_\xi \text{NetVoII}(\xi)$ only when this value is positive. After seeing the response, it updates π_t, re-scores the same valid action set, and sends the message with the lowest expected loss. Keeping the candidate set fixed in the hypothetical branches separates the value of receiver information from random differences in generation.

4 Decision-Theoretic Results

The controller above is a finite Bayesian sender–receiver decision problem: the receiver has a latent type, the sender holds a posterior over that type, and the sender chooses a message after possibly observing another signal. It raises two questions: when can receiver information change the best message, and how much can a query be worth? The results below answer these questions and connect the controller to an information-theoretic view of LLM communication. Proofs are in Section C.

Value of Receiver Conditioning.

Fix context $C = (z^\star, x_S)$. For candidate m_j with cost c_j and failure indicator F_j, let $p_j(e) = \mathbb{P}(F_j = 1 \mid E = e, C)$ and $J_j(e) = c_j + L_I p_j(e)$, where E is the sender's receiver information. A receiver-agnostic sender chooses before observing E, a receiver-aware one after. Their difference is

$$\Delta_E := \min_j \mathbb{E}[J_j(E)] - \mathbb{E}\left[\min_j J_j(E)\right]. \tag{11}$$

Theorem 4.1 (Receiver-conditioning value). *For J candidates and bounded PIR losses:*

$$\Delta_E = \mathbb{E}[J_{j_0}(E) - J_{j^\star(E)}(E)] \geq 0, \quad \Delta_E = 0 \iff J_{j_0}(E) = \min_j J_j(E) \text{ a.s.} \tag{12}$$

for some receiver-agnostic minimiser $j_0 := \arg\min_j \mathbb{E}[J_j(E)]$ and receiver-aware minimiser $j^\star(E) := \arg\min_j J_j(E)$. If $\mathbb{P}(J_{j_0}(E) - \min_j J_j(E) \geq \gamma) \geq \alpha$, then $\Delta_E \geq \alpha\gamma$. Moreover,

$$\Delta_E \leq L_I \sqrt{\sum_{j=1}^{J} \mathrm{Var}[p_j(E)]}, \quad \Delta_E \leq L_I \sqrt{\sum_{j=1}^{J} I(F_j; E \mid C)}. \tag{13}$$

The result separates heterogeneity from usefulness. PIR can vary widely across receivers while the same message stays optimal for all of them, and then receiver conditioning has no decision value. Positive value appears only when receiver information moves probability mass across message decision regions. The variance and mutual-information bounds show how large that value can be.

Information-to-Communication Frontier The first theorem assumes receiver information is already available. The second asks how much can be gained by acquiring it. Let the latent payoff-relevant receiver state be the discrete receiver type R and condition on $I_t = (H_t, C)$. Define the best loss attainable over message policies ρ when at most B nats of receiver information may influence the final message,

$$D_t(B) = \min_{\rho(m|r,I_t):\, I(R;M|I_t) \leq B} \mathbb{E}[L(M, R) \mid I_t], \tag{14}$$

where M takes values in $\mathcal{A}_t$, with $L(m, r) = J_r(m)$, and let $G_t(B) = D_t(0) - D_t(B)$.

Theorem 4.2 (Information-to-communication frontier). *For finite receiver and message spaces and bounded loss, $D_t(B)$ is non-increasing and convex in B, and $G_t(B)$ is non-decreasing and concave. For any query ξ with response Y_ξ, followed by any message-selection rule based on that response,*

$$\mathrm{VoII}_t(\xi) \leq G_t(I(R; Y_\xi \mid I_t)). \tag{15}$$

Conditioned on I_t, the Markov chain $R \to Y_\xi \to M$ gives the bound by data processing (Cover & Thomas, 2006). This explains why IG and VoII differ: a query can identify the receiver well and still have no value if the optimal communication action does not change. The concavity of G_t also captures the diminishing returns of more receiver information.

5 Experimental Design

Datasets. We use six knowledge-base question-answering (KBQA) collections: FreebaseQA (Jiang et al., 2019), WebQSP (Yih et al., 2016), GrailQA (Gu et al., 2021), SimpleQuestions-Wikidata (SQ-WD) (Bordes et al., 2015; Diefenbach et al., 2017), PopQA (Mallen et al., 2023), and Mintaka (Sen et al., 2022). The message m is the question rendered by the fixed PromptSource template `qa_template_basic` (Bach et al., 2022). The sender is fixed on purpose: every receiver sees the same message, which isolates receiver-dependent interpretation in Experiments 1–3. Experiments 4–5 then test LLM revision and query control. The intended task $z^\star$ is the benchmark task, and each message carries one contrast task from a fixed set (entity versus category for entity-valued questions, value versus attribute for class-valued questions). The test split has 6,629 messages and 33,145 message–receiver pairs (per-dataset sizes in Table 7, Section I.1, and splits in Section D.2).

Receivers. The receiver pool has five open-weight models from different families: Nemotron-3 Nano 30B (NVIDIA, 2025), Gemma 4 26B (Gemma Team, 2026), gpt-oss-20b (OpenAI, 2025), Mistral Small 3.2, and Qwen3.6 35B. All run frozen through vLLM (Kwon et al., 2023) at temperature 0, so each label is a deterministic function of message and receiver.

Baselines. Following Table 6 (Section H), A0 uses global and per-receiver base rates, A1 is a message-only ambiguity judge, A2 is a semantic-equivalence judge that sees $(m, z^\star, x_S)$ but not the receiver, A3 is our model without e_R, A4 is our model trained on Y^A and scored against Y^I, and A5 is a theory-of-mind (ToM) listener that predicts the receiver's probe choice and uses $1 - P(z^\star)$ as risk (Zhu et al., 2021). A shuffled-receiver control permutes receiver identities across training pairs.

Protocols. Each probe offers the intended task, the contrast task, and None of these. Execution is a separate answer call scored by gold-alias containment. Prediction is scored by areas under

the receiver operating characteristic and precision–recall curves (AUROC, AUPRC), negative log-likelihood (NLL), Brier score, and expected calibration error (ECE). Labels, the predictor, and full protocols for Experiments 3–5 are in Section G, with metrics, statistics, and scope in Sections G.5 and G.6.

6 RESULTS

Interpretation and Execution Failures Separate Empirically

Three facts support treating Y^I as its own target. First, the two failures do not coincide. Table 7 in Section I.1 gives the $Y^I \times Y^A$ decomposition. Both off-diagonal cells are non-zero on every dataset: 56.4% of pairs read the task correctly but still fail, and 1.4% read it wrongly yet pass the answer check. Pooled over datasets, 32% of all misread pairs still pass, so an answer-based score would call them successes. Second, the execution outcome carries almost no information about interpretation (Figure 2a): pooled over receivers, the execution-failure rates after a misread and after a correct read differ by at most five points on five of the six datasets (14 points on WebQSP, the smallest set), while the rate itself spans 24–90% across datasets.

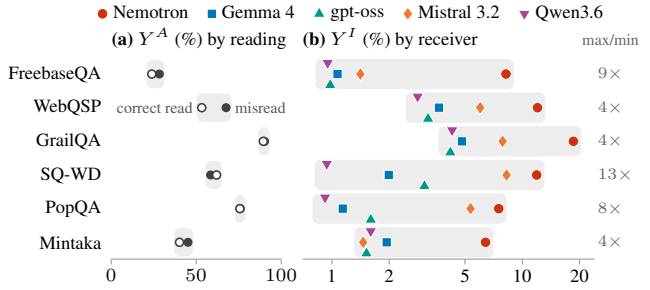


Figure 2: Interpretation failure is separate from execution failure and depends on the receiver (test split). (a) Y^A after a correct read (○) and a misread (●) of the probe. (b) Y^I per receiver. Bands span the lowest to the highest receiver.

Third, the receiver matters (Figure 2b). Pooled Y^I ranges from 1.7% (Qwen3.6) to 10.7% (Nemotron) on the same messages, and within each dataset the best and worst receiver differ by 4–13×. Nemotron is highest on every dataset, but the lowest receiver changes (Qwen3.6 on four datasets, gpt-oss on GrailQA, Mistral on Mintaka), so receiver risk is not a single number per receiver. Heterogeneity survives a control for option position: with the intended task listed first or second, the highest and lowest receivers still differ by 3.8× (95% confidence interval [CI] 3.4–4.3), and by 1.7–14× within datasets (Section D.3, which traces most of Nemotron's excess to the last listed option). Five temperature-0 repeats reproduce every dataset's Y^I within 0.07 points.

Receiver Conditioning Improves Prospective PIR Prediction

Table 1 compares the receiver-conditioned predictor with the baselines of Table 6. Five points follow. (i) Interpretation risk is predictable before sending: with pairs ranked within each receiver and then averaged (per receiver), AUROC is 0.732 and AUPRC 0.135 against a positive rate of 0.045. (ii) The signal is specific to interpretation. The action-failure predictor (A4), trained on Y^A with the same inputs, scores 0.511 on Y^I and is below chance on three datasets. The message-only judge (A1, 0.552) and the equivalence judge (A2, 0.500) stay near chance. Neither general difficulty nor message ambiguity explains the target.

Table 1: Prospective PIR prediction (test split, six-dataset means, seed 0). Ours is best on all seven metrics (bold, second best underlined, A0 excluded).

	Per receiver↑		Across receivers↑		Calibration↓		
Method	AUROC	AUPRC	AUROC	AUPRC	Brier	NLL	ECE
A0 Global prior	.500	.045	.500	.045	.0428	.180	.029
A0 Receiver prior	.500	.045	.697	.086	.0415	.168	.002
A1 Message judge	.552	.057	.540	.055	.0428	.180	.031
A2 Equiv. judge	.500	.045	.500	.045	.0428	.180	.029
A4 Y^A predictor	.511	.048	.530	.050	.4229	1.329	.540
A5 Listener/ToM	.716	.104	.782	.129	.0412	.160	.013
A3 No receiver	.729	.128	.701	.115	.0414	.168	.032
A3 + recv. calib.	.728	.128	<u>.796</u>	<u>.158</u>	<u>.0399</u>	<u>.155</u>	<u>.012</u>
Shuffled receiver	<u>.730</u>	<u>.130</u>	.700	.115	.0414	.168	.032
Ours	**.732**	**.135**	**.800**	**.164**	**.0398**	**.154**	**.010**

(iii) Receiver information mainly improves calibration. Removing e_R (A3) leaves within-receiver AUROC almost unchanged (0.729 vs. 0.732) but triples ECE (0.032 vs. 0.010) and worsens NLL and Brier on all six datasets, with paired intervals that exclude zero (Table 10 and Figure 4 in Section I.2). When all pairs are ranked together (across receivers), the same information lifts AUROC from 0.701 to 0.800 and AUPRC from 0.115 to 0.164, because the model must then rank one message across receivers, which is exactly the sender's decision. (iv) Most of the gain comes from learning each

receiver's risk level, including Nemotron's last-option effect (Section D.3): recalibrating A3 with the full model's per-receiver map (A3 + recv. calib.) recovers 93% of the ECE reduction and 96% of the across-receiver AUROC gain. The full model still ranks better within receivers (AUROC $+0.004$ and AUPRC $+0.007$, with paired intervals excluding zero), so message–receiver interaction adds its own ranking gain (Section I.2). (v) The gain needs the correct receiver: the shuffled-receiver control matches A3 within 0.002 on every averaged metric. The listener model (A5), the strongest non-ablation baseline, is worse than ours on all seven averaged metrics and on Brier in every dataset (Table 9).

Behavioural History Sharpens Receiver Beliefs

The receiver is never named at test time. The sender only sees stored responses to calibration tasks. Figure 5 in Section I.3 shows that these responses carry receiver information that grows with history length. Accuracy rises from the uniform prior (20.0%, five receivers) to 22.6% after one response, 24.9% after five, and 29.0% after twenty, while NLL falls from 1.609 to 1.584 and Brier from 0.800 to 0.790. The shuffled-history control stays at chance (18.4–21.3%) at every length, so the gain comes from the receiver's own behaviour, not from the tasks. The confusion matrix (Table 12 in Section I.3) shows where the evidence is: predictions concentrate on Nemotron and Mistral, the two receivers with the highest interpretation risk (Nemotron histories are recognised 71% of the time), whereas Gemma is almost never predicted. This is the pattern the framework needs, since the posterior only has to separate receivers whose risk differs. Because the posterior stays broad even at twenty responses, the controller keeps the full posterior π_t instead of committing to a MAP receiver. Experiment 4 compares it with a uniform prior and the true identity.

Risk-Guided Revision Reduces Measured Misinterpretation

Experiment 4 runs the full method of Section 3 on 1,579 held-out questions, with s_{rk} learned on 600 separate training questions and the Experiment 3 posterior after five stored responses (Section G.3). All rewrite conditions share the rewrite model, intended task, four candidates, and verifier. They differ only in where learned risk is used. Revision lowers Y^I from 4.16% to 2.31%, a 44% drop, and beats all four pre-registered references by 1.11 to 1.85 points ($p < 0.001$ each after Holm (1979) correction, Table 2). It does not hurt execution: Y^A is 1.00 point lower than for the original and within a 1-point non-inferiority margin for every reference. The drop holds for all five receivers and six datasets and is largest where the original risk is highest: 4.96 points for Nemotron, 2.62 for Mistral, and 0.40–0.69 for the other three (Section I.4).

Table 2: Message revision on 1,579 held-out questions: ours beats every pre-registered reference on Y^I (‡ Holm $p < 0.001$). Gen.: learned risk guides generation (π_t: posterior only). Sel.: it selects the sent message.

Condition	Gen.	Sel.	$Y^I \downarrow$ (%)	ΔY^I (ours−row) points [95% CI]	$Y^A \downarrow$ (%)
Original message	–	–	4.16	-1.85 $[-2.09, -1.62]^\ddagger$	59.33
Generic rewrite	–	–	3.85	-1.54 $[-1.78, -1.30]^\ddagger$	58.15
Selection only	–	✓	3.47	-1.15 $[-1.41, -0.93]^\ddagger$	58.20
Posterior only	π_t	✓	3.43	-1.11 $[-1.33, -0.90]^\ddagger$	58.59
Guidance only	✓	–	2.29	$+0.03$ $[+0.01, +0.05]$	58.33
Ours	✓	✓	**2.31**	–	58.33
True identity	✓	✓	2.30	$+0.02$ $[-0.08, +0.12]$	58.58
One feature per step	✓	✓	2.12	$+0.20$ $[+0.03, +0.37]$	58.75

The gain comes from guidance, not selection: guided candidates sent without selection are 1.56 points below the generic rewrite, while selection among generic candidates gains 0.38. Of the guided rewrites, 97% keep the question and add a line that states the output (e.g., "Give the name of the Italian fashion designer."), while 95% of generic rewrites leave it unstated. Most of the gain thus comes from one repair that helps every receiver, and the framework shows when receiver-specific adaptation is needed rather than assuming it always is. Here all five receivers share the top risk feature, an unstated output, so a uniform belief changes only 2 of 7,895 decisions and the true identity gives almost the same Y^I (2.30% vs. 2.31%): the case $\Delta_E \approx 0$ of Theorem 4.1, where receiver information barely changes the best message. One feature per step lowers Y^I further, to 2.12%, at the cost of four dependent rewrite rounds and about twice the tokens.

VoII Allocates Receiver Queries by Decision Value

A query is worth asking only if its answer can change the message. Experiment 5 tests this rule. Table 3 reports the primary cohort (6,629 test questions, five receivers, three fresh repeats, 99,435 episodes) and a replication on a second cohort of new questions. Each episode starts from five stored responses and the Experiment 3 posterior, and every policy asks at most one query from the same bank and sends the message chosen by a risk head (Section G.4). Without a query, measured

interpretation failure is 3.84%. With the true receiver identity it would be 3.77%, a gap of 0.06 points. At a 20% budget, NetVoII reaches 3.79% and closes 78% of this gap, while IG and random querying reach 3.83% and 3.84% and close 5% and 3%. The paired differences are consistent: NetVoII lowers Y^I by 0.03 to 0.06 points against both controls at every budget from 10% to 50% (Figure 8b in Section I.5), and all eight pre-registered contrasts in net utility $1 - Y^I - c\,r$ (query rate r, query cost c) survive Holm correction ($p \leq 0.012$). The strict rule queries only where $\mathrm{NetVoII} > 0$ at $c = 0.001$. It queries 18.2% of episodes and ends at 3.79%, -0.046 points below no query (95% CI $[-0.068, -0.025]$), which is 72% of the way to the true-identity reference. At the same query count, IG and random querying stay at 3.83% and 3.84%, so the strict rule improves net utility by $+0.0004$ over the stronger of the two. The adapted value-of-information (VoI) rule (Dong et al., 2026), a secondary comparator, replaces the risk heads by LLM utility estimates. These are so coarse that one candidate is best, or tied for best, for every receiver in every question, so no query can change the message. Ties, broken toward the original as for every policy, keep the original in 81% of episodes (18% with the heads), and Y^I stays at 4.42% (Section I.5).

Figure 8 shows why decision value, not information, is the right criterion. IG identifies the receiver at least as well as NetVoII: at 20% the posterior mode is the true receiver in 28.5% of episodes for IG and 28.3% for NetVoII, and in 44.0% when every episode is queried (panel c). Yet querying every episode lowers Y^I by only 0.04 points, to 3.80%, and on the replication cohort it raises Y^I (3.96% vs. 3.93%). Entropy falls, but the preferred message rarely changes. NetVoII spends its queries where a changed posterior would change the message. It queries about half of the PopQA and Mintaka episodes and none of WebQSP or GrailQA, and nearly all of its gain comes from those two datasets (Table 17 in Section I.5).

Table 3: Query policies (interpretation objective, $c = 0.001$). Gap: share of the no-query to true-identity Y^I gap that is closed. Ident.: posterior mode is correct. IG: information gain. † Secondary comparator. Bold: best deployable value.

	Primary cohort				Replication	
Policy (query rate)	Y^I (%)	Gap (%)	Net utility	Ident. (%)	Y^I (%)	Gap (%)
No query (0%)	3.84	0	.9616	24.6	3.93	0
Exact 20% quota						
Random (3 seeds)	3.84	3	.9614	25.7	3.93	4
IG	3.83	5	.9615	28.5	3.94	<0
NetVoII (ours)	**3.79**	**78**	**.9619**	28.3	**3.91**	**76**
No quota						
NetVoII strict (18%)	**3.79**	72	**.9619**	28.0	3.92	47
Always, IG (100%)	3.80	55	.9610	**44.0**	3.96	<0
Adapted VoI† (0%)	4.42	<0	.9558	24.6	4.27	<0
True-identity reference	3.77	100	.9623	100.0	3.91	100

The size of this gain is set by how much decision-relevant receiver information is left after five stored responses (Section I.5, which also reports task success). On a base of 3.84%, the true identity would lower Y^I by 0.06 points, and NetVoII lowers it by about 0.05. On the replication cohort, NetVoII at 20% again closes 76% of the gap (Table 3), a directional replication.

7 Conclusion

We introduced PIR, the probability, before sending, that a receiver reconstructs a task other than intended. Interpretation failure is receiver-specific, distinct from execution failure, and predictable before sending. Our theory shows that receiver information pays off exactly when it changes the preferred message. Empirically, receiver information reduces PIR calibration error by 68% relative to a receiver-agnostic predictor. PIR-guided revision lowers measured interpretation failure by 44% relative to the original message and 40% relative to a generic rewrite. For interpretation, VoII allocates queries better than IG and random querying at matched cost, closing 78% of the 0.06-point gap to the true-identity reference at a 20% budget. More broadly, PIR suggests that multi-agent LLM systems should check messages for likely misreading before sending, rather than after failure. Beyond these results, our framework could become part of the communication infrastructure for future multi-agent systems, supporting reliable massive agentic interaction by making interpretation failure something agents can anticipate and mitigate before sending.

AI USE STATEMENT

We used generative artificial intelligence (AI) tools for language editing, drafting support, literature search, and to support some implementation methods. The authors reviewed all AI-assisted text, technical claims, derivations, experimental specifications, and code, and take full responsibility for the final content.

REPRODUCIBILITY STATEMENT

The paper specifies the PIR labels, receiver-inference model, query-response model, VoII controller, rewrite procedure, baselines, and ablations. Full proofs are in Section C, and probe construction, receiver inference, message revision, experimental protocols, baselines, pseudocode, and full experimental details are in Sections D to H, J and K.

REFERENCES

Stephen H. Bach, Victor Sanh, Zheng-Xin Yong, Albert Webson, Colin Raffel, Nihal V. Nayak, Abheesht Sharma, Taewoon Kim, M. Saiful Bari, Thibault Fevry, et al. PromptSource: An integrated development environment and repository for natural language prompts. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pp. 93–104. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.acl-demo.9.

Antoine Bordes, Nicolas Usunier, Sumit Chopra, and Jason Weston. Large-scale simple question answering with memory networks. *arXiv preprint arXiv:1506.02075*, 2015.

Glenn W. Brier. Verification of forecasts expressed in terms of probability. *Monthly Weather Review*, 78(1):1–3, 1950. doi: 10.1175/1520-0493(1950)078⟨0001:VOFEIT⟩2.0.CO;2.

Mert Cemri, Melissa Z. Pan, Shuyi Yang, Lakshya A. Agrawal, Bhavya Chopra, Rishabh Tiwari, Kurt Keutzer, Aditya Parameswaran, Dan Klein, Kannan Ramchandran, Matei Zaharia, Joseph E. Gonzalez, and Ion Stoica. Why do multi-agent LLM systems fail? In *Advances in Neural Information Processing Systems*, volume 38, pp. 135676–135729, 2025. doi: 10.52202/085713-4082.

Lingjiao Chen, Matei Zaharia, and James Zou. How is ChatGPT's behavior changing over time? *Harvard Data Science Review*, 6(2), 2024. doi: 10.1162/99608f92.5317da47.

Younwoo Choi, Changling Li, Yongjin Yang, and Zhijing Jin. Agent-to-agent theory of mind: Testing interlocutor awareness among large language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 28895–28928. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.emnlp-main.1471.

Jeremy Cole, Michael Zhang, Daniel Gillick, Julian Eisenschlos, Bhuwan Dhingra, and Jacob Eisenstein. Selectively answering ambiguous questions. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 530–543. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.emnlp-main.35.

Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. Wiley, 2 edition, 2006.

Vincent P. Crawford and Joel Sobel. Strategic information transmission. *Econometrica*, 50(6): 1431–1451, 1982. doi: 10.2307/1913390.

DeepSeek-AI. DeepSeek-V4: Towards highly efficient million-token context intelligence. *arXiv preprint arXiv:2606.19348*, 2026.

Dennis Diefenbach, Thomas Pellissier Tanon, Kamal Deep Singh, and Pierre Maret. Question answering benchmarks for Wikidata. In *Proceedings of the ISWC 2017 Posters & Demonstrations and Industry Tracks*, volume 1963 of *CEUR Workshop Proceedings*, 2017.

Yijiang River Dong, Tiancheng Hu, Zheng Hui, Caiqi Zhang, Ivan Vulić, Andreea Bobu, and Nigel Collier. Value of information: A framework for human–agent communication. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 42879–42896. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.acl-long.1987.

Bradley Efron and Robert J. Tibshirani. *An Introduction to the Bootstrap*. Chapman and Hall/CRC, 1994. doi: 10.1201/9780429246593.

Sebastian Farquhar, Jannik Kossen, Lorenz Kuhn, and Yarin Gal. Detecting hallucinations in large language models using semantic entropy. *Nature*, 630:625–630, 2024. doi: 10.1038/s41586-024-07421-0.

Daniel Fried, Ronghang Hu, Volkan Cirik, Anna Rohrbach, Jacob Andreas, Louis-Philippe Morency, Taylor Berg-Kirkpatrick, Kate Saenko, Dan Klein, and Trevor Darrell. Speaker-follower models for vision-and-language navigation. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

Gemma Team. Gemma 4 technical report. *arXiv preprint arXiv:2607.02770*, 2026.

Yu Gu, Sue Kase, Michelle Vanni, Brian Sadler, Percy Liang, Xifeng Yan, and Yu Su. Beyond I.I.D.: Three levels of generalization for question answering on knowledge bases. In *Proceedings of the Web Conference 2021 (WWW)*, pp. 3477–3488, 2021. doi: 10.1145/3442381.3449992.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1321–1330. PMLR, 2017.

John C. Harsanyi. Games with incomplete information played by "Bayesian" players, I–III. Part I. The basic model. *Management Science*, 14(3):159–182, 1967. doi: 10.1287/mnsc.14.3.159.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Sture Holm. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6(2):65–70, 1979.

Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, Chenyu Ran, Lingfeng Xiao, Chenglin Wu, and Jürgen Schmidhuber. MetaGPT: Meta programming for a multi-agent collaborative framework. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=VtmBAGCN7o.

Ronald A. Howard. Information value theory. *IEEE Transactions on Systems Science and Cybernetics*, 2(1):22–26, 1966.

Zhibo Hu, Chen Wang, Yanfeng Shu, Hye-young Paik, and Liming Zhu. Concept based ambiguity resolution in LLMs. In *Conference on Parsimony and Learning*, volume 328 of *Proceedings of Machine Learning Research*, pp. 940–956. PMLR, 2026.

Kelvin Jiang, Dekun Wu, and Hui Jiang. FreebaseQA: A new factoid QA data set matching trivia-style question-answer pairs with Freebase. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pp. 318–323, 2019. doi: 10.18653/v1/N19-1028.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, et al. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*, 2022.

Emir Kamenica and Matthew Gentzkow. Bayesian persuasion. *American Economic Review*, 101(6): 2590–2615, 2011.

Hyuhng Joon Kim, Youna Kim, Cheonbok Park, Junyeob Kim, Choonghyun Park, Kang Min Yoo, Sang-goo Lee, and Taeuk Kim. Aligning language models to explicitly handle ambiguity. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 1989–2007. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.emnlp-main.119.

Gary King and Langche Zeng. Logistic regression in rare events data. *Political Analysis*, 9(2): 137–163, 2001.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with PagedAttention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, pp. 611–626, 2023. doi: 10.1145/3600006.3613165.

Guohao Li, Hasan Abed Al Kader Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. CAMEL: Communicative agents for "mind" exploration of large language model society. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Bohan Lin, Kuo Yang, Zelin Tan, Yingchuan Lai, Chen Zhang, Guibin Zhang, Xinlei Yu, Miao Yu, Xu Wang, Yudong Zhang, and Yang Wang. AgentAsk: Multi-agent systems need to ask. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 28055–28077. Association for Computational Linguistics, 2026a. doi: 10.18653/v1/2026.acl-long.1294.

Shi Lin, Chenpei Wang, Peng Qian, Dezhang Kong, Minghao Li, Yufeng Li, and Xun Wang. Before agents speak: Pre-hoc failure risk inference in multi-agent systems. *arXiv preprint arXiv:2607.26836*, 2026b.

Qin Lv, Pei Cao, Edith Cohen, Kai Li, and Scott Shenker. Search and replication in unstructured peer-to-peer networks. In *Proceedings of the 16th International Conference on Supercomputing*, pp. 84–95. ACM, 2002. doi: 10.1145/514191.514206.

Bartosz Maćkowiak, Filip Matějka, and Mirko Wiederholt. Rational inattention: A review. *Journal of Economic Literature*, 61(1):226–273, 2023. doi: 10.1257/jel.20211524.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9802–9822, 2023. doi: 10.18653/v1/2023.acl-long.546.

Filip Matějka and Alisdair McKay. Rational inattention to discrete choices: A new foundation for the multinomial logit model. *American Economic Review*, 105(1):272–298, 2015. doi: 10.1257/aer.20130047.

David Henry Mguni, Yaqi Sun, Haojun Chen, Wanrong Yang, Amir Darabi, Larry Olanrewaju Orimoloye, and Yaodong Yang. Learning robust multi-agent policies via selective adversarial fault induction. In *Forty-third International Conference on Machine Learning*, 2026.

Sewon Min, Julian Michael, Hannaneh Hajishirzi, and Luke Zettlemoyer. AmbigQA: Answering ambiguous open-domain questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 5783–5797. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.emnlp-main.466.

Moran Mizrahi, Guy Kaplan, Dan Malkin, Rotem Dror, Dafna Shahaf, and Gabriel Stanovsky. State of what art? A call for multi-prompt LLM evaluation. *Transactions of the Association for Computational Linguistics*, 12:933–949, 2024. doi: 10.1162/tacl_a_00681.

Mahdi Pakdaman Naeini, Gregory F. Cooper, and Milos Hauskrecht. Obtaining well calibrated probabilities using Bayesian binning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29, pp. 2901–2907, 2015. doi: 10.1609/aaai.v29i1.9602.

NVIDIA. Nemotron 3 nano: Open, efficient mixture-of-experts hybrid Mamba-transformer model for agentic reasoning. *arXiv preprint arXiv:2512.20848*, 2025.

OpenAI. gpt-oss-120b & gpt-oss-20b model card. *arXiv preprint arXiv:2508.10925*, 2025.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pp. 27730–27744, 2022.

Dario Pasquini, Evgenios M. Kornaropoulos, and Giuseppe Ateniese. LLMmap: Fingerprinting for large language models. In *34th USENIX Security Symposium (USENIX Security 25)*, pp. 299–318. USENIX Association, 2025.

John C. Platt. Probabilities for SV machines. In Alexander J. Smola, Peter Bartlett, Bernhard Schölkopf, and Dale Schuurmans (eds.), *Advances in Large-Margin Classifiers*, pp. 61–74. MIT Press, 2000. doi: 10.7551/mitpress/1113.003.0008.

Chen Qian, Wei Liu, Hongzhang Liu, Nuo Chen, Yufan Dang, Jiahao Li, Cheng Yang, Weize Chen, Yusheng Su, Xin Cong, Juyuan Xu, Dahai Li, Zhiyuan Liu, and Maosong Sun. ChatDev:

Communicative agents for software development. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15174–15186. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.810.

Sudha Rao and Hal Daumé III. Learning to ask good questions: Ranking clarification questions using neural expected value of perfect information. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2737–2746. Association for Computational Linguistics, 2018.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. Beyond accuracy: Behavioral testing of NLP models with CheckList. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4902–4912. Association for Computational Linguistics, 2020.

Marco Saerens, Patrice Latinne, and Christine Decaestecker. Adjusting the outputs of a classifier to new a priori probabilities: A simple procedure. *Neural Computation*, 14(1):21–41, 2002.

Irina Saparina and Mirella Lapata. Disambiguate first, parse later: Generating interpretations for ambiguity resolution in semantic parsing. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 16825–16839. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.findings-acl.863.

Melanie Sclar, Yejin Choi, Yulia Tsvetkov, and Alane Suhr. Quantifying language models' sensitivity to spurious features in prompt design or: How I learned to start worrying about prompt formatting. In *The Twelfth International Conference on Learning Representations*, 2024.

Priyanka Sen, Alham Fikri Aji, and Amir Saffari. Mintaka: A complex, natural, and multilingual dataset for end-to-end question answering. In *Proceedings of the 29th International Conference on Computational Linguistics (COLING)*, pp. 1604–1619, 2022.

Christopher A. Sims. Implications of rational inattention. *Journal of Monetary Economics*, 50(3): 665–690, 2003. doi: 10.1016/S0304-3932(03)00029-1.

Christopher A. Sims. Rational Inattention and Monetary Economics. In *Handbook of Monetary Economics*, volume 3, pp. 155–181. Elsevier, 2010. ISBN 9780444534705. doi: 10.1016/B978-0-444-53238-1.00004-1. URL https://linkinghub.elsevier.com/retrieve/pii/B9780444532381000041.

Mingjie Sun, Yida Yin, Zhiqiu Xu, J. Zico Kolter, and Zhuang Liu. Idiosyncrasies in large language models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 57854–57885. PMLR, 2025.

Manan Suri, Puneet Mathur, Nedim Lipka, Franck Dernoncourt, Ryan A. Rossi, and Dinesh Manocha. Structured uncertainty guided clarification for LLM agents. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 40811–40838. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.findings-acl.2028.

Naftali Tishby, Fernando C. Pereira, and William Bialek. The information bottleneck method, 2000. Presented at the 37th Annual Allerton Conference on Communication, Control, and Computing, 1999, pp. 368–377.

Miles Turpin, Julian Michael, Ethan Perez, and Samuel R. Bowman. Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting. In *Advances in Neural Information Processing Systems*, volume 36, pp. 74952–74965, 2023.

Vladimir Vapnik and Akshay Vashist. A new learning paradigm: Learning using privileged information. *Neural Networks*, 22(5–6):544–557, 2009. doi: 10.1016/j.neunet.2009.06.042.

Rose Wang, Julia White, Jesse Mu, and Noah Goodman. Calibrate your listeners! robust communication-based training for pragmatic speakers. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pp. 977–984. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.findings-emnlp.83.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pp. 24824–24837, 2022.

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

Chenyang Yang, Yike Shi, Qianou Ma, Michael Xieyang Liu, Christian Kästner, and Tongshuang Wu. What prompts don't say: Understanding and managing underspecification in LLM prompts. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 9072–9101. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.findings-acl.441.

Wen-tau Yih, Matthew Richardson, Chris Meek, Ming-Wei Chang, and Jina Suh. The value of semantic parse labeling for knowledge base question answering. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 201–206, 2016. doi: 10.18653/v1/P16-2033.

Michael J. Q. Zhang and Eunsol Choi. Clarify when necessary: Resolving ambiguity through interaction with LMs. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 5541–5558. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.findings-naacl.306.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. Qwen3 Embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*, 2025.

Chujie Zheng, Hao Zhou, Fandong Meng, Jie Zhou, and Minlie Huang. Large language models are not robust multiple choice selectors. In *The Twelfth International Conference on Learning Representations*, 2024.

Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. Large language models are human-level prompt engineers. In *The Eleventh International Conference on Learning Representations*, 2023.

Hao Zhu, Graham Neubig, and Yonatan Bisk. Few-shot language coordination by modeling theory of mind. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 12901–12911. PMLR, 2021.

APPENDIX

A Related Work **16**

B Notation **16**

C Proofs and Additional Formal Results **19**
C.1 Receiver-Conditioning Value 19
C.2 Information-to-Communication Frontier 21

D Interpretation-Probe Construction **24**
D.1 Task Identities and Surface Realisations 24
D.2 Task Bank and Splits 24
D.3 Probe Robustness Checks 24

E Receiver-Type Inference and Query Bank **25**
E.1 Calibration Bank 25
E.2 History Encoder 25

F Message-Revision Implementation **26**

G Experimental Protocols **27**
G.1 Labels and Predictor 27
G.2 Receiver Inference (Experiment 3) 28
G.3 Message Revision (Experiment 4) 28
G.4 Query Control (Experiment 5) 28
G.5 Metrics and Statistics 29
G.6 Scope of the Reported Experiments 29

H Baselines and Ablations **29**
H.1 Essential Prediction Baselines 29
H.2 Essential Ablations 29

I Additional Results **30**
I.1 Experiment 1: Interpretation and Execution Failures 30
I.2 Experiment 2: Prospective PIR Prediction 31
I.3 Experiment 3: Receiver-Type Inference 35
I.4 Experiment 4: Message Revision 36
I.5 Experiment 5: Query Control 38

J Pseudocode and Full Experimental Specification **42**
J.1 End-to-End Deployment Pseudocode 42
J.2 Training Pseudocode 42
J.3 Instantiation in Experiments 4 and 5 44
J.4 Full Experimental Details for Reproducibility 44

K Implementation and Reproducibility Checklist **45**

L Discussion and Limitations **46**
L.1 What the Results Show 46
L.2 Limitations 47
L.3 Future Directions 47

A Related Work

Uncertainty, ambiguity, and clarification. Most language-model uncertainty methods estimate confidence in a model's own output through verbalised confidence, sampling consistency, or semantic uncertainty (Kadavath et al., 2022; Farquhar et al., 2024). Ambiguity work instead studies inputs with multiple plausible readings or trains models to request missing information (Min et al., 2020; Saparina & Lapata, 2025; Cole et al., 2023; Kim et al., 2024; Hu et al., 2026; Yang et al., 2026; Zhang & Choi, 2025). These are natural points of comparison, but neither asks whether a given receiver will reconstruct the sender's intended task. PIR can therefore be low for a linguistically ambiguous message if a receiver resolves it consistently, or high for a message that looks clear but that this receiver tends to misread.

Partner modelling and multi-agent communication. Pragmatic speakers and listener-aware generation model how a listener may respond (Fried et al., 2018; Zhu et al., 2021; Wang et al., 2021), and recent agent-to-agent work studies interlocutor awareness between LLMs (Choi et al., 2025). Other work classifies multi-agent communication failures (Cemri et al., 2025) or predicts how hallucinations spread before interaction (Lin et al., 2026b). Our target is more specific: a calibrated, receiver-specific probability of task misinterpretation measured independently from execution and used directly to choose a communication action.

Strategic communication and costly information. Our sender–receiver formulation is related to Bayesian communication and cheap-talk games (Crawford & Sobel, 1982), but we do not require conflicting preferences: uncertainty enters through the receiver's latent behavioural type and type-dependent interpretation. The query decision is closer to value-of-information (Dong et al., 2026) and rational-inattention (Sims, 2003; Matějka & McKay, 2015) problems, where information is acquired only when its decision value justifies its cost. This distinction matters at scale. Always asking for clarification is not a neutral baseline in a networked system: query flooding is a classic cause of traffic and scalability problems in distributed peer-to-peer search (Lv et al., 2002). VoII makes the same trade-off explicit for LLM communication.

Privileged information. The training setup is closest in spirit to learning using privileged information (Vapnik & Vashist, 2009) and distillation (Hinton et al., 2015). Offline, we observe a receiver-indexed response surface: how the same handoff is interpreted and executed across heterogeneous real LLMs. Deployment sees none of these population-wide outcomes, only ordinary behavioural history for the current receiver. The goal is therefore not to model each LLM's full text behaviour, but to distil the lower-dimensional, decision-relevant variation that matters for communication control before sending.

B Notation

Table 4 lists every symbol defined in the paper. Subscripts r and t index a receiver type and a decision step, superscripts I and C mark interpretation and capability (execution), and a bar marks an average over the receiver, as in $\bar{J}_t(m) = \sum_r \pi_t(r) J_r(m)$. A few letters also have a second, local meaning, among them K, g, G_t, R, r, B, and α. The table lists each use.

Table 4: Notation, grouped by the part of the framework that introduces each symbol. The last column gives where the symbol is defined, and – marks standard notation.

Symbol	Meaning	Defined in
General		
$\mathbb{E}$, $\mathbb{P}$ (or P), Var	Expectation, probability, and variance	–
$\mathbf{1}\{\cdot\}$	Indicator: 1 if the condition holds and 0 otherwise	–
$H(\cdot)$, $I(\cdot\,;\cdot \mid \cdot)$	Shannon entropy and conditional mutual information, in nats	–
$D_{\mathrm{KL}}(\cdot\|\cdot)$	Kullback–Leibler divergence	–
$\mathrm{Ber}(p)$	Bernoulli distribution with mean p	–
$\mathrm{BCE}(q, y)$	Binary cross-entropy, $-y \log q - (1-y)\log(1-q)$	–
Tasks, messages, and outcomes		
$\theta \sim p(\theta)$	Task and its distribution	Sec. 2

Table 4 (continued)

Symbol	Meaning	Defined in
S, x_S	Sender and the task/environment information visible to it (in the experiments, the original question, used as a proxy)	Sec. 2, App. G.1
$z^\star$, $\mathcal{Z}$	Intended semantic task of the sender and the task space, with $z^\star \in \mathcal{Z}$	Sec. 2
g	Intent map from the task and the sender's information to the intended task, $z^\star = g(\theta, x_S)$	Sec. 2
$\mathcal{Z}^\star$	Semantic equivalence class of $z^\star$: all tasks with the same meaning	Sec. 2
m	Natural-language message from the sender	Sec. 2
R	Receiver type: a frozen model and configuration, defined by its behaviour and latent at deployment	Sec. 2
r, R_r	A value of R, which indexes receivers in sums and subscripts, and the receiver of type r	Sec. 3
Z_R	Task that the receiver reconstructs from m (latent)	Eq. (1)
A_R (A_r)	Action or output of the receiver	Eq. (1)
$\mathrm{score}(A_R, \theta)$, δ	Task score of the output and its success threshold	Eq. (2)
Y^I	Interpretation failure, $\mathbf{1}\{Z_R \notin \mathcal{Z}^\star\}$	Eq. (2)
Y^A	Execution failure, $\mathbf{1}\{\mathrm{score}(A_R, \theta) < \delta\}$, so $1 - Y^A$ is task success	Eq. (2)
$\mathcal{R}_I(m, z^\star, x_S, R)$	Prospective interpretation risk (PIR), $\mathbb{P}(Y^I = 1 \mid m, z^\star, x_S, R)$	Eq. (3)
$\mathcal{R}_C$	Conditional capability risk, $\mathbb{P}(Y^A = 1 \mid Y^I = 0, m, z^\star, x_S, R)$	Eq. (4)
Interpretation probe		
$z_1^-, \ldots, z_K^-$	Nearby but different contrast tasks offered as probe options, where K counts them (one in the experiments)	Sec. 2
$\widehat{Z}_R$	Task that the receiver picks in the probe, a measurement of Z_R	Sec. 2
K_p	Probe repeats per message–receiver pair ($K_p = 6$)	Sec. 2
i, $\widehat{Z}_{R,i}^{(k)}$, $\mathcal{Z}_i^\star$	Index of a message–receiver pair, the task picked in its k-th probe, and the equivalence class of its intended task	Eq. (5)
$\widetilde{y}_i^I$ ($\widetilde{y}_r^I$, $\widetilde{y}^I$)	Soft interpretation label: share of parsed probes that pick a wrong option	Eq. (5)
y_r^A	Measured execution-failure label (Y^A) for receiver r	Sec. 3
$\mathcal{E}(z)$	Set $\{T_z^{(1)}, \ldots, T_z^{(R)}\}$ of validated descriptions of task z, where R counts descriptions	App. D
$\mathcal{E}_{\mathrm{send}}$, $\mathcal{E}_{\mathrm{probe}}$, $\widetilde{T}_z$	Disjoint banks of descriptions for messages and for probes, and a probe description of z	App. D
$s(T)$	Task signature $(\mathcal{I}_T, \mathcal{C}_T, \mathrm{op}_T, \mathcal{O}_T, \mathcal{F}_T)$ of description T: visible input fields, extra context, semantic operation, output meaning, and output format or type	App. D
Risk predictor		
q_ϕ^I, q_ϕ^C, ϕ	Interpretation and capability heads, two separate multilayer perceptrons (MLPs, q^I and q^C for short), and their parameters	Sec. 3
$q_r^I(m)$, $q_r^C(m)$	Head outputs for receiver r, $q_\phi^I(m, z^\star, x_S, e_r)$ and $q_\phi^C(m, z^\star, x_S, e_r)$: predicted PIR and capability risk	Sec. 3
e_r (e_R)	Learned 16-dimensional embedding of receiver type r	Sec. 3, App. G.1
$\mathcal{L}^I$, $\mathcal{L}^C$	Losses of the heads, $\mathrm{BCE}(q^I, \widetilde{y}^I)$ and $(1 - \widetilde{y}^I)\,\mathrm{BCE}(q^C, y^A)$, with $\mathcal{L}_r^I(m)$ and $\mathcal{L}_r^C(m)$ for one message and receiver	Sec. 3, Alg. 3
g	Scalar gap: root-mean-square difference of the standardised embeddings of x_S and m (not the intent map)	App. G.1
Receiver inference		
$\mathcal{R} = \{R_1, \ldots, R_K\}$	Closed set of K receiver types (five in the experiments), with $\mathcal{R}_{\mathrm{train}} = \mathcal{R}_{\mathrm{test}}$	Sec. 3
t	Decision step	Sec. 3
H_t, $\lvert H_t \rvert$	Behavioural history $\{(\xi_s, y_s)\}_{s<t}$ of calibration queries and responses, and its length	Sec. 3
H_0	Starting history for one-shot evaluation, built from stored responses of the hidden receiver	Sec. 3
$\pi_t(r)$	Calibrated posterior over receiver types, $P(R = r \mid H_t)$	Sec. 3
$\bar{q}_t^I(m)$, $\bar{q}_t^C(m)$	Posterior-averaged risks, $\sum_r \pi_t(r) q_r^I(m)$ and $\sum_r \pi_t(r) q_r^C(m)$	Sec. 3

Table 4 (continued)

Symbol	Meaning	Defined in
$\pi_\omega(r \mid H)$, ω	Receiver-type model, $\mathrm{softmax}_r(Wh_H + b)$ with output weights W and bias b, and all its parameters	App. E
h_s, f_{int}	Encoding $f_{\mathrm{int}}(\xi_s, y_s)$ of one interaction and the interaction encoder	App. E
h_H, n	History encoding $\mathrm{Pool}(h_1, \ldots, h_n)$ and the number of interactions in the history	App. E
Communication loss and message revision		
$J_r(m)$	Loss of sending m to receiver r, $c_M(m) + L_I q_r^I(m) + L_C(1 - q_r^I(m))q_r^C(m)$	Eq. (6)
$c_M(m)$	Cost of sending message m	Eq. (6)
L_I, L_C	Weights of interpretation and execution failure, where $L_C = 0$ isolates interpretation	Eq. (6)
$\bar{J}_t(m)$	Posterior expected loss, $\sum_r \pi_t(r) J_r(m)$	Eq. (7)
m_0	Initial (original) message	Sec. 3
k, d	Index of a predefined mutable message feature and the number of such features ($d = 8$ in Experiment 4)	Sec. 3, Alg. 3
s_{rk}	Learned risk effect of mutable feature k for receiver r	Sec. 3, App. F
$\bar{s}_{tk}$	Posterior-weighted risk effect, $\sum_r \pi_t(r) s_{rk}$	Sec. 3
G_t, n_g	Guidance set: the at most n_g mutable features with the largest positive $\bar{s}_{tk}$ ($n_g = 3$ in Experiment 4), not the gain $G_t(B)$	Alg. 1
K_m, m'_j	Rewrite candidates per request ($K_m = 4$) and the j-th candidate	Sec. 3
V_{sem}, v_j	Semantic-equivalence verifier, which returns PASS or FAIL, and its decision on m'_j	Sec. 3, Alg. 1
$\mathcal{A}_t$	Valid action set, $\{m_0\} \cup \{m'_j : V_{\mathrm{sem}}(m'_j) = \mathrm{PASS}\}$	Sec. 3
$\widehat{m}_t^\star$	Revised message, $\arg\min_{m \in \mathcal{A}_t} \bar{J}_t(m)$	Sec. 3
$\widehat{m}^\star$	Message finally sent, after an optional query	Alg. 2
$\Delta f_k(m)$	Change in binary feature k from m_0 to the rewrite m	App. F
b_r	Intercept of receiver r in the linear risk model of Experiment 4: the effect of rewriting as such	App. F
α	Ridge penalty ($\alpha = 1$ in Experiment 4, $\alpha \in \{10, 100, 1000\}$ in Experiment 5), not the α of Theorem 4.1	App. F
Receiver queries and VoII		
ξ, $\mathcal{Q} = \{\xi_1, \ldots, \xi_M\}$	Calibration query and the fixed bank of M queries, kept separate from the test tasks	Sec. 3, App. E
Y_ξ, $\mathcal{Y}_\xi$, y	Response signature of query ξ (binary execution success in Experiment 5), its value set, and an observed response	App. E
$\widehat{P}(y \mid R = r, \xi)$	Smoothed likelihood $\widehat{P}(Y_\xi = y \mid R = r, \xi)$ of response y for receiver r	Sec. 3, App. E
$\widehat{P}(y \mid H_t, \xi)$	Predictive response distribution, $\sum_r \pi_t(r) \widehat{P}(y \mid R = r, \xi)$	Sec. 3
$\pi_{t+1}^{\xi,y}$, π_{t+1}	Posterior after response y to query ξ (Bayes update) and the posterior after the query actually asked	Eq. (8), Alg. 2
$\mathrm{IG}(\xi)$	Information gain, the expected entropy reduction $H(\pi_t) - \mathbb{E}_y H(\pi_{t+1}^{\xi,y})$	Sec. 3
V_0	Value without a query, $\min_{m \in \mathcal{A}_t} \sum_r \pi_t(r) J_r(m)$	Sec. 3
V_ξ	Expected value after query ξ, with $\mathcal{A}_t$ held fixed	Eq. (9)
$\mathrm{VoII}_t(\xi)$	Value of interpretation information, $V_0 - V_\xi$	App. C.2
$\mathrm{NetVoII}(\xi)$, c_ξ	Net value of query ξ, $V_0 - V_\xi - c_\xi$, and the cost of the query	Eq. (10)
$\xi^\star$	Query with the largest NetVoII, asked only if this value is positive	Sec. 3
Evaluation		
ΔY^I	Difference in Y^I between two conditions, in percentage points (ours minus the comparison)	Tab. 2
p	Probability of the posterior mode	App. I.3
c	Query cost in Experiment 5 ($c = 0.001$ unless stated otherwise)	App. G.4
$1 - Y^I - c\,r$, r	Net utility of a query policy and its query rate, the share of episodes queried (not a receiver)	Sec. 6, App. G.5

Table 4 (continued)

Symbol	Meaning	Defined in
$B\%$	Query quota: the share of episodes a policy may query (not the budget B in nats)	App. J.3
$Y^I + (1 - Y^I)Y^A$	Combined loss that also scores execution	App. J.4
Decision-theoretic results		
C	Context $(z^\star, x_S)$	Sec. 4
E, e, $\pi(e)$	Receiver information available to the sender, a value of it, and its distribution	Sec. 4, App. C.1
m_j, J	Candidate message j and the number of candidates	Sec. 4
F_j	Interpretation failure when m_j is sent	Sec. 4
$p_j(e)$	Failure probability of m_j given $E = e$, $\mathbb{P}(F_j = 1 \mid E = e, C)$	Sec. 4
c_j, $J_j(e)$	Cost of m_j and its loss $c_j + L_I p_j(e)$	Sec. 4
Δ_E	Value of receiver conditioning, $\min_j \mathbb{E}[J_j(E)] - \mathbb{E}[\min_j J_j(E)]$	Eq. (11)
j_0, $j^\star(E)$	Receiver-agnostic minimiser $\arg\min_j \mathbb{E}[J_j(E)]$ and receiver-aware minimiser $\arg\min_j J_j(E)$	Thm. 4.1
γ, α	Loss margin and the probability of reaching it, in the bound $\Delta_E \geq \alpha\gamma$	Thm. 4.1
I_t	Information state (H_t, C)	Sec. 4
B	Budget, in nats, of receiver information that may influence the message	Eq. (14)
M	Message as a random variable (not the bank size)	Eq. (14)
$\rho(m \mid r, I_t)$, $\bar{\rho}(m \mid I_t)$	Message policy, a channel from receiver type to message, and its marginal $\sum_r P(r \mid I_t)\rho(m \mid r, I_t)$	Eq. (14), App. C.2
$L(m, r)$, $P(r \mid I_t)$	Loss of sending m to receiver type r and the belief over receiver types given I_t	Eq. (14), App. C.2
$D_t(B)$	Lowest expected loss when at most B nats of receiver information may influence the message	Eq. (14)
$G_t(B)$	Gain from receiver information, $D_t(0) - D_t(B)$	Sec. 4
Used only in the proofs		
$p(e)$, $\bar{p}_j$	Vector $(p_1(e), \ldots, p_J(e))$ of failure probabilities and the mean failure probability $\mathbb{E}[p_j(E)]$	App. C.1
$g(u)$	$\min_j\{c_j + L_I u_j\}$ for $u \in [0, 1]^J$, so that $\Delta_E = g(\mathbb{E}[p(E)]) - \mathbb{E}[g(p(E))]$	App. C.1
$r \in \mathcal{R}$, $m \in \mathcal{A}_t$	Receiver types and messages	App. C.2
$B_1 < B_2$, $\mathcal{F}_1$, $\mathcal{F}_2$	Two information budgets and the sets of policies feasible under each	App. C.2
$\rho_1^\star$, $\rho_2^\star$, P_1, P_2	Optimal policies under B_1 and B_2 and their joint distributions of (m, r) (in Section C.1, P_1 and P_2 are $\mathrm{Ber}(p_j(E))$ and $\mathrm{Ber}(\bar{p}_j)$)	App. C.2
μ, B_μ	Mixing weight in $[0, 1]$ and the mixed budget $\mu B_1 + (1 - \mu)B_2$	App. C.2
P_μ, D_{t,P_μ}	Mixture $\mu P_1 + (1 - \mu)P_2$ of the two joint distributions and its expected loss	App. C.2
λ	Cost per nat of receiver information in the rational-inattention relaxation	App. C.2
$\mathcal{S}[\rho]$, $\chi(r)$	Objective of the relaxation with the normalisation constraint and the Lagrange multiplier of that constraint	App. C.2
$\rho_\lambda^\star$, $Z_\lambda^{I_t}(r)$	Optimal policy of the relaxation, of Gibbs form, and its normalising constant	App. C.2

C Proofs and Additional Formal Results

C.1 Receiver-Conditioning Value

This subsection proves Theorem 4.1:

Decision-value identity $\Delta_E \geq 0$:

We rewrite $\Delta_E = \mathbb{E}[J_{j^0}(E)] - \mathbb{E}[J_{j^\star}(E)] = \mathbb{E}[J_{j^0}(E) - J_{j^\star}(E)]$ based on the definition of Δ_E, receiver-agnostic minimiser j_0 and receiver-aware minimiser $j^\star(E)$[1]. From the definition of expectation, we have:

$$\Delta_E = \sum_e \pi(e)\left(J_{j_0}(e) - J_{j^\star}(e)\right) = \sum_e \pi(e)\left(J_{j_0}(e) - \min_j J_j(e)\right)$$

where $\pi(E)$ is the prior distribution over E, and satisfies $\pi(e) \geq 0$. By the definition of the minimising element j, we have $J_{j_0}(e) - \min_j J_j(e) \geq 0$, and hence $\Delta_E \geq 0$.

Characterisation of $\Delta_E = 0$:

Since $\pi(E)$ is the distribution of receiver information available to the sender, we have $J_{j_0}(E) = \min_j J_j(E)$ almost surely iff $\Delta_E = 0$, meaning that the receiver-agnostic cost $J_{j_0}(E)$ is equal to the receiver-aware cost for all receivers.

Lower bound $\Delta_E \geq \alpha\gamma$:

For $\alpha, \gamma > 0$, we assume:

$$\mathbb{P}\left(J_{j_0}(E) - \min_j J_j(E) \geq \gamma\right) \geq \alpha \tag{16}$$

Since $J_{j_0}(E) - J_{j^\star}(E) \geq 0$, according to Markov's inequality, we have:

$$\mathbb{P}\left(J_{j_0}(E) - \min_j J_j(E) \geq \gamma\right) \leq \frac{\mathbb{E}[J_{j_0}(E) - \min_j J_j(E)]}{\gamma} \tag{17}$$

The numerator on the right-hand side of (17) is Δ_E. Bounding the left-hand side with the assumption (16) and multiplying through by $\gamma > 0$, we have $\Delta_E \geq \alpha\gamma$.

Information-theoretic upper bound of Δ_E:

For each receiver state e, we define $p_j(e) := \mathbb{P}(F_j = 1 \mid E = e, C) \in [0, 1]$, and collect these scalar failure probabilities into $p(e) := (p_1(e), \ldots, p_J(e)) \in [0, 1]^J$.

Defining $g(p_1, \ldots, p_J) = \min_j\{c_j + L_I p_j\}$, we can re-express the decision-value as $\Delta_E = g(\mathbb{E}[p(E)]) - \mathbb{E}[g(p(E))]$, where the function g acts on the finite-dimensional probability vector defined as

$$g : [0, 1]^J \to \mathbb{R},\ g(u) := \min_j\{c_j + L_I u_j\}$$

Since each $u \mapsto c_j + L_I u_j$ is affine, g is the pointwise minimum of finitely many affine functions and is therefore concave. Moreover, g is L_I-Lipschitz with respect to the ℓ_∞-norm, i.e. $|g(p) - g(q)| \leq L_I\|p - q\|_\infty$. Therefore, we have:

$$\begin{aligned}\Delta_E &= g(\mathbb{E}[p(E)]) - \mathbb{E}[g(p(E))] = \mathbb{E}[g(\mathbb{E}[p(E)]) - g(p(E))] \\ &\leq L_I\mathbb{E}[\|\mathbb{E}[p(E)] - p(E)\|_\infty] \leq L_I\mathbb{E}[\|\mathbb{E}[p(E)] - p(E)\|_2]\end{aligned} \tag{18}$$

Noticing that for $\mathbb{E}[\|\mathbb{E}[p(E)] - p(E)\|_2]$, we have according to Jensen's inequality the bound

$$\left(\mathbb{E}\left[\|p(E) - \mathbb{E}[p(E)]\|_2\right]\right)^2 \leq \mathbb{E}\left[\|p(E) - \mathbb{E}[p(E)]\|_2^2\right] = \sum_{j=1}^{J} \mathrm{Var}[p_j(E)] \tag{19}$$

Taking the square root of (19) and inserting into (18), we find

$$\Delta_E \leq L_I\sqrt{\sum_{j=1}^{J} \mathrm{Var}[p_j(E)]} \tag{20}$$

Let $\bar{p}_j = \mathbb{E}[p_j(E)]$. Message transmission is modelled as a Bernoulli trial with failure probability $p_j(E)$ for the receiver-aware sender and $\bar{p}_j$ for the receiver-agnostic sender. Taking $P_1 := \mathrm{Ber}(p_j(E))$, $P_2 := \mathrm{Ber}(\bar{p}_j)$, we have

$$\|P_1 - P_2\|_1^2 \leq 2D_{\mathrm{KL}}(P_1\|P_2)$$

[1]Throughout this subsection, expectations and variances are taken over E conditional on the fixed context C, and this conditioning is suppressed in the notation.

via Pinsker's inequality. Since $\|P_1 - P_2\|_2 \leq \|P_1 - P_2\|_1$, we find:

$$\|\mathrm{Ber}(p_j(E)) - \mathrm{Ber}(\bar{p}_j)\|_2^2 \leq 2D_{\mathrm{KL}}(\mathrm{Ber}(p_j(E))||\mathrm{Ber}(\bar{p}_j)) \tag{21}$$

The left-hand side of (21) evaluates to $2|p_j(E) - \bar{p}_j|^2$. Dividing both sides by two gives $|p_j(E) - \bar{p}_j|^2 \leq D_{\mathrm{KL}}\left(\mathrm{Ber}(p_j(E))\|\mathrm{Ber}(\bar{p}_j)\right)$.

The variance of $p_j(E)$ is:

$$\mathrm{Var}[p_j(E)] = \mathbb{E}[(p_j(E) - \bar{p}_j)^2] \leq \mathbb{E}\left(D_{\mathrm{KL}}(\mathrm{Ber}(p_j(E)) \,||\, \mathrm{Ber}(\bar{p}_j))\right) \tag{22}$$

The right-hand side of (22) is the conditional mutual information $I(F_j; M|C)$, hence the variance is bounded by $\mathrm{Var}[p_j(E)] \leq I(F_j; M|C)$. Finally, from (20), the upper bound of the decision-value is given by

$$\Delta_E \leq L_I \sqrt{\sum_{j=1}^{J} I(F_j; E \mid C)}\,. \tag{23}$$

C.2 Information-to-Communication Frontier

This subsection proves Theorem 4.2:

In the sequel, we consider fixed $I_t = (H_t, C)$ and suppress the range of summations for clarity (they will be over the full space of the associated index unless otherwise indicated).

To connect this result to the controller, take $L(m, r) = J_r(m)$ and $P(r \mid I_t) = \pi_t(r)$. All query-related expectations and mutual information are evaluated under the same joint model $\pi_t(r)\widehat{P}(y \mid R = r, \xi)$. With estimated risks and likelihoods, the result bounds model-based VoII and does not by itself guarantee realised gains under model misspecification.

Non-increasing *The feasible set in Equation (14) expands with B, so $D_t(B)$ is non-increasing:*

The mutual information from the receiver-type $r \in \mathcal{R}$ and messages $m \in \mathcal{A}_t$ conditioned on I_t is given by

$$I(R; M \mid I_t) = \sum_{m,r} P(r|I_t)\, \rho(m|r, I_t)\, \log\left(\frac{\rho(m|r, I_t)}{\bar{\rho}(m)}\right) \tag{24}$$

where

$$\bar{\rho}(m|I_t) = \sum_{r} P(r|I_t)\rho(m|r, I_t) \tag{25}$$

Consider $I(R; M \mid I_t)$ as a functional of the conditional probability $\rho(m|r, I_t)$, for a fixed distribution $P(r|I_t)$.

Let $B_1 < B_2$ and define the feasible sets

$$\mathcal{F}_1 := \{\rho(m|r, I_t) \;:\; I(R; M \mid I_t) \leq B_1\} \qquad \mathcal{F}_2 := \{\rho(m|r, I_t) \;:\; I(R; M \mid I_t) \leq B_2\} \tag{26}$$

Because the mutual information (24) is positive semi-definite and continuous in $\rho(m|r, I_t)$, then a given $\rho(m|r, I_t) \in \mathcal{F}_1$ also lies in $\mathcal{F}_2$, and

$$\mathcal{F}_1 \subseteq \mathcal{F}_2 \tag{27}$$

Consider $D_t(B_1)$ and $D_t(B_2)$, with

$$D_t(B_i) = \min_{\rho_i(m|r,I_t) \in \mathcal{F}_i} \sum_{m,r} P(r|I_t)\, \rho_i(m|r, I_t)\, L(m, r) \tag{28}$$

and let $\rho_1^\star(m|r, I_t) \in \mathcal{F}_1$ be the conditional policy that provides the minimum $D_t(B_1)$. In determining the value of $D_t(B_2)$, there might be a $\rho_2(m|r, I_t) \in \mathcal{F}_2$ but $\notin \mathcal{F}_1$ might possibly provide a lower minimum of $\mathbb{E}[L(M, R)]$ so that $D_t(B_2) \leq D_t(B_1)$. If not, $\rho_1^\star(m|r, I_t) \in \mathcal{F}_1$ also lies in $\mathcal{F}_2$ by (27), and so at worst $D_t(B_2) = D_t(B_1)$.

This is true for all $0 \leq B_1 < B_2$, and so $D_t(B)$ must be non-increasing.

Convexity *Convexity follows from time sharing between two receiver-state-to-message channels with an independent selector:*

For $D_t(B)$ to be convex, we require that for $0 \le B_1 < B_2$, $\mu \in [0,1]$

$$D_t(\mu B_1 + (1-\mu)B_2) \le \mu D_t(B_1) + (1-\mu)D_t(B_2) \tag{29}$$

Let $\rho_1^\star(m|r,I_t)$ and $\rho_2^\star(m|r,I_t)$ be the conditional distributions that produce $D_t(B_1)$ and $D_t(B_2)$, and let

$$P_\mu(m,r|I_t) := \mu\, P(r|I_t)\rho_1^\star(m|r,I_t) + (1-\mu)\, P(r|I_t)\rho_2^\star(m|r,I_t) \tag{30}$$

be the joint distribution that interpolates between $P_1(m,r|I_t) = P(r|I_t)\rho_1^\star(m|r,I_t)$ and $P_2(m,r|I_t) = P(r|I_t)\rho_2^\star(m|r,I_t)$.

The expected loss is linear in $\rho(m|r,I_t)$, so we define $D_{t,P_\mu} := \mathbb{E}_{P_\mu}[L(M,R)]$ and see that

$$\begin{aligned}
D_{t,P_\mu} &= \mu \left(\sum_{m,r} P(r|I_t)\, \rho_1^\star(m|r,I_t)\, L(m,r) \right) + (1-\mu) \left(\sum_{m,r} P(r|I_t)\, \rho_2^\star(m|r,I_t)\, L(m,r) \right) \\
&= \mu \left(\min_{\substack{\rho(m|r,I_t): \\ I(R;M|I_t) \le B_1}} \sum_{m,r} P(r|I_t)\, \rho(m|r,I_t)\, L(m,r) \right) \\
&\quad + (1-\mu) \left(\min_{\substack{\rho(m|r,I_t): \\ I(R;M|I_t) \le B_2}} \sum_{m,r} P(r|I_t)\, \rho(m|r,I_t)\, L(m,r) \right) \\
&= \mu D_t(B_1) + (1-\mu) D_t(B_2)
\end{aligned} \tag{31}$$

For fixed $P(r|I_t)$, $I(R; M \mid I_t)$ is known to be convex in $\rho(m|r,I_t)$ (by suitably altering the analogous argument for the unconditioned mutual information in (Cover & Thomas, 2006)), so

$$I_{P_\mu}(R; M \mid I_t) \le \mu I_{P_1}(R; M \mid I_t) + (1-\mu) I_{P_2}(R; M \mid I_t) \tag{32}$$

where the right-hand side is at most $B_\mu := \mu B_1 + (1-\mu)B_2$, since each policy satisfies its respective information constraint.

Defining

$$D_t(B_\mu) := \min_{\rho(m|r,I_t)\,:\, I(R;M|I_t) \le B_\mu} \mathbb{E}[L(M,R)] \tag{33}$$

we see that the inequality (32) implies that a $\rho(m|r,I_t)$ can be found corresponding to a loss D_{t,P_μ} needing $B \le B_\mu$. Since $D_t(B)$ is non-increasing, this implies

$$D_t(B_\mu) \le D_{t,P_\mu} = \mu D_t(B_1) + (1-\mu) D_t(B_2) \tag{34}$$

and hence $D_t(B)$ is convex.

Hence $G_t(B) = D_t(0) - D_t(B)$ is non-decreasing and concave.

Bound on VoII(ξ): Taking a fixed set of actions $\mathcal{A}_t$, the no-query decision value value $V_0 = \min_{m \in \mathcal{A}_t} \sum_r \pi_t(r) J_r(m)$, and the expected query value V_ξ defined in (9), we can construct the value of interpretation information for a given query ξ:

$$\text{VoII}_\xi = V_0 - V_\xi$$

For a practical query ξ, any final message chosen only through its observed response satisfies the conditional Markov relation

$$R \to Y_\xi \to M \qquad \text{given } I_t.$$

The data processing inequality (Cover & Thomas, 2006) gives

$$I(R; M \mid I_t) \le I(R; Y_\xi \mid I_t).$$

The induced message policy is feasible at information budget $I(R; Y_\xi \mid I_t)$, so

$$V_\xi \ge D_t(I(R; Y_\xi \mid I_t))\,.$$

Since $D_t(0) = V_0$, subtracting gives

$$\mathrm{VoII}_t(\xi) \leq G_t(I(R; Y_\xi \mid I_t)) \,.$$

Relation to rational inattention and information bottlenecks *We relate our formulation of VoII to some prior work in information and decision theory:*

Our sender S acts as an agent facing the decision problem of picking a message $m \in \mathcal{A}_t$, given imperfect information about the payoff-relevant receive type $r \in \mathcal{R}$ with some prior belief $P(r|I_t)$.

The action S takes will come at some loss $L(m, r)$ depending on the mismatch between m and r, while the information strategy S takes comes at a cost which will be taken to be a constant $\lambda > 0$ times the mutual information $I(M; R|I_t)$. This setup is similar to the "rational inattention" work described in (Sims, 2003; 2010; Matějka & McKay, 2015; Maćkowiak et al., 2023) and elsewhere, and the "information bottleneck" method from (Tishby et al., 2000).

The optimisation problem facing a sender S is to find the action strategy $\rho(m|r, I_t)$ satisfying

$$\min_{\rho(m|r,I_t)} \mathbb{E}[L(M, R)] + \lambda I(R; M \mid I_t) \tag{35}$$

where the expected loss is defined as

$$\mathbb{E}[L(M, R)] = \sum_{m,r} P(r|I_t)\rho(m|r, I_t)L(m, r) \tag{36}$$

and

$$I(R; M \mid I_t) = \sum_{m,r} P(m, r) \log\left(\frac{P(m, r)}{\bar{\rho}(m|I_t)P(r|I_t)}\right) = \sum_{m,r} P(r|I_t)\rho(m|r, I_t) \log\left(\frac{\rho(m|r, I_t)}{\bar{\rho}(m|I_t)}\right) \tag{37}$$

subject to the normalisation condition

$$\sum_m \rho(m|r, I_t) = 1\,, \qquad \forall r \in \mathcal{R} \tag{38}$$

Here $\bar{\rho}(m|I_t) = \sum_r P(r|I_t)\rho(m|r, I_t)$ is the conditional distribution of m given I_t having marginalised over all receiver types.

The optimisation problem can be expressed in terms of the following functional:

$$\begin{aligned}\mathcal{S}[\rho(m|r, I_t)] = &\sum_{m,r} P(r|I_t)\rho(m|r, I_t)\left(L(m, r) + \lambda \log\left(\frac{\rho(m|r, I_t)}{\bar{\rho}(m|I_t)}\right)\right) \\ &+ \sum_r P(r \mid I_t)\chi(r)\left(\sum_m \rho(m \mid r, I_t) - 1\right)\end{aligned} \tag{39}$$

where $\chi(r)$ acts as a Lagrange multiplier enforcing (38).

Taking the variation with respect to $\rho(m|r, I_t)$, we find

$$\begin{aligned}\frac{\delta\mathcal{S}}{\delta\rho(m|r, I_t)} = P(r|I_t)&\left(L(m, r) + \lambda \log\left(\frac{\rho(m|r, I_t)}{\bar{\rho}(m|I_t)}\right) + \lambda - \frac{\lambda}{\bar{\rho}(m|I_t)}\sum_{r'}\left(P(r'|I_t)\rho(m|r', I_t)\right)\right) \\ &+ \chi(r)P(r|I_t)\end{aligned} \tag{40}$$

where the third and fourth terms in brackets cancel.

Finding the extremum $\rho^\star_\lambda(m, r|I_t)$ of (39) by setting (40) to zero, we have

$$\log\left(\frac{\rho^\star_\lambda(m|r, I_t)}{\bar{\rho}^\star_\lambda(m)}\right) = -\frac{L(m, r)}{\lambda} - \frac{\chi(r)}{\lambda} \tag{41}$$

Defining $\log Z^{I_t}_\lambda(r) := \chi(r)/\lambda$, the optimal action policy is

$$\rho^\star_\lambda(m|r, I_t) = \frac{\bar{\rho}^\star_\lambda(m|I_t)\exp\left(-L(m, r)/\lambda\right)}{Z^{I_t}_\lambda(r)} \tag{42}$$

From (38), we have

$$1 = \sum_{m'} \frac{\vec{\rho}^{\star}_{\lambda}(m'|I_t)\exp\left(-L(m',r)/\lambda\right)}{Z^{I_t}_{\lambda}(r)} \qquad \Rightarrow Z^{I_t}_{\lambda}(r) = \sum_{m'} \vec{\rho}^{\star}_{\lambda}(m'|I_t)\exp\left(-L(m',r)/\lambda\right)$$

so $Z^{I_t}_{\lambda}(r)$ is a normalisation constant.

From equation (42), we see that as one expects, when $\lambda \to \infty$ the policy becomes receiver-agnostic, and as $\lambda \to 0$ it concentrates on receiver-specific loss minimisers, as expected from e.g. (Tishby et al., 2000; Matějka & McKay, 2015).

D Interpretation-Probe Construction

D.1 Task Identities and Surface Realisations

The benchmark defines semantic task identities separately from their natural-language wordings. For task z, let

$$\mathcal{E}(z) = \{T_z^{(1)}, \ldots, T_z^{(R)}\}$$

be validated descriptions of that task. The sender and probe banks are disjoint:

$$m \sim \mathcal{E}_{\text{send}}(z^{\star}), \qquad \widetilde{T}_z \sim \mathcal{E}_{\text{probe}}(z), \qquad \mathcal{E}_{\text{send}} \cap \mathcal{E}_{\text{probe}} = \varnothing.$$

The receiver chooses among task identities rather than matching the sender's wording.

When prompt-template resources give several transformations of the same record, we extract a task signature

$$s(T) = (\mathcal{I}_T, \mathcal{C}_T, \text{op}_T, \mathcal{O}_T, \mathcal{F}_T),$$

which lists the visible input fields, extra context, semantic operation, output meaning, and output format/type. Candidate pairs are first labelled as equivalent, extra-information variants, different tasks, or unclear. Only pairs already labelled as different tasks are then ranked by semantic closeness. This order keeps an opaque distance score from deciding task equivalence.

A strong distractor shares the same underlying example/interface and information, differs in one or two task-semantic components, and gives no trivial output-format cue. Easy/matched/hard difficulty is stored separately from the semantic relation.

D.2 Task Bank and Splits

We apply this protocol to a bank of concrete task/message examples drawn from FreebaseQA, WebQSP, GrailQA, SQ-WD, PopQA, and Mintaka. Prompt-template resources such as PromptSource (Bach et al., 2022) help find neighbouring task transformations, but the benchmark consists of concrete instances with a known intended task, plausible nearby alternatives, and an executable or otherwise objective task score. Splits are made by normalised question group within each task family, so paraphrases and closely related transformations never cross the train, validation, and test sets. Interpretation failures are rare (4.4% of test pairs). If extra failure cases are used to train or analyse the predictor, this enrichment stays in training/analysis data and is tagged by provenance. Calibration and headline evaluation always use untouched natural-prevalence validation and test sets.

The six datasets share one useful property: every question has a gold answer, so execution can be scored without a judge, while the questions range widely in style, from trivia to compositional queries. We split each dataset 70/15/15 by normalised question group, keeping related transformations within the same split.

D.3 Probe Robustness Checks

The headline measurement uses a correct task option, one contrast task, and None of these. Instead of a separate paraphrase study, every message–receiver pair gets six probe instances with different wording, and the label averages over them. Five temperature-0 repeats of the full test protocol reproduce every dataset's Y^I within 0.07 percentage points. Averaging over the six probes spreads wording and order effects over all messages but does not remove a systematic position effect, which

we check below. A dedicated paraphrase-bank study and a construct-validation set of clear/easy, confusing/easy, and clear/hard messages are left for future work. NONE OF THESE counts as a wrong task but is kept as a separate raw outcome. Unparsed probe responses are dropped from the denominator of Equation (5).

Position of the intended task. The intended task is listed first, second, or last in 33.1, 33.6, and 33.2% of the test probes, and Table 5 gives the probe error rate by this position. Nemotron misreads 24.5% of the probes that list the intended task last, against 3.1% and 4.5% when it is first or second. It picks the last option in 25.8% of probes, against 33.3% for the other receivers, and nearly all of its errors at the last position are picks of the contrast task (24.43 of 24.49 points). The effect is one of position, not wording: under each of the six probe wordings, Nemotron's error rate is 18.5–31.0% with the intended task last and 1.7–4.7% with it first or second. The other receivers show small position effects in both directions. When only probes with the intended task first or second are kept, the error rate still ranges from 1.45% (Qwen3.6) to 5.50% (Mistral), a ratio of 3.8 (95% CI 3.4–4.3, against 6.2 on all probes), and Mistral exceeds Qwen3.6 by 4.1 points [3.7, 4.4]. Within datasets the ratio is 1.7–2.6 on FreebaseQA, GrailQA, Mintaka, and WebQSP, 6.6 on PopQA, and 14.3 on SQ-WD. Qwen3.6 is the lowest receiver on all six datasets, and the highest is Mistral on four and Nemotron on two. Receiver heterogeneity therefore survives the position control, although it is smaller, and most of Nemotron's excess Y^I comes from the last position. Experiment 2 shows the same pattern. We rescore it against a label built only from probes with the intended task first or second, refitting on validation data either one global calibration map or one map per receiver. Per-receiver maps then add about 0.02 to across-receiver AUROC (A3: 0.750 to 0.770, full model: 0.754 to 0.772), against about 0.10 on the full label, and the full model still ranks slightly better within receivers (AUROC 0.756 vs. 0.751 for A3). Much of the across-receiver gain of Experiment 2 thus reflects Nemotron's position effect.

Table 5: Probe error rate (%) by the position of the intended task (test split, parsed probes). Last column: probes with the intended task first or second, with question-cluster bootstrap 95% CIs.

Receiver	First	Second	Last	All	First or second [95% CI]
Qwen3.6	1.28	1.61	2.33	1.74	1.45 [1.26, 1.65]
Gemma	1.59	2.49	2.57	2.22	2.04 [1.81, 2.28]
gpt-oss	3.27	2.08	1.63	2.33	2.68 [2.42, 2.95]
Nemotron	3.14	4.51	24.49	10.72	3.83 [3.57, 4.09]
Mistral	5.26	5.74	4.45	5.16	5.50 [5.15, 5.85]
Highest/lowest	4.1	3.6	15.0	6.2	3.8 [3.4, 4.3]

E RECEIVER-TYPE INFERENCE AND QUERY BANK

E.1 CALIBRATION BANK

The fixed query bank is

$$\mathcal{Q} = \{\xi_1, \ldots, \xi_M\},$$

built from training/calibration instances only. Each item stores its task/message content, a fixed response parser, and repeated genuine responses from every receiver type. Responses are parsed into a discrete signature $Y_\xi \in \mathcal{Y}_\xi$ and smoothed likelihoods

$$\widehat{P}(Y_\xi = y \mid R = r, \xi)$$

are estimated by Laplace/Dirichlet smoothing. Raw responses are kept for rescoring.

E.2 HISTORY ENCODER

A small receiver classifier encodes each interaction and pools over the history:

$$h_s = f_{\text{int}}(\xi_s, y_s), \qquad h_H = \text{Pool}(h_1, \ldots, h_n),$$

$$\pi_\omega(r \mid H) = \text{softmax}_r(W h_H + b).$$

Mean or sum pooling is the default, and the main experiment needs no recurrent or transformer history encoder. Training uses receiver-type cross-entropy, and probabilities are calibrated on held-out

histories. For $n = 0$ we use an explicit receiver prior, which is uniform unless there is an independent reason to choose otherwise.

Histories are built by subsampling stored interactions from one receiver, and every response is a genuine output of that receiver. Query choice, history length, filenames, application programming interface (API) metadata, and other incidental fields are matched across receiver types, so the classifier cannot exploit leakage.

F Message-Revision Implementation

Estimators. PIR is the quantity being estimated, not one fixed predictor. Experiment 2 estimates it on natural messages with the heads of Section G.1. Rewrites differ in distribution from natural messages (for example, they state the output or turn the question into a request), so Experiments 4 and 5 use candidate-level estimators fitted to measured rewrites: a linear model of mutable features (Experiment 4) and paired risk heads (Experiment 5), both described below.

Feature scores. The feature set is fixed in advance: features are labelled as immutable/task-inherent or mutable/style-structural before rewriting. What is learned from training data is the risk effect s_{rk} of mutable feature k for receiver r, such as the difference in measured interpretation-failure rate when the feature/value is present versus absent. Under posterior π_t, use

$$\bar{s}_{tk} = \sum_r \pi_t(r) s_{rk}$$

and pass only positive, currently relevant mutable features to the rewriter.

Generation. Generate a fixed number $K_m = 4$ of candidates, with the instruction to keep the exact task meaning while reducing the selected mutable features. All generate-and-select methods share this fixed budget.

Semantic verifier. The verifier is a separate stateless call to the same model version as the rewriter. It receives only $(z^\star, x_S, m_0, m'_1, \ldots, m'_{K_m})$ and returns PASS/FAIL plus a short failure reason for each candidate. It never sees the receiver identity or posterior, PIR scores, risk-feature values, or candidate preference. The original m_0 is always kept, and failed candidates are logged, not regenerated.

Selection. For every valid candidate, evaluate every receiver-conditioned head and compute Equation (7). Select by posterior expected loss, not by the MAP receiver. After a real query response, update the receiver posterior and re-score the same fixed action set.

Experiment 4 implementation. Experiment 4 uses a linear risk model with eight binary mutable features: the output is not stated, the question is replaced by a request, the instruction line is missing, surface errors (lower-case start or no question mark), a long question, a question word that is not first, a parenthetical, and a pronoun. For each receiver r, the change in soft Y^I from the original to a rewrite is regressed (ridge, $\alpha = 1$) on the change Δf in these features. The intercept b_r is the effect of rewriting itself, and the slopes are the scores s_{rk} (Figure 7a). The model is fitted on 4,255 rewrites of 600 training questions (100 per dataset), each measured by all five receivers. Guidance lists up to three features with the largest $\bar{s}_{tk}$, among those for which $\bar{s}_{tk}$ and its 2.5% bootstrap quantile (500 question-cluster resamples) are both positive, each with a fixed edit instruction. Selection sends the message with the lowest predicted risk $\sum_r \pi_t(r)\,(b_r + \sum_k s_{rk}\Delta f_k)$ and keeps the original unless the predicted gain exceeds a margin of 0.001, chosen on 300 validation questions. The rewriter is DeepSeek V4 Flash (DeepSeek-AI, 2026) at temperature 0.4 with $K_m = 4$. Each call holds eight requests from one condition only, so that conditions cannot copy one another. The instruction for an unstated output asks the rewriter to state the output in the question's own words (e.g., "Give the name of the ⟨thing asked for⟩.") and without task-type words such as kind, class, category, type, attribute, or label. Deterministic checks reject candidates that, for example, repeat the original, add a gold answer, change numbers, negations, or protected qualifiers, or add task-type words, and the verifier above then checks the rest. The posterior π_t is the type model of Experiment 3 after five stored responses from the audit bank, and all conditions of a question share this history.

Paired risk heads for Experiment 5. Experiment 5 chooses among a fixed pool of generic rewrites. Candidates are generic clarity rewrites generated without feature guidance (four per message by DeepSeek V4 Flash, English v1.1 prompts), verified by the stateless batched call above, and embedded with Qwen3-Embedding-8B (Zhang et al., 2025). The scoring head is trained on paired candidate measurements: for 600 training questions per dataset, every accepted candidate and the original were measured by all five receivers (one fresh repeat, six probes and one execution call each), giving one soft Y^I per question–message–receiver. Inputs are 64-component principal component analysis (PCA) projections (fitted on training texts only) of the embeddings of the message, the intended task, and the original message, plus the message–original difference, surface features, edit geometry, and a receiver one-hot. The MLP has hidden widths 128 and 64, dropout 0.15, learning rate 10^{-3}, weight decay 0.01, and at most 100 epochs with early stopping (patience 12) on 80 held-out validation questions, and two seeds are ensembled. Three model families are compared: an absolute soft-risk head, a paired-loss head that adds eight times the squared error of the predicted within-question risk differences, and ridge regressions of the within-question risk change on text and edit features ($\alpha \in \{10, 100, 1000\}$). Scores are calibrated with a common positive slope and per-receiver intercepts on 60 calibration questions, and the family together with the minimum predicted improvement required to leave the original ($\{0, 0.002, 0.005\}$) was chosen on 60 further policy-selection questions by the measured Y^I of the posterior policy. The selections are the absolute head (FreebaseQA, WebQSP, SQ-WD), the paired head (GrailQA), and ridge (PopQA with $\alpha = 1000$, Mintaka with $\alpha = 10$). The posterior π_t is the type model of Experiment 3 applied to five stored responses drawn from the audit bank. The same history and the same candidate pool serve all conditions of a question, and the true identity is used only to construct the environment and the true-identity reference. Weights, calibration, and margins were frozen before any outcome of the replication cohort of Experiment 5 was read. Its questions are the first 400 valid probe instruments per dataset in a frozen hash order over questions never used before (500 attempts per dataset, extended under a pre-registered amendment to at most 1,200 for Mintaka, where about half of the attempts gave a valid instrument, for 3,044 attempts in total). Rejected attempts were never selected on receiver outcomes.

G Experimental Protocols

The experiments test the claims in the same order as the framework. Experiment 1 asks whether real LLM receivers reconstruct tasks differently and whether interpretation failure is distinct from execution failure. Experiment 2 asks whether this lower-dimensional risk is predictable before sending and whether privileged cross-receiver supervision adds information beyond the message alone. Experiment 3 asks how much ordinary behavioural history, rather than model identity, reveals about the latent receiver type. Experiment 4 then tests whether posterior-guided rewriting reduces *measured* misinterpretation, and Experiment 5 tests whether VoII allocates queries better for the interpretation objective. Experiments 1–3 fix the message (one template, no sender model), so every receiver reads the same text and differences come from the receiver. Experiments 4 and 5 then let an LLM revise the message and control queries. Experiments 1–3 use the frozen test split, Experiment 4 uses 1,579 held-out questions from outside the test split, and Experiment 5 uses the test split as its primary cohort and a separate set of 2,400 new questions as a replication. This section gives the parts of the design in Section 5 that do not fit in the main text. Implementation details and hyperparameters are in Section J.4.

G.1 Labels and Predictor

Labels. Each probe offers three options: the intended task, the contrast task, and NONE OF THESE. Probe text is generated per message by DeepSeek V4 Flash, passed through deterministic checks and a verifier, and issued $K_p = 6$ times with independently varied wording and option order. The label $\widetilde{y}^I$ is the fraction of parsed probes that choose a wrong option (Equation (5)). Execution is one separate answer call, scored by gold-alias containment. The test split alone uses 198,870 probe calls and 33,145 answer calls.

Predictor. The heads q^I and q^C are separate small MLPs on frozen Qwen3-Embedding-8B features of m and of the original question x_S (our proxy for sender context), with an optional difference vector $m - x_S$ and a 16-dimensional learned receiver embedding e_R. Each MLP output is added to a logistic offset fitted first on the training data: a per-receiver intercept and slope on the scalar gap g,

the root-mean-square difference between the standardised embeddings of x_S and m (A3 uses one intercept and slope). q^I is trained with BCE on $\widetilde{y}^I$, and q^C with BCE on Y^A weighted by $1 - \widetilde{y}^I$. Configurations (27 per dataset, 3 seeds) are selected on validation NLL only. A Platt map (Platt, 2000) with a per-receiver slope and intercept and shared terms in g is then fitted on the calibration part of the validation split (A3 uses one global map). Candidates in the rewrite experiments are scored by the estimators of Section F, and only the combined objective of Experiment 5 reuses this q^C.

G.2 Receiver Inference (Experiment 3)

Experiment 3 uses a calibration bank of 384 tasks separate from the test messages, with 9,600 archived responses from the five receivers, from which we build 20,800 histories of length 1 to 20. The type model embeds each response (PCA to 32 dimensions), scores it against its query with a small interaction MLP, mean-pools over the history, and outputs a five-way softmax with temperature calibration (Guo et al., 2017). Tasks are split by role (192 fit, 48 early stop, 48 calibration, 96 audit), and every reported number uses 960 audit histories per length. A shuffled-history control pairs each query with responses from a random receiver.

G.3 Message Revision (Experiment 4)

Experiment 4 tests the full revision step of Section 3: learned risk guides the rewriter and then selects the sent message (Section F). All conditions share the rewriter, the intended task, four new candidates per request, the checks, and the verifier. They differ only in where learned risk is used. The conditions are the original message, the generic rewrite (the first valid candidate of a request for clarity only), selection only (the same generic pool, selected by the risk model), posterior only (the rewriter sees π_t but no risk features, and the risk model selects), ours (guidance and selection), guidance only (our pool, first valid candidate), ours with a uniform belief or with the true identity in place of π_t (a diagnostic, not a deployable method), one feature per step (four steps, each editing the top risk feature of the current message, then selection), and ours before the fix of the output instruction (Section I.4). The comparisons of ours with the original, the generic rewrite, selection only, and posterior only were pre-registered, with Holm correction and a non-inferiority margin of one point on Y^A. Before the main run, a check on 300 validation questions (50 per dataset) had to show that no fixed condition adds task-type words, that candidate pools do not collapse to the original, and that ours lowers Y^I against the original. All three held. The main run uses 1,579 held-out questions, 300 per dataset in a frozen hash order (WebQSP has only 79 unused questions). Of these, 1,304 have the entity-versus-category contrast and 275 the value-versus-attribute contrast. They come from the training and validation splits of Experiments 1–3, not from the test split, and none was used to fit the risk model, to choose the margin, or to develop the prompts. The questions, all sent messages, the measurement protocol, and the analysis script were frozen before any outcome was read. Each distinct message is measured with each receiver by 36 probes (six wordings times six option orders, with a neutral NONE option) and one execution call. Here Y^I is the share of the 36 probes that choose a wrong option, with unparsed probes counted as wrong, so its level is not comparable with Experiments 1–3. A blind audit by Gemini 3.8 Flash judges whether each sent message keeps the task, and a pre-registered sensitivity analysis sends the original in place of every flagged message.

G.4 Query Control (Experiment 5)

Experiment 5 tests the query rule of Section 3. The primary cohort is the frozen test split: 6,629 questions, five receivers, and three fresh answer repeats (99,435 episodes). A separate cohort of 2,400 new questions (36,000 episodes) is analysed as a replication. Every episode starts from five stored responses and the calibrated posterior π_t of Experiment 3. The candidate pool is the original message and its four generic rewrites with frozen verifier decisions, scored by the frozen paired heads of Section F, and no head was refit. The query bank is the 192 fit-role tasks of Experiment 3 with their archived responses. A query returns whether the receiver solved the task, and the posterior is updated by an exact Bayes step with Laplace-smoothed likelihoods. A policy may ask at most one query per episode and then sends the candidate with the lowest posterior expected interpretation risk. NetVoII (Equation (10)) is compared with IG (the query with the largest expected entropy reduction), random querying (three pre-declared seeds), and, as a secondary comparator, an adapted VoI rule (Dong et al., 2026) that keeps the same posterior, bank, candidates, and channel but replaces the

learned heads by DeepSeek V4 Flash utility estimates. Under every policy, ties in expected loss go to the earlier candidate, so the original wins a tie. Three references fix the scale: never query, always query, and the true receiver identity in place of π_t, a diagnostic reference rather than an upper bound (Section I.5). Budgets of 10, 20, 30, and 50% of episodes are exact quotas filled in each policy's own ranking order. The strict rule instead queries only where $\mathrm{NetVoII} > 0$ at $c = 0.001$ (five other costs in Table 16), and IG and random querying are re-run at its query count. All schedules were frozen before any receiver call.

G.5 Metrics and Statistics

We report AUROC, AUPRC, Brier score (Brier, 1950), NLL, and ECE (Naeini et al., 2015) for prediction, and accuracy, NLL, Brier, and ECE for receiver inference. Main-text prediction metrics are computed within each receiver and macro-averaged, so they measure how well messages are ranked for a known receiver. Pooled metrics, which also reward ranking across receivers, appear in Table 10. Intervals are 95% paired bootstraps (Efron & Tibshirani, 1994) over question groups (1,000 resamples) for Experiment 2, binomial intervals for Experiment 3, question-level normal intervals for Experiment 1, and paired bootstraps over questions (10,000 resamples, Holm-corrected over the four pre-registered comparisons) for Experiment 4. For Experiment 5 we report the share of episodes queried, measured Y^I and task success $1 - Y^A$, net utility $1 - Y^I - c\,r$ for query rate r, and the share of episodes whose posterior mode is the true receiver, with paired question-cluster bootstraps (2,000 resamples) Holm-corrected over the eight pre-registered net-utility contrasts (NetVoII against IG and against random querying at the four budgets). ECE uses 10 equal-mass bins in Experiment 2 and 10 equal-width bins in Experiment 3.

G.6 Scope of the Reported Experiments

Three questions lie outside the reported experiments: a separate paraphrase/order robustness study (each pair already has six wording/order variants, five temperature-0 repeats reproduce Y^I within 0.07 points, and Section D.3 controls for option position), a step-by-step input ladder (because x_S is a proxy that fixes $z^\star$, the m and $(m, z^\star)$ steps are not separately identifiable), and open-set receivers. Experiment 4 uses one rewriter and does not compare the posterior with a MAP receiver. Experiment 5 compares NetVoII with never, always, and random querying, IG, an adapted VoI rule (secondary), and the true-identity reference, each with at most one query per episode. A self-confidence control and rankings by q^I, q^C, generic action risk, or receiver-agnostic PIR are natural extensions.

H Baselines and Ablations

H.1 Essential Prediction Baselines

The reported runs use A0–A5 plus the shuffled-receiver control and the receiver-calibrated A3, and all learned methods use the same question-group split and validation-only hyperparameter selection. A1 and A2 use DeepSeek V4 Flash as judge. Of 26,528 judge calls, 54 were unparsable and fall back to the training mean. Other candidate baselines (semantic entropy (Farquhar et al., 2024), a zero-shot receiver-aware judge, sampled receiver entropy, intent entropy, structured expected value of perfect information (Suri et al., 2026), and AgentAsk-style clarification (Lin et al., 2026a)) are outside the reported comparison.

H.2 Essential Ablations

The input-information ladder $q^I(m) \to q^I(m, z^\star) \to q^I(m, z^\star, x_S) \to q^I(m, z^\star, x_S, e_R)$ cannot be split step by step here: the sender-context input x_S is the original benchmark question, which already fixes $z^\star$, so the first two steps are not separately identifiable. We therefore report the last step, which adds the receiver information, in four forms: full receiver information, receiver information removed (A3), receiver information shuffled (Table 10), and receiver information used only to recalibrate A3 (Table 8), together with Y^I versus Y^A supervision (A4). For the receiver belief, the rewrite experiment compares the history posterior with a uniform prior and with the true identity.

Table 6: Baseline definitions.

ID	Method	Definition
A0	Base-rate priors	Global $P(Y^I = 1)$ and receiver-specific $P(Y^I = 1 \mid R)$ estimated on training data.
A1	Message-only ambiguity	Fixed strong judge sees m only and scores whether materially different task interpretations are possible.
A2	Semantic equivalence	Judge sees $(m, z^\star, x_S)$ and predicts whether m faithfully specifies the intended task, with no receiver information.
A3	Receiver-agnostic PIR	Same learned PIR architecture but remove e_R / receiver posterior information.
A4	Action-failure predictor	Same architecture/input budget trained against Y^A rather than Y^I, evaluated against both targets.
A5	Listener/ToM	Predict $P(\widehat{Z}_R = z \mid m, x, e_R)$ over probe task identities and use $1 - P(\widehat{Z}_R = z^\star)$ as risk.
–	Shuffled receiver	Full model trained with receiver identities randomly permuted across pairs, to test whether the gain requires the correct receiver.
–	Receiver-calibrated A3	A3 passed through the full model's calibration map (per-receiver slope and intercept, shared gap terms), refitted on the same validation questions, to test whether receiver information adds more than receiver-specific risk levels.

For query control, Experiment 5 compares NetVoII with IG, random querying, an adapted VoI rule with LLM utility estimates (a secondary comparator), never and always querying, and the true-identity reference, at exact query quotas and under the strict rule (Section 6). The contrast between IG and NetVoII directly tests the claim that interpretation-specific uncertainty and decision value are distinct.

For rewriting, Experiment 4 turns each part of the method off in turn: selection only and posterior only drop the guidance, guidance only drops the selection, and a uniform belief or the true identity replaces the posterior (Section G.3). One feature per step tests a sequential form of the guidance, at about twice the tokens and four dependent rounds (Section I.4).

I Additional Results

This section gives the full results behind Section 6, one subsection per experiment. Experiments 1–3 test the premises of the framework: interpretation risk differs across receivers (Experiment 1), it can be predicted before sending (Experiment 2), and behaviour carries information about the receiver type (Experiment 3). Experiments 4 and 5 test the two decisions built on these premises, message revision and receiver queries, and we relate their outcomes to Theorems 4.1 and 4.2.

I.1 Experiment 1: Interpretation and Execution Failures

Interpretation risk depends on the pair of receiver and message, not on the receiver alone. Figure 3a gives Y^I for every receiver and dataset. Within a dataset, the riskiest receiver fails 4.3 to 12.7 times as often as the least risky one (1.7 to 14.3 times when the intended task is listed first or second, Section D.3). Nemotron is the riskiest receiver on every dataset, but the least risky one changes: Qwen on four datasets, gpt-oss on GrailQA, and Mistral on Mintaka, although Mistral is the second riskiest receiver overall. A single score per receiver would miss these reversals, which is why the predictor conditions on both the message and the receiver. Heterogeneity alone does not make receiver information useful: by Theorem 4.1, it pays off only when it changes the preferred message, which Experiments 4 and 5 test.

Answer checks miss many interpretation failures. Figure 3b splits the misread pairs of each dataset by the answer check, and the last two columns of Table 7 give both off-diagonal cells. Pooled over datasets, 1.4% of all pairs misread the task yet pass the answer check, which is 32% of all misreads, and on FreebaseQA the share is 72%. The reverse case is far more common: 56.4% of pairs read the task correctly and still fail. An answer-based score therefore mixes the two failures, and Y^I has to be measured on its own, as the probes do.

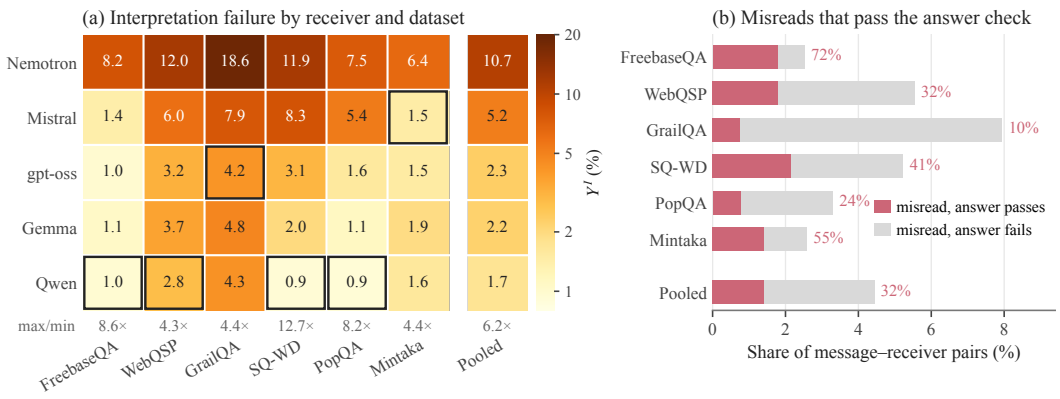


Figure 3: Interpretation failure depends on the receiver and the dataset (test split). (a) Y^I (%) per receiver and dataset on a log colour scale. Boxes mark the least risky receiver in each dataset, and the bottom row gives the ratio of the highest to the lowest receiver. (b) Misread pairs split by the answer check. Labels give the share of misreads that pass.

Table 7: Interpretation (Y^I) and execution (Y^A) failures per dataset (test split, pooled over five receivers, 95% question-level normal intervals).

		Marginal rates (%)		Off-diagonal cells (%)	
Dataset	Pairs	Y^I [95% CI]	Y^A [95% CI]	misread, pass	read, fail
FreebaseQA	6,550	2.5 [1.7, 3.4]	24.0 [21.7, 26.4]	1.8	23.3
WebQSP	1,095	5.5 [2.5, 8.6]	54.1 [47.5, 60.7]	1.8	50.3
GrailQA	6,430	7.9 [6.5, 9.4]	89.6 [87.9, 91.3]	0.8	82.4
SQ-WD	7,090	5.2 [4.1, 6.4]	61.9 [59.3, 64.4]	2.2	58.8
PopQA	7,255	3.3 [2.4, 4.2]	75.7 [73.5, 77.9]	0.8	73.2
Mintaka	4,725	2.6 [1.6, 3.6]	40.3 [37.2, 43.4]	1.4	39.1
Pooled	33,145	4.4	59.5	1.4	56.4

I.2 Experiment 2: Prospective PIR Prediction

Receiver information makes the predicted risk calibrated. Without it (A3), within-receiver ECE is 1.7 to 4.7 times higher on every dataset (Figure 4a). The shuffled-receiver control stays at the A3 level, so the gain needs the correct receiver, not extra inputs. The per-receiver base rate (A0) is calibrated by construction but cannot rank messages (AUROC 0.500), whereas the full model is both calibrated and discriminative.

Figure 4b shows where the gap comes from. A3 predicts almost the same mean risk for every receiver of a dataset, so it overestimates the risk of the low-risk receivers and underestimates that of Nemotron. The full model tracks the measured level in each of the 30 receiver–dataset cells, with a mean absolute error of 0.54 points against 2.97 for A3. This matters for the decision rule because posterior expected loss compares each message across receivers, so systematically compressing receiver-specific risk levels can distort message selection. The full model learns these levels from privileged supervision: during training, the same messages are measured on every receiver.

Figure 4c separates the two ways in which the prediction is used. Within a receiver, removing the receiver changes AUROC by at most 0.015 on any dataset, so the message features alone rank messages for a fixed receiver. Across receivers, AUROC rises from 0.66–0.75 to 0.76–0.84 on every dataset (Table 10). This across-receiver ranking is what the decision rule needs, since the expected loss $\bar{J}_t(m) = \sum_r \pi_t(r) J_r(m)$ weighs each message across receivers.

Receiver levels or interaction? A3 predicts almost the same risk for every receiver, so part of the gain of the full model could come from receiver-specific base rates alone. We test this by passing A3's predictions through the calibration map of the full model, with a per-receiver slope and intercept and shared terms in the gap g, refitted on the same validation questions (Table 8). This receiver-calibrated A3 recovers 93% of the ECE reduction, 90–93% of the Brier and NLL gains, and

96% of the across-receiver AUROC gain. A per-receiver intercept alone recovers 79% of the ECE reduction, and a richer map fitted on all validation data, which uses more calibration data than the full model, recovers 99%. The full model keeps an advantage within receivers, where recalibration barely changes the ranking: AUROC $+0.004$ [$+0.001$, $+0.008$] and AUPRC $+0.007$ [$+0.002$, $+0.016$]. These two differences also hold against the richer map, whereas its Brier, NLL, and ECE differences from the full model include zero. Over three seeds, the receiver-calibrated A3 again recovers 93% of the ECE reduction and 96% of the across-receiver AUROC gain. Receiver information therefore acts mainly through receiver-specific risk levels, and the message–receiver interaction adds a small gain in ranking messages within a receiver. Both enter the expected loss $\bar{J}_t$.

Table 8: Receiver levels or interaction (Experiment 2, test split, six-dataset means, seed 0). Maps are refitted on the validation questions used by the full model (* on all validation data). Difference: ours minus A3 + recv. calib., with paired question-group bootstrap 95% CIs.

	Per receiver↑		Across receivers↑		Calibration↓		
Method	AUROC	AUPRC	AUROC	AUPRC	Brier	NLL	ECE
A3 (no receiver)	.729	.128	.701	.115	.0414	.168	.032
A3 + recv. intercept	.729	.128	.792	.158	.0401	.156	.015
A3 + recv. calib.	.728	.128	.796	.158	.0399	.155	.012
A3 + recv. map*	.728	.128	.797	.160	.0398	.154	.010
Ours	.732	.135	.800	.164	.0398	.154	.010
Difference	+.004	+.007	+.004	+.006	−.0002	−.0010	−.0015
95% CI	[.001, .008]	[.002, .016]	–	–	[−.0003, −.0001]	[−.0015, −.0006]	[−.0022, −.0001]

Table 9 gives all metrics per dataset, and its AUROC intervals use 1,000 paired bootstrap resamples over frozen question groups, paired across methods and receivers. Table 10 gives the receiver-information ablation. There, pooled AUROC ranks all receivers together, whereas macro AUROC and ECE are computed within each receiver and averaged. Brier and NLL are the same under both views.

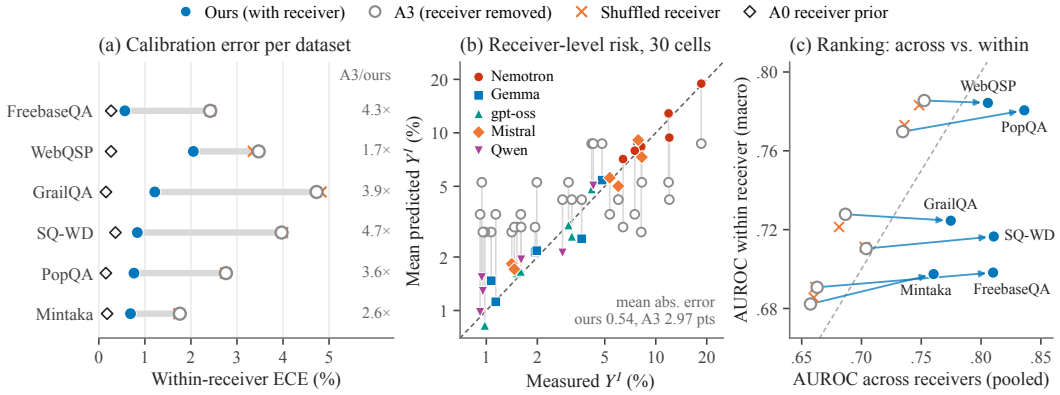


Figure 4: Receiver information calibrates the predicted risk (test split, seed 0 in (a, b), three-seed means in (c)). (a) Within-receiver ECE per dataset. (b) Mean predicted versus measured Y^I in each of the 30 receiver–dataset cells. (c) AUROC across versus within receivers, with arrows from A3 to ours.

Table 9: Experiment 2 results per dataset (test split, seed 0, with metrics computed within each receiver and macro-averaged). AUROC has a paired 95% bootstrap CI. Bold: best value per dataset among learned methods.

Dataset	Method	AUROC [95% CI]	AUPRC	Brier↓	NLL↓	ECE↓
FreebaseQA	A0 global base rate	0.500	0.025	0.0246	0.118	0.024
	A0 per-receiver base rate	0.500	0.025	0.0238	0.105	0.003
	A1 message-only judge	0.529 [0.481, 0.578]	0.029	0.0246	0.118	0.024
	A2 equivalence judge	0.500	0.025	0.0246	0.118	0.024
	A3 receiver-agnostic PIR	0.690 [0.645, 0.730]	0.085	0.0243	0.113	0.024
	A4 action-failure predictor	0.558 [0.518, 0.599]	0.032	0.115	0.377	0.234
	A5 listener / ToM	0.658 [0.614, 0.699]	0.043	0.0241	0.105	0.007
	Shuffled receiver	0.693 [0.649, 0.733]	0.084	0.0243	0.113	0.024
	Receiver-conditioned PIR	**0.696** [0.652, 0.735]	**0.095**	**0.0234**	**0.100**	**0.006**
WebQSP	A0 global base rate	0.500	0.055	0.0523	0.214	0.028
	A0 per-receiver base rate	0.500	0.055	0.0512	0.204	0.003
	A1 message-only judge	0.599 [0.521, 0.673]	0.080	0.0524	0.215	0.034
	A2 equivalence judge	0.500	0.055	0.0525	0.217	0.025
	A3 receiver-agnostic PIR	**0.791** [0.736, 0.829]	0.187	0.0500	0.194	0.035
	A4 action-failure predictor	0.617 [0.541, 0.682]	0.076	0.378	1.128	0.516
	A5 listener / ToM	0.762 [0.704, 0.800]	0.144	0.0513	0.195	0.022
	Shuffled receiver	0.788 [0.736, 0.828]	**0.197**	0.0497	0.193	0.034
	Receiver-conditioned PIR	0.787 [0.731, 0.826]	0.191	**0.0484**	**0.182**	**0.021**
GrailQA	A0 global base rate	0.500	0.079	0.0731	0.277	0.042
	A0 per-receiver base rate	0.500	0.079	0.0701	0.260	0.002
	A1 message-only judge	0.570 [0.540, 0.601]	0.096	0.0730	0.277	0.046
	A2 equivalence judge	0.502 [0.501, 0.504]	0.080	0.0731	0.278	0.045
	A3 receiver-agnostic PIR	0.726 [0.700, 0.750]	**0.198**	0.0701	0.261	0.047
	A4 action-failure predictor	0.473 [0.444, 0.503]	0.072	0.739	2.548	0.810
	A5 listener / ToM	0.705 [0.680, 0.729]	0.145	0.0693	0.250	0.019
	Shuffled receiver	0.725 [0.701, 0.748]	0.188	0.0706	0.262	0.048
	Receiver-conditioned PIR	**0.729** [0.705, 0.753]	0.193	**0.0672**	**0.241**	**0.012**
SQ-WD	A0 global base rate	0.500	0.052	0.0496	0.205	0.039
	A0 per-receiver base rate	0.500	0.052	0.0478	0.188	0.004
	A1 message-only judge	0.552 [0.521, 0.581]	0.063	0.0496	0.206	0.038
	A2 equivalence judge	0.500	0.052	0.0496	0.206	0.038
	A3 receiver-agnostic PIR	0.709 [0.686, 0.734]	0.113	0.0483	0.192	0.040
	A4 action-failure predictor	0.467 [0.437, 0.494]	0.048	0.442	1.292	0.560
	A5 listener / ToM	0.695 [0.672, 0.717]	0.098	0.0476	0.180	0.013
	Shuffled receiver	0.716 [0.693, 0.739]	0.114	0.0484	0.193	0.040
	Receiver-conditioned PIR	**0.718** [0.693, 0.742]	**0.116**	**0.0456**	**0.172**	**0.008**
PopQA	A0 global base rate	0.500	0.033	0.0320	0.145	0.025
	A0 per-receiver base rate	0.500	0.033	0.0313	0.134	0.002
	A1 message-only judge	0.509 [0.465, 0.552]	0.036	0.0319	0.145	0.025
	A2 equivalence judge	0.500	0.033	0.0320	0.145	0.025
	A3 receiver-agnostic PIR	0.775 [0.738, 0.802]	0.116	0.0308	0.133	0.028
	A4 action-failure predictor	0.408 [0.382, 0.435]	0.028	0.617	1.941	0.718
	A5 listener / ToM	**0.787** [0.753, 0.815]	**0.138**	0.0300	**0.118**	**0.006**
	Shuffled receiver	0.780 [0.747, 0.805]	0.121	0.0308	0.132	0.028
	Receiver-conditioned PIR	0.777 [0.743, 0.804]	0.133	**0.0296**	0.119	0.008
Mintaka	A0 global base rate	0.500	0.026	0.0252	0.120	0.016
	A0 per-receiver base rate	0.500	0.026	0.0248	0.114	0.002
	A1 message-only judge	0.554 [0.487, 0.617]	0.036	0.0252	0.121	0.016
	A2 equivalence judge	0.500	0.026	0.0252	0.120	0.016
	A3 receiver-agnostic PIR	0.681 [0.629, 0.729]	0.067	0.0249	0.116	0.018
	A4 action-failure predictor	0.541 [0.482, 0.601]	0.034	0.246	0.687	0.400
	A5 listener / ToM	0.687 [0.641, 0.733]	0.056	0.0249	0.112	0.010
	Shuffled receiver	0.676 [0.625, 0.723]	0.077	0.0249	0.116	0.017
	Receiver-conditioned PIR	**0.688** [0.636, 0.737]	**0.083**	**0.0244**	**0.109**	**0.007**

Table 10: Receiver-information ablation (test split, three-seed means). Δ: full minus receiver-agnostic, with paired 95% CIs in the row below. Last row per dataset: A4 scored on Y^I, with its AUROC on Y^A in parentheses.

Variant	AUROC (pooled)↑	AUROC (macro)↑	Brier↓	NLL↓	ECE (macro)↓
FreebaseQA					
Full model (with e_R)	0.810	0.698	0.0234	0.100	0.005
Receiver removed (A3)	0.663	0.691	0.0243	0.113	0.024
Receiver shuffled	0.661	0.691	0.0243	0.113	0.024
Δ full − A3	+0.147	+0.008	−0.0009	−0.013	
95% CI	[0.124, 0.170]		[−0.0012, −0.0007]	[−0.015, −0.011]	
A4 on Y^I (on Y^A)	0.582 (0.802)	0.561			
WebQSP					
Full model (with e_R)	0.806	0.784	0.0486	0.183	0.021
Receiver removed (A3)	0.752	0.786	0.0500	0.194	0.034
Receiver shuffled	0.748	0.783	0.0501	0.195	0.034
Δ full − A3	+0.053	−0.001	−0.0014	−0.012	
95% CI	[0.028, 0.088]		[−0.0020, −0.0008]	[−0.016, −0.008]	
A4 on Y^I (on Y^A)	0.599 (0.774)	0.602			
GrailQA					
Full model (with e_R)	0.775	0.725	0.0673	0.241	0.014
Receiver removed (A3)	0.687	0.728	0.0700	0.260	0.048
Receiver shuffled	0.681	0.721	0.0706	0.262	0.048
Δ full − A3	+0.088	−0.003	−0.0027	−0.019	
95% CI	[0.073, 0.104]		[−0.0033, −0.0021]	[−0.021, −0.016]	
A4 on Y^I (on Y^A)	0.507 (0.779)	0.469			
SQ-WD					
Full model (with e_R)	0.811	0.717	0.0456	0.172	0.008
Receiver removed (A3)	0.704	0.710	0.0483	0.192	0.040
Receiver shuffled	0.703	0.711	0.0484	0.193	0.039
Δ full − A3	+0.107	+0.006	−0.0027	−0.020	
95% CI	[0.096, 0.118]		[−0.0031, −0.0023]	[−0.022, −0.018]	
A4 on Y^I (on Y^A)	0.469 (0.844)	0.463			
PopQA					
Full model (with e_R)	0.836	0.781	0.0296	0.119	0.007
Receiver removed (A3)	0.734	0.770	0.0309	0.133	0.027
Receiver shuffled	0.736	0.773	0.0309	0.133	0.027
Δ full − A3	+0.102	+0.011	−0.0012	−0.014	
95% CI	[0.086, 0.117]		[−0.0016, −0.0009]	[−0.016, −0.012]	
A4 on Y^I (on Y^A)	0.445 (0.860)	0.388			
Mintaka					
Full model (with e_R)	0.760	0.697	0.0244	0.108	0.006
Receiver removed (A3)	0.657	0.682	0.0249	0.115	0.017
Receiver shuffled	0.660	0.686	0.0248	0.115	0.017
Δ full − A3	+0.103	+0.015	−0.0005	−0.007	
95% CI	[0.078, 0.128]		[−0.0007, −0.0002]	[−0.009, −0.005]	
A4 on Y^I (on Y^A)	0.549 (0.801)	0.538			

I.3 Experiment 3: Receiver-Type Inference

Behavioural evidence about the receiver accumulates with history (Figure 5a,b). Accuracy rises from 20.0% under the uniform prior to 29.0% after twenty responses, and from three responses onward, the calibrated NLL stays below the prior's 1.609, reaching 1.584. The posterior sharpens, but identification stays weak in absolute terms. The shuffled-history control, which pairs each query with the responses of a random receiver, stays near chance (18.4–21.3%) and above the prior's NLL at every length, so the evidence comes from the receiver's own behaviour. After one response the calibrated NLL (1.634) is still above the prior's. Table 11 adds Brier scores (summed over classes), ECE (10 equal-width bins), and the NLL before calibration.

The posterior separates the receivers whose risk differs most (Figure 5c). At $|H_t| = 5$, the one-vs-rest AUROC is 0.62 for Nemotron and 0.60 for Mistral, with 95% CIs above chance, and these are the two riskiest receivers. The three low-risk receivers are not told apart (AUROC 0.45–0.53), but their pooled Y^I lie within 0.6 points of each other, so mistaking one for another barely changes the predicted level of risk. This is the pattern the decision rule needs: by Theorem 4.1, receiver information has value only where it changes the preferred message, which requires receivers whose risk differs. The confusion matrix in Table 12 shows the same pattern: predictions concentrate on Nemotron and Mistral, and Gemma and gpt-oss are almost never predicted. The same table gives the results per dataset.

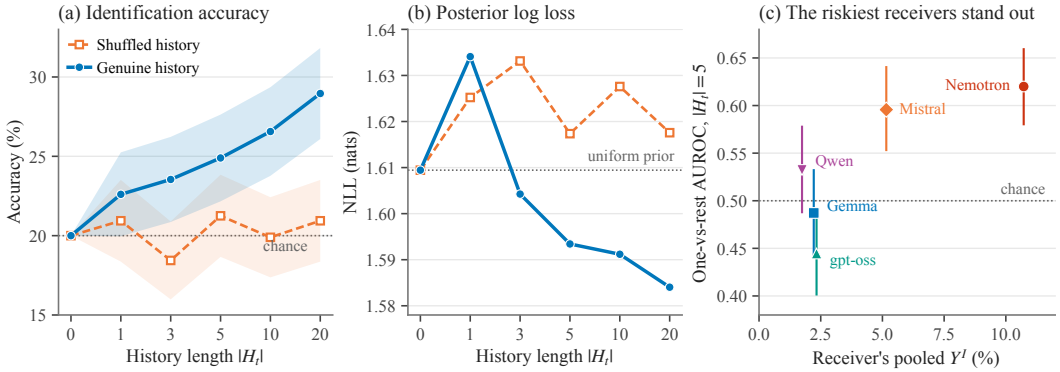


Figure 5: Behavioural history sharpens the receiver belief (960 audit histories per length). (a) Accuracy and (b) calibrated NLL by history length for genuine and shuffled history, with binomial 95% CIs in (a). (c) One-vs-rest AUROC at $|H_t| = 5$ against each receiver's pooled Y^I, with 95% bootstrap CIs.

Table 11: Receiver-type inference by history length $|H_t|$ (960 audit histories per length). Length 0 is the uniform prior over five receivers, and "calibrated" uses temperature scaling fitted on the calibration split.

	Calibrated, genuine history				Raw	Shuffled history	
$\lvert H_t\rvert$	Accuracy↑	NLL↓	Brier↓	ECE↓	NLL↓	Accuracy	NLL
0	20.0%	1.609	0.800	0.000	–	–	–
1	22.6%	1.634	0.811	0.056	1.626	20.9%	1.625
3	23.5%	1.604	0.798	0.013	1.603	18.4%	1.633
5	24.9%	1.593	0.794	0.014	1.594	21.3%	1.617
10	26.6%	1.591	0.793	0.031	1.593	19.9%	1.628
20	29.0%	1.584	0.790	0.063	1.587	20.9%	1.618

Posterior versus MAP receiver. Even after twenty responses, the posterior spreads its mass over several receivers (Table 11). Collapsing it to its MAP receiver would discard this remaining uncertainty. The controller therefore averages over π_t throughout, and the rewrite experiment compares this posterior with a uniform prior and with the true identity.

Table 12: Receiver-type inference at $|H_t| = 5$: by dataset of the calibration tasks (top, 960 histories each) and row-normalised confusion matrix (bottom, %, true receiver in rows).

Dataset	Accuracy↑	NLL↓	Brier↓	ECE↓
FreebaseQA	21.5%	1.604	0.798	0.025
WebQSP	24.6%	1.598	0.796	0.028
GrailQA	20.5%	1.608	0.799	0.040
SQ-WD	18.3%	1.610	0.801	0.051
PopQA	25.9%	1.583	0.790	0.008
Mintaka	25.9%	1.588	0.792	0.017

True \ predicted	Nemotron	Gemma	gpt-oss	Mistral	Qwen
Nemotron	**71.4**	0.0	0.5	18.8	9.4
Gemma	61.5	**0.0**	0.0	31.8	6.8
gpt-oss	50.5	0.0	**0.5**	42.7	6.3
Mistral	47.4	0.5	0.0	**44.3**	7.8
Qwen	53.6	0.0	0.0	38.0	**8.3**

I.4 Experiment 4: Message Revision

Figure 6 shows the paired differences behind Table 2, pooled over datasets and receivers (a) and per dataset (b). Table 13 lists every condition and Table 14 gives the results per receiver. Intervals use 10,000 bootstrap resamples clustered by question. The four pre-registered contrasts are Holm-adjusted, and all other contrasts are unadjusted.

Risk-guided revision beats every reference on every dataset. All four pre-registered contrasts lie well below zero (Figure 6a), and in each of the six datasets ours is below both the original, by 0.81 to 3.81 points, and the generic rewrite, by 0.46 to 3.66 points, with every interval below zero (Figure 6b, post hoc). The secondary contrasts place ours with the other guided conditions: guidance only and the true identity are within 0.03 points of it, one feature per step is 0.20 points lower, and the method before the fix is 0.36 points higher.

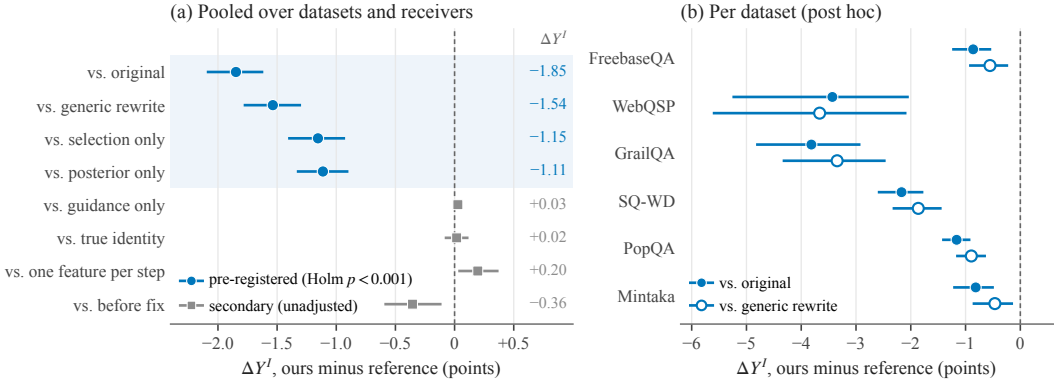


Figure 6: Risk-guided revision lowers measured interpretation failure on the 1,579 held-out questions. Paired differences in Y^I, ours minus each reference (points, 95% CIs, negative favours ours). (b) Per dataset against the original (filled) and the generic rewrite (hollow), post hoc.

Why guidance works. Figure 7 traces the gain to one risk feature that all receivers share. The learned scores differ in size across receivers, from 5.1 points for Nemotron to 0.8 for Qwen, but an unstated output is the top feature for every receiver and passes the guidance rule for all five (panel a). The same instruction, to state the output, therefore lowers risk for every receiver, and receiver information barely changes the preferred message: a uniform belief gives the same Y^I (2.31%, Table 13). This is the case $\Delta_E \approx 0$ of Theorem 4.1, in which receivers differ in how much risk they carry but hardly in which message is best. The framework thus shows that receiver-specific adaptation is not needed here, rather than assuming that it always is. It would be needed where receivers disagree about the best message. The intercepts b_r, the effect of rewriting itself, are small or positive (up to

1.1 points for gpt-oss), so a rewrite helps only when it removes a risky feature. The gain follows each receiver's original risk (panel b): 4.96 points for Nemotron (from 8.73%), 2.62 for Mistral (from 5.27%), and 0.40 to 0.69 for the three low-risk receivers. The generic rewrite, in contrast, raises Y^I for Gemma, gpt-oss, and Qwen by 0.09 to 0.20 points (Table 14).

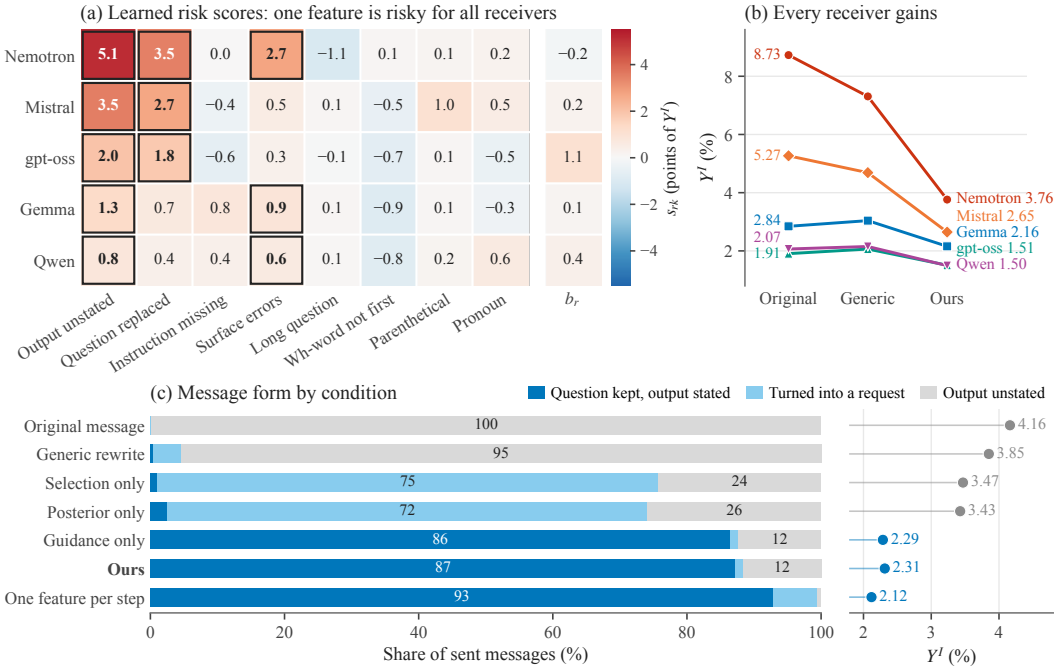


Figure 7: One risk feature is shared by all receivers, and guidance removes it. (a) Learned scores s_{rk} and intercepts b_r (points of Y^I). Boxed: the score and its 2.5% bootstrap quantile are both positive, the guidance rule for a known receiver. (b) Y^I per receiver. (c) Form of all sent messages (unchanged originals included) and Y^I per condition, with guided conditions in blue.

Message form. Figure 7c shows what each condition does to the message. Among changed messages, 95% of generic rewrites leave the output unstated and 84% drop the instruction line. Selection only and posterior only mostly turn the question into a request such as "Name the ..." (83–86% of changed messages), which states the output but replaces the question. The guided conditions keep the question and add a line that states the output (86–93% of all sent messages and 97% of our changed messages), and they have the lowest Y^I in the panel (2.12–2.31%). Guidance thus improves the candidate set itself: selection among generic candidates gains only 0.38 points over the generic rewrite, whereas guided candidates sent without selection gain 1.56.

Table 13: Experiment 4 results by condition. Y^I, Y^A, NONE, and Distr. (probes that choose the contrast task) are percentages on the 1,579 held-out questions, and Changed is the share of messages rewritten. Val. Y^I is on the 300 validation questions of the check before the main run.

Condition	Y^I	Y^A	NONE	Distr.	Changed	Val. Y^I
Original message	4.16	59.33	0.93	3.20	0.0	5.01
Generic rewrite	3.85	58.15	0.88	2.95	97.2	4.97
Selection only	3.47	58.20	0.86	2.59	86.7	4.60
Posterior only	3.43	58.59	0.81	2.60	86.1	4.23
Ours	**2.31**	58.33	0.72	1.56	90.0	2.83
Guidance only	2.29	58.33	0.70	1.54	91.5	2.66
Uniform belief	2.31	58.33	0.72	1.56	89.9	2.83
True identity	2.30	58.58	0.72	1.54	88.0	2.81
One feature per step	2.12	58.75	0.70	1.37	99.4	2.44
Before fix	2.67	59.34	0.74	1.88	87.0	2.69

Further analysis. Four results add detail. First, the gain is smallest on value-versus-attribute questions for Gemma and gpt-oss. There, ours matches the original (+0.07 [−0.60, +0.77] and +0.01 [−0.62, +0.72], Table 14), while the generic rewrite, selection only, and posterior only raise

Table 14: Experiment 4 results per receiver on the held-out questions. Ours minus each reference and the generic rewrite minus the original are paired differences in Y^I (points), and Value is ours minus the original on the 275 value-versus-attribute questions.

	Y^I (%)		Ours minus				Generic minus	
Receiver	Original	Ours	original [95% CI]	generic	sel. only	post. only	original	Value
Nemotron	8.73	3.76	-4.96 [-5.34, -4.59]	-3.54	-2.70	-2.66	-1.42	-5.60
Gemma	2.84	2.16	-0.69 [-1.04, -0.35]	-0.88	-0.55	-0.42	$+0.20$	$+0.07$
gpt-oss	1.91	1.51	-0.40 [-0.62, -0.17]	-0.56	-0.63	-0.52	$+0.16$	$+0.01$
Mistral	5.27	2.65	-2.62 [-3.00, -2.25]	-2.04	-1.50	-1.60	-0.58	-4.09
Qwen	2.07	1.50	-0.57 [-0.85, -0.31]	-0.66	-0.40	-0.36	$+0.09$	-0.38

Y^I by 1.0 to 1.4 points (post hoc). Only one feature per step falls below the original there (-0.77 for Gemma and -0.57 for gpt-oss, the latter not significant). Second, the fix matters. Before the fix, the instruction for an unstated output pointed to the intended task, whose text for value questions uses words such as "kind" and "attribute". The rewriter copied such words into 7.0% of sent messages (40% on value questions), and gpt-oss read them literally and chose the contrast task. The method before the fix has 0.36 points higher Y^I [$+0.11$, $+0.59$] and 1.01 points higher Y^A [$+0.43$, $+1.61$] than ours, and on gpt-oss value questions it raises Y^I by 2.65 points over the original [$+1.18$, $+4.19$]. The fix was designed on earlier development questions, not on the held-out questions. Third, the receiver belief does not change the result. Under every posterior in the cohort, guidance lists the same three features as under a uniform belief (output unstated, question replaced, surface errors), and selection differs in only 2 of 7,895 decisions. With the true identity, four receivers get a different, shorter list (Figure 7a), yet Y^I does not change (-0.02 [-0.12, $+0.08$]). Fourth, the audit supports the measured gain. It judged 12,402 distinct sent messages: 98.1% keep the task and 1.1% add a hint. Flagged messages make up 2.7–3.6% of the unguided conditions and 0.5–0.7% of the guided ones, and sending the original in their place leaves the four pre-registered contrasts at -1.81, -1.50, -1.14, and -1.07.

Validation check and cost. On the 300 validation questions, ours lowered Y^I from 5.01% to 2.83% (-2.18 [-2.95, -1.51]), but its Y^A was not shown to be non-inferior to the original there (-0.47 [-2.07, $+1.07$]) because the interval is wide. The held-out run measured 51,953 distinct message–receiver pairs with 1,870,308 probe calls and 51,953 execution calls. Both stages together used 10.99 A100 hours for the receivers and $11.41 of API calls for generation, verification, and the audit. One feature per step is the most expensive condition: it needs four dependent generate-and-verify rounds instead of one, 2,435 API tokens per question against about 1,220 for ours ($0.53 against about $0.32 for the 1,579 questions), and about four times the latency by its call structure, which we did not time. Its lower Y^I (2.12% vs. 2.31%, unadjusted $+0.20$ [$+0.03$, $+0.37$]) comes from a secondary contrast, so we keep the single-call method as the default.

I.5 Experiment 5: Query Control

Figure 8 shows the query-control results of Table 3 in detail. NetVoII reaches 3.78–3.80% at every quota, below both controls. IG reaches 3.80% only when it queries every episode, and random querying stays within 0.01 points of no query (panel a). All eight paired differences exclude zero (panel b). At a matched quota the query costs cancel, so each net-utility contrast equals the Y^I contrast with its sign reversed, and all eight survive Holm correction ($p \leq 0.012$). NetVoII does not identify the receiver better than IG (panel c), so its gain does not come from identification. Intervals use 2,000 bootstrap resamples clustered by question, and random querying is always the mean of three seeds. Table 15 gives the budget grids for both cohorts, at exact quotas of 10, 20, 30, and 50% of episodes and for querying every episode with the IG or random choice of query (Always). Table 16 gives the cost sweep, and Table 17 and Figure 10 give the per-dataset and per-receiver levels.

Information is not value. Figure 9 shows why the two criteria diverge. The most informative query carries almost the same expected information in every episode, 0.19 nats (range 0.186–0.193), yet the share of episodes with positive predicted VoII ranges from 1.4% on WebQSP and GrailQA to 73–75% on PopQA and Mintaka (panel a). Theorem 4.2 explains the gap: it bounds VoII by G_t, the largest loss reduction that the query's information can buy, and G_t is zero when one message is best

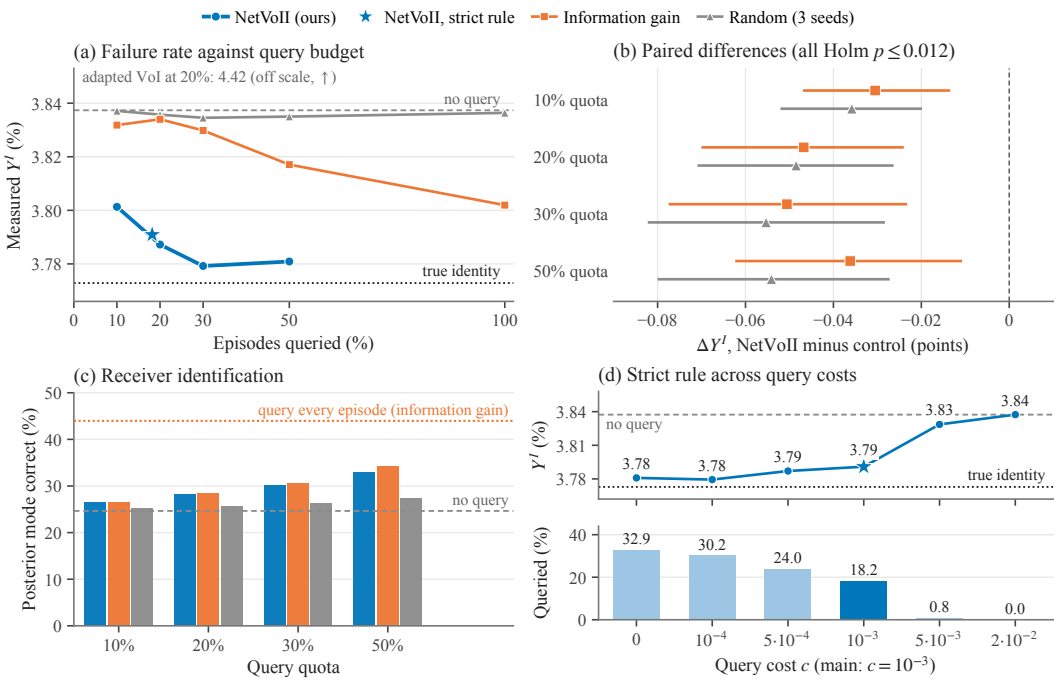


Figure 8: NetVoII allocates queries by decision value (primary cohort, interpretation objective, $c = 0.001$). (a) Y^I against the share of episodes queried. (b) Paired differences at matched quotas with 95% CIs, equal to the net-utility contrasts with the sign reversed. (c) Share of episodes whose posterior mode is the true receiver. (d) Strict rule at six query costs c: measured Y^I (top) and share of episodes queried (bottom).

for every receiver type, however informative the query. IG cannot see this difference, since every episode offers about the same information. NetVoII spends its queries where the answer can change the message.

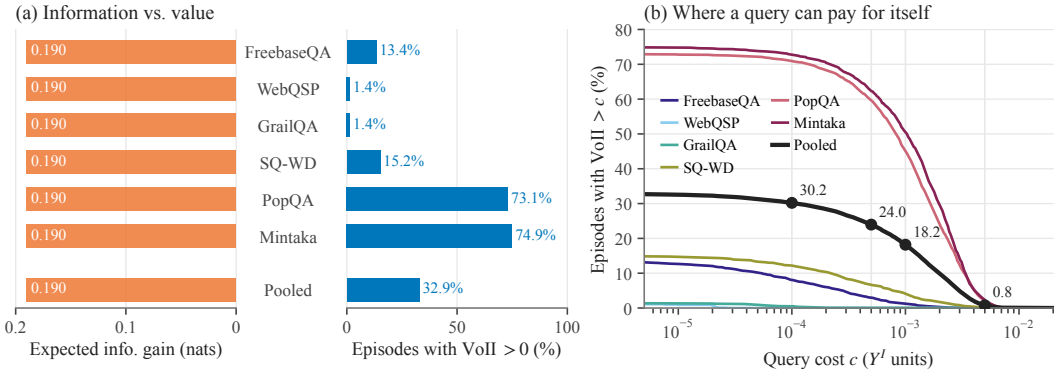


Figure 9: Information is not value (primary cohort). (a) Expected IG of the most informative query (nats) and share of episodes with positive predicted VoII, per dataset. (b) Share of episodes whose best query is worth more than a cost c, with pooled values marked at four of the tested costs.

Sparsity of predicted value. Predicted value is sparse and small. Only 32.9% of episodes have positive predicted VoII, the median positive value is 0.0012, close to the main cost $c = 0.001$, and 0.8% of episodes exceed 0.005 (Figure 9b). Because NetVoII ranks episodes by predicted value, returns fall as the quota grows: of the episodes added from 30% to 50%, 86% have zero predicted value, and the added queries change Y^I by $+0.002$ points ($[-0.005, +0.008]$). The strict rule follows the same curve. At each of six frozen costs c it queries exactly the episodes whose predicted VoII exceeds c (Figure 8d, Table 16): as c rises from 0 to 0.02, the query rate falls from 32.9% to zero

and Y^I moves from 3.78% back to 3.84%, so the controller adjusts smoothly to the cost rather than switching off. A higher cost drops the queries with the lowest predicted value first, and these barely matter: raising c from 0 to 10^{-4} lowers the query rate by 2.7 points and leaves Y^I at 3.78%.

Scope of the gain. Three results set the scope of the gain. First, its size is set by how much decision-relevant receiver information is left after five stored responses. The posterior is still broad (its mode is correct in 24.6% of episodes), but little of this uncertainty can change the message under the current candidate set: in half of the questions (50.4%) one candidate is best for every receiver, so no answer can change the choice. On a base of 3.84%, the true identity would lower Y^I by only 0.06 points, and NetVoII lowers it by about 0.05. Second, the gain is in interpretation, the target VoII optimises. On the primary cohort, task success is 35.7% for every policy but the adapted VoI rule (35.3%), and all paired differences between NetVoII and the controls lie within 0.1 points of zero. On the replication cohort, NetVoII's task success is 0.2 to 0.3 points below both controls at matched budgets (unadjusted intervals exclude zero), and the strict rule lowers it by 0.23 points relative to no query ($[-0.40, -0.05]$). Under a combined objective that also scores execution, $Y^I + (1 - Y^I)Y^A$, none of the eight contrasts survives correction (Holm $p = 1$). Third, the replication is directional but not significant: NetVoII is at or below both controls at every budget, but every interval includes zero, and the strict rule's net utility there is slightly below no query.

Adapted VoI comparator. The adapted VoI rule shares the posterior, query bank, candidates, verifier decisions, and argmin with NetVoII. Only the risk estimates differ, coming from DeepSeek V4 Flash instead of the learned heads (122 of 9,029 estimate items fell back to training base rates). These estimates are coarse: five values (0, 0.01, 0.02, 0.05, and 0.10) make up 92% of them. As a result, one candidate is best, or tied for best, for every receiver in every question, against 50% (primary) and 55% (replication) of questions for the heads, so its VoII is zero everywhere and the strict rule never queries. The minimum expected loss is tied in 94.5% of episodes (92.5% in the replication), and since ties go to the earlier candidate under every policy, the rule sends the original in 81% of episodes, against 18% (primary) and 21% (replication) for the heads without a query. Tied episodes account for 0.56 of its 0.58-point gap to the heads (0.30 of 0.34 in the replication). In the 5.5% of episodes without a tie, its messages give 4.91% against 4.44% for the heads. A random tie-break would narrow the gap: on the tied episodes whose tied candidates were all measured (5.9% of tied episodes), a random tied candidate gives 3.86%, against 3.99% for the original and 3.62% for the heads' choice (replication, 7.4%: 3.39%, 3.65%, and 3.55%), but these subsets are small and selected by what was sent. The comparison therefore tests the LLM utility estimates inside the same VoI machinery, not the VoI rule itself, and we treat it as a secondary comparator.

Per-dataset allocation. Because the quota is pooled, NetVoII spends it where predicted value is high (Figure 10a). At 20% it queries about half of the PopQA and Mintaka episodes and none of WebQSP or GrailQA (Table 17). At 50% it queries about 80% of PopQA and Mintaka, and only at this quota does it query more than 0.4% of WebQSP or GrailQA. The gain appears where the queries go (Figure 10b): at 20%, NetVoII lowers Y^I by 0.19 points on PopQA and 0.04 on Mintaka and stays within 0.02 points of no query elsewhere, while IG and random querying move Y^I by at most 0.03 points on any dataset. The true identity in Table 17 is a diagnostic, not a lower bound: it can sit above no query where the learned risk heads are imperfectly calibrated for a receiver (e.g., WebQSP, 5.14% vs. 5.07%) and above NetVoII on PopQA (2.52% vs. 2.42%) and for Gemma and Mistral.

Table 15: Experiment 5 budget grid (interpretation objective): measured Y^I (%) by exact query quota, with net utility at $c = 0.001$ in parentheses. Bold: net-utility contrasts against both IG and random querying survive Holm correction. † Secondary comparator: its estimates never let a query change the message.

Policy	10%	20%	30%	50%	Always
Primary cohort (test split). No query: 3.84 (0.9616), true identity: 3.77 (0.9623), adapted VoI† without queries: 4.42 (0.9558).					
Random (3-seed mean)	3.84 (0.9615)	3.84 (0.9614)	3.83 (0.9614)	3.83 (0.9612)	3.84 (0.9606)
IG	3.83 (0.9616)	3.83 (0.9615)	3.83 (0.9614)	3.82 (0.9613)	3.80 (0.9610)
Adapted VoI†	4.42 (0.9557)	4.42 (0.9556)	4.42 (0.9555)	4.42 (0.9553)	–
NetVoII (ours)	**3.80 (0.9619)**	**3.79 (0.9619)**	**3.78 (0.9619)**	**3.78 (0.9617)**	–
Replication cohort. No query: 3.93 (0.9607), true identity: 3.91 (0.9609), adapted VoI† without queries: 4.27 (0.9573).					
Random (3-seed mean)	3.93 (0.9606)	3.93 (0.9605)	3.93 (0.9604)	3.93 (0.9602)	3.92 (0.9598)
IG	3.94 (0.9605)	3.94 (0.9604)	3.94 (0.9603)	3.96 (0.9599)	3.96 (0.9594)
Adapted VoI†	4.27 (0.9572)	4.27 (0.9571)	4.27 (0.9570)	4.27 (0.9568)	–
NetVoII (ours)	3.92 (0.9607)	3.91 (0.9607)	3.92 (0.9605)	3.92 (0.9603)	–

Table 16: Strict rule of Experiment 5 across query costs c (interpretation objective, net utility at each row's cost). The last two rows of each block re-run IG and random querying at the query count of the $c = 0.001$ rule.

Cost c	Queried (%)	Y^I (%)	$1 - Y^A$ (%)	Net utility
Primary cohort (test split)				
0	32.9	3.78	35.66	0.9622
0.0001	30.2	3.78	35.68	0.9622
0.0005	24.0	3.79	35.62	0.9620
0.001 (main)	18.2	3.79	35.68	0.9619
0.005	0.8	3.83	35.72	0.9617
0.02	0.0	3.84	35.73	0.9616
IG at the $c = 0.001$ count	18.2	3.83	35.72	0.9615
Random at the $c = 0.001$ count	18.2	3.84	35.72	0.9615
Replication cohort				
0	31.2	3.92	36.67	0.9608
0.0001	28.8	3.92	36.65	0.9608
0.0005	23.4	3.92	36.66	0.9607
0.001 (main)	17.8	3.92	36.68	0.9606
0.005	0.8	3.94	36.88	0.9606
0.02	0.0	3.93	36.90	0.9607
IG at the $c = 0.001$ count	17.8	3.94	36.90	0.9604
Random at the $c = 0.001$ count	17.8	3.93	36.91	0.9605

Table 17: Experiment 5 results per dataset and per receiver (primary cohort, interpretation objective): Y^I (%) at the pooled 20% quota and under the strict rule, with the share of episodes that NetVoII queries.

		Y^I at the 20% quota			Queried by NetVoII (%)			
Unit	No query	NetVoII	Inf. gain	Random	20% quota	Strict	Y^I strict	True identity
FreebaseQA	2.16	2.16	2.15	2.16	1.5	1.1	2.16	2.10
WebQSP	5.07	5.07	5.07	5.07	0.0	0.0	5.07	5.14
GrailQA	6.96	6.96	6.96	6.96	0.0	0.0	6.96	6.96
SQ-WD	4.53	4.51	4.50	4.53	4.9	4.1	4.51	4.45
PopQA	2.61	2.42	2.62	2.60	49.8	45.1	2.42	2.52
Mintaka	2.49	2.45	2.49	2.49	54.5	50.5	2.46	2.34
Nemotron	8.53	8.43	8.54	8.53	19.9	17.9	8.43	8.31
Gemma	2.39	2.34	2.37	2.39	19.9	18.2	2.34	2.37
gpt-oss	2.40	2.40	2.40	2.40	19.9	18.3	2.40	2.39
Mistral	4.16	4.06	4.16	4.15	20.1	18.4	4.07	4.12
Qwen	1.72	1.72	1.70	1.71	20.1	18.2	1.71	1.68

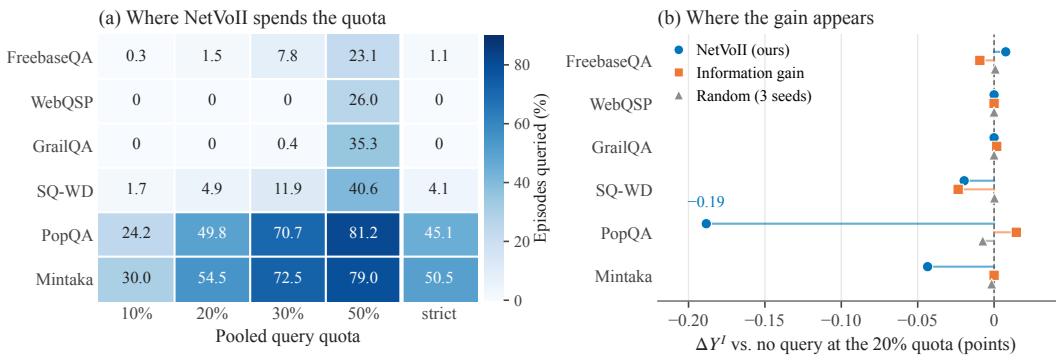


Figure 10: NetVoII spends a pooled quota where predicted value is high (primary cohort, interpretation objective). (a) Share of each dataset's episodes queried at pooled quotas of 10–50% and under the strict rule. (b) Change in Y^I against no query at the 20% quota.

J Pseudocode and Full Experimental Specification

Algorithms 1 and 2 are the two decisions made at deployment (Section 3), and Algorithm 3 is the offline stage that fits their learned parts. Where a line implements an equation of the main text, its comment cites it. Section J.3 states how Experiments 4 and 5 instantiate these steps, and Section J.4 lists every setting.

J.1 End-to-End Deployment Pseudocode

At deployment the receiver identity is hidden, and the sender knows the receiver only through the behavioural history H_t. Algorithm 2 calls Algorithm 1 once. The action set $\mathcal{A}_t$ it returns stays fixed through the query decision, so a query can change which candidate is sent but never adds a new one. Rewrite is one request to the rewrite model for K_m candidates that keep the exact task while reducing the features in G_t. Verify is a separate stateless call to the same model version. It receives only its arguments and never sees π_t, risk scores, feature values, or which candidate is preferred. In the experiments, the probes and the execution call run only after the message is sent, and their labels are used only for evaluation.

Algorithm 1 Posterior-guided message revision

Input: intended task $z^\star$, context x_S, initial message m_0, receiver posterior π_t, heads q^I_ϕ, q^C_ϕ and risk effects s_{rk} from Algorithm 3, loss terms c_M, L_I, L_C, candidate budget K_m, and guidance size n_g

Output: fixed valid action set $\mathcal{A}_t$ and revised message $\widehat{m}^\star_t$

function Revise($z^\star, x_S, m_0, \pi_t$)
  $\bar{s}_{tk} \leftarrow \sum_r \pi_t(r)\, s_{rk}$ for every mutable feature k ▷ posterior-weighted risk effect
  $G_t \leftarrow$ the at most n_g mutable features with the largest positive $\bar{s}_{tk}$
  $(m'_1, \ldots, m'_{K_m}) \leftarrow$ Rewrite$(m_0, z^\star, x_S, G_t)$ ▷ fixed budget, no regeneration
  $(v_1, \ldots, v_{K_m}) \leftarrow$ Verify$(z^\star, x_S, m_0, m'_1, \ldots, m'_{K_m})$ ▷ one stateless batched call
  $\mathcal{A}_t \leftarrow \{m_0\} \cup \{m'_j : v_j = \text{Pass}\}$ ▷ the original is always kept
  for each $m \in \mathcal{A}_t$ and each $r \in \mathcal{R}$ **do**
    $J_r(m) \leftarrow c_M(m) + L_I\, q^I_r(m) + L_C\,\big(1 - q^I_r(m)\big)\, q^C_r(m)$ ▷ Eq. (6)
  $\bar{J}_t(m) \leftarrow \sum_r \pi_t(r)\, J_r(m)$ for every $m \in \mathcal{A}_t$ ▷ Eq. (7)
  return $\mathcal{A}_t$ and $\widehat{m}^\star_t = \arg\min_{m \in \mathcal{A}_t} \bar{J}_t(m)$ ▷ posterior, not MAP receiver

J.2 Training Pseudocode

Offline, every message of the training and validation splits is sent to every frozen receiver, which gives the privileged labels of Section 2 (lines 1–3 of Algorithm 3). All learned parts are fitted on training data, selected and calibrated on validation data, and frozen before any headline evaluation. Line 4 gives the two losses of Section 3: the heads are separate networks, and the capability loss

Algorithm 2 Deployment: revision, VoII-gated receiver query, and send

Input: intended task $z^\star$, context x_S, initial message m_0, behavioural history H_t, query bank $\mathcal{Q}$ with likelihoods $\widehat{P}(y \mid R = r, \xi)$ and costs c_ξ, and frozen outputs of Algorithm 3
Output: the sent message $\widehat{m}^\star$

$\pi_t \leftarrow \pi_\omega(\cdot \mid H_t)$ ▷ calibrated posterior, the prior if $H_t = \emptyset$
$(\mathcal{A}_t, \widehat{m}_t^\star) \leftarrow \textsc{Revise}(z^\star, x_S, m_0, \pi_t)$ ▷ Algorithm 1
$V_0 \leftarrow \min_{m \in \mathcal{A}_t} \sum_r \pi_t(r)\, J_r(m)$ ▷ value without a query
for each query $\xi \in \mathcal{Q}$ **do** ▷ the bank may be shortlisted by $\mathrm{IG}(\xi)$
  for each response $y \in \mathcal{Y}_\xi$ **do**
    $\widehat{P}(y \mid H_t, \xi) \leftarrow \sum_r \pi_t(r)\, \widehat{P}(y \mid R = r, \xi)$ ▷ predictive distribution
    $\pi_{t+1}^{\xi,y}(r) \leftarrow \widehat{P}(y \mid R = r, \xi)\, \pi_t(r) \,/\, \widehat{P}(y \mid H_t, \xi)$ for every r ▷ Eq. (8)
  $V_\xi \leftarrow \sum_y \widehat{P}(y \mid H_t, \xi)\, \min_{m \in \mathcal{A}_t} \sum_r \pi_{t+1}^{\xi,y}(r)\, J_r(m)$ ▷ same fixed $\mathcal{A}_t$, Eq. (9)
  $\mathrm{NetVoII}(\xi) \leftarrow V_0 - V_\xi - c_\xi$ ▷ Eq. (10)
$\xi^\star \leftarrow \arg\max_{\xi \in \mathcal{Q}} \mathrm{NetVoII}(\xi)$
if $\mathrm{NetVoII}(\xi^\star) > 0$ **then** ▷ ask only if the query pays for itself
  ask $\xi^\star$, observe the response y, and set $\pi_{t+1} \leftarrow \pi_{t+1}^{\xi^\star,y}$
  $\widehat{m}^\star \leftarrow \arg\min_{m \in \mathcal{A}_t} \sum_r \pi_{t+1}(r)\, J_r(m)$ ▷ re-score the same $\mathcal{A}_t$
else
  $\widehat{m}^\star \leftarrow \widehat{m}_t^\star$
send $\widehat{m}^\star$ to the receiver
return $\widehat{m}^\star$

weights each pair by $1 - \widetilde{y}^I$, the share of probes read correctly, a soft form of the condition $Y^I = 0$ in Equation (4).

Algorithm 3 Offline training with privileged receiver supervision

Input: training and validation messages m (each with intended task $z^\star$, context x_S, and contrast tasks), frozen receivers $\mathcal{R} = \{R_1, \ldots, R_K\}$, calibration bank $\mathcal{Q}$ disjoint from the test tasks, predefined mutable features $k = 1, \ldots, d$, and probe repeats K_p
Output: calibrated heads $q_r^I(m) = q_\phi^I(m, z^\star, x_S, e_r)$ and $q_r^C(m) = q_\phi^C(m, z^\star, x_S, e_r)$, type model π_ω, likelihoods $\widehat{P}(y \mid R = r, \xi)$, and learned risk effects s_{rk}

Stage 1: privileged labels (offline only)
for each message m and each receiver $r \in \mathcal{R}$ **do**
  $\widetilde{y}_r^I(m) \leftarrow$ the mean error over parsed probe responses ▷ Eq. (5)
  $y_r^A(m) \leftarrow \mathbf{1}\{\mathrm{score}(A_r, \theta) < \delta\}$ ▷ separate execution rollout, Eq. (2)
Stage 2: receiver-conditioned risk heads
$\mathcal{L}_r^I(m) \leftarrow \mathrm{BCE}\big(q_r^I(m), \widetilde{y}_r^I(m)\big), \quad \mathcal{L}_r^C(m) \leftarrow \big(1 - \widetilde{y}_r^I(m)\big)\, \mathrm{BCE}\big(q_r^C(m), y_r^A(m)\big)$
fit q_ϕ^I and q_ϕ^C as separate networks minimising $\sum_{m,r} \mathcal{L}_r^I(m)$ and $\sum_{m,r} \mathcal{L}_r^C(m)$ on the training split ▷ configuration selected by validation NLL
calibrate q_ϕ^I and q_ϕ^C on natural-prevalence validation data ▷ per-receiver Platt map
Stage 3: query likelihoods and receiver-type model
for each query $\xi \in \mathcal{Q}$ and each receiver $r \in \mathcal{R}$ **do**
  $\widehat{P}(y \mid R = r, \xi) \leftarrow$ Laplace-smoothed frequency of y in the stored responses of r to ξ
build histories $H = \{(\xi_s, y_s)\}$, each from the stored responses of one receiver
$\omega \leftarrow \arg\min_\omega \sum_{(H,r)} -\log \pi_\omega(r \mid H)$, then calibrate π_ω on held-out histories ▷ Section E
Stage 4: learned risk effects of predefined mutable features
for each receiver $r \in \mathcal{R}$ and each mutable feature k **do**
  $s_{rk} \leftarrow$ training-set risk effect of feature k on the interpretation failure of r ▷ Section F
freeze ϕ, ω, $\widehat{P}$, $\{s_{rk}\}$, all prompts, and all thresholds before headline evaluation
return $q_\phi^I,\ q_\phi^C,\ \pi_\omega,\ \widehat{P},\ \{s_{rk}\}$

J.3 Instantiation in Experiments 4 and 5

Experiment 4 runs Algorithm 1 without a query, with the posterior of Experiment 3 after five stored responses. Its risk scorer is the linear model of Section F, which predicts the change in soft Y^I from the original to a rewrite m as $\sum_r \pi_t(r)\,(b_r + \sum_k s_{rk}\Delta f_k(m))$. In line 10 the original is kept unless this predicted drop exceeds 0.001, and in line 3 G_t holds at most $n_g = 3$ features, each with a positive 2.5% bootstrap quantile of $\bar{s}_{tk}$. Deterministic checks run before VERIFY. The other conditions change single steps: the generic rewrite and selection only set $G_t = \emptyset$, the generic rewrite and guidance only send the first valid candidate instead of the $\arg\min$, posterior only passes π_t to REWRITE in place of G_t, the uniform-belief and true-identity diagnostics replace π_t, and one feature per step runs REWRITE four times with $n_g = 1$, each time on the current message, before selection.

Experiment 5 runs Algorithm 2 with $G_t = \emptyset$ (generic rewrites), frozen verifier decisions, and the frozen paired heads of Section F in place of q^I_ϕ. Its bank holds the 192 fit-role tasks of Experiment 3, a response y is binary execution success, and $L_C = 0$ except under the combined objective, which adds the frozen q^C_ϕ. The strict rule is line 11 of Algorithm 2 with $c_\xi = 0.001$. A budget of $B\%$ instead queries the $B\%$ of episodes with the largest $\mathrm{NetVoII}(\xi^\star)$. The IG control ranks queries and episodes by $\mathrm{IG}(\xi)$ instead, random querying ranks them at random, and under the strict rule both controls are matched to its query count.

J.4 Full Experimental Details for Reproducibility

Datasets. Six KBQA collections in their public releases: FreebaseQA, WebQSP, GrailQA, SQ-WD, PopQA, and Mintaka. Questions are normalised and grouped, and the groups are split 70/15/15 into train/validation/test. Test questions per dataset: 1,310 / 219 / 1,286 / 1,418 / 1,451 / 945 (6,629 in total), giving 33,145 message–receiver pairs. Each question yields one message from the PromptSource template `qa_template_basic`, one intended task, and one contrast task from the fixed contrast set (entity-versus-category or value-versus-attribute). No synthetic enrichment is used at any stage.

Models. Receivers: Nemotron-3 Nano 30B, Gemma 4 26B, gpt-oss-20b, Mistral Small 3.2, and Qwen3.6 35B, all served frozen through vLLM at temperature 0 with default system prompts and no tools. Probe generation, probe verification, and the A1/A2 judges use DeepSeek V4 Flash. Predictor features come from Qwen3-Embedding-8B (4,096 dimensions).

Interpretation probe. Three options per probe (intended task, contrast task, NONE OF THESE) and six differently worded instances per message–receiver pair, each a separate temperature-0 call. Probe text is generated per message, checked deterministically (option distinctness, no answer leakage), and verified by a second call. Parsed responses that choose the contrast task or NONE OF THESE count as wrong, and unparsed responses are dropped from the denominator. Execution is one separate answer call scored by gold-alias containment.

Receiver histories and query bank. Calibration bank of 384 tasks disjoint from the test messages, answered by all five receivers (9,600 archived responses). Histories are sampled from the archive at lengths 1, 3, 5, 10, and 20 (20,800 in total), and length 0 is the uniform prior. Tasks are split by role into 192 fit / 48 early-stop / 48 calibration / 96 audit, and every reported number uses 960 audit histories per length. The shuffled-history control pairs each query with responses from a randomly chosen receiver.

Training details. PIR heads: two separate MLPs (q^I and q^C) with hidden widths in $\{[256, 64], [64, 16], [512, 128]\}$, dropout 0.3–0.7, and a 16-dimensional receiver embedding. Inputs are the embeddings of m and x_S and optionally their difference, chosen per dataset. The MLP output is added to a fixed logistic offset fitted first on the training data (per-receiver intercept and slope on the gap g, one intercept and slope for A3). 27 configurations times 3 seeds per dataset, selected on validation NLL only, followed by a Platt map with per-receiver slope and intercept and shared terms in g (one global map for A3), fitted on the calibration part of the validation split. q^C is trained on Y^A with example weights $1 - \widetilde{y}^I$. Receiver-type model: PCA to 32 dimensions, an interaction MLP between query and response embeddings, mean pooling over the history, and a

five-way softmax, with 3 seeds times 2 regularisation strengths, early-stopped on the early-stop role split and temperature-calibrated on the calibration role split.

Rewrite module. Number of generated candidates $K_m = 4$ per request, with no regeneration. In Experiment 4 the rewriter (DeepSeek V4 Flash, temperature 0.4, eight requests of one condition per call) receives the guidance of its condition, the verifier is one stateless batched call, and selection uses the linear risk model with no further LLM call. In Experiment 5 the pool is four generic rewrites per message, verified with the frozen v1.1 rules (acceptance 92.8% of candidates) and scored by the paired heads, which add one embedding call per candidate. Both models are described in Section F.

Query module. Query bank: the 192 fit-role tasks of Experiment 3 with their archived responses from all five receivers. Signature: binary execution success against the known bank answer. Likelihoods: Laplace-smoothed per receiver and task. Update: one exact Bayes step from the frozen posterior of Experiment 3 after five stored responses, with at most one query per episode. Candidate pool: the original message and its four generic rewrites with the frozen verifier decisions (rejected slots map to the original). Scores: the six frozen paired heads of Section F, and the frozen q^C for the combined objective, which was not refit on rewrites. Budgets are exact quotas over pre-response scores within each cohort, random schedules use three pre-declared seeds, and the strict rule uses six frozen costs. The adapted VoI rule (Dong et al., 2026) enumerates the same finite posterior with DeepSeek V4 Flash estimates of candidate and receiver utilities (2,260 calls, with 122 of 9,029 estimate items falling back to training base rates under a pre-declared rule). Ties in expected loss go to the earlier candidate (the original first), as for every policy (Section I.5). All schedules were frozen before any receiver call. Experiment 5 then issued 3,159,599 receiver calls (642,210 queries, 359,627 executions, and 2,157,762 probes over the 135,435 episodes of both cohorts).

Baselines and ablations. A0–A5 as defined in Table 6, plus the shuffled-receiver control and the receiver-calibrated A3 (A3 predictions passed through the full model's calibration map, refitted on the same validation questions). A1 and A2 are DeepSeek V4 Flash judge calls with a fixed rubric. Of 26,528 calls, 54 returned unparsable output and fall back to the training-mean score. A4 shares the architecture, inputs, and selection protocol of the full model but is trained on Y^A. A5 is a listener classifier over the three probe options with the same inputs as the full model. Rewrite ablations follow Section H and are reported in Section 6, and query controls are reported in Section 6 and Tables 15 to 17.

Evaluation metrics. Prediction: AUROC, AUPRC, Brier, NLL, and ECE (10 equal-mass bins), computed within each receiver and macro-averaged in the main text, and pooled over receivers in Table 10. Receiver inference: accuracy, NLL, Brier (summed over classes), and ECE (10 equal-width bins). Rewrite: measured Y^I and Y^A per condition, change rate, and verifier acceptance, with paired question-cluster bootstrap intervals. Query control: share of episodes queried, measured Y^I and $1 - Y^A$, net utility $1 - Y^I - c\,r$ at the stated cost, the combined loss $Y^I + (1 - Y^I)Y^A$, and the share of posteriors whose mode is the true receiver, with paired question-cluster bootstrap intervals.

Statistical reporting. Main-text values of Experiment 2 use seed 0 with 1,000 paired bootstrap resamples over frozen question groups, paired across methods and receivers. Three-seed means appear in Table 10. Intervals are binomial with $n = 960$ in Experiment 3 and question-level normal approximations in Experiment 1. In Experiment 4 they are 10,000 paired bootstrap resamples over the 1,579 held-out questions, with Holm correction over the four pre-registered comparisons. Per-receiver, per-dataset, and secondary contrasts are unadjusted. In Experiment 5 they are 2,000 paired bootstrap resamples over question clusters that keep all five receivers and three repeats together, with Holm correction over the eight pre-registered net-utility contrasts on the primary cohort (NetVoII against IG and against random querying at four budgets). The replication cohort, the combined objective, the strict-rule comparison with no query, and per-dataset contrasts are unadjusted. Model selection, calibration, and thresholds use validation data only.

K Implementation and Reproducibility Checklist

For each receiver–message rollout, log: run ID, code commit, timestamp, seed, dataset/split/task-family/item identifiers, intended-task and distractor IDs, sender/receiver model versions, receiver-

history IDs, current posterior π_t, raw message and template ID, complete probe options and order, raw/parsed probe response, Y^I, raw execution response and Y^A, raw and calibrated head outputs, query-bank scores and response likelihoods, entropy reduction, gross/net VoII, rewrite candidates, semantic-verifier decisions/reasons, final message, token counts, latency, and monetary/API cost where available.

Raw model outputs are stored separately from parsed labels so that parsers and scoring rules can be rerun without repeating inference. No private chain-of-thought is requested or treated as ground truth.

Statistical reporting. For each headline table, report the point estimate and a paired 95% bootstrap CI over question groups, paired across methods and receivers. Report macro receiver averages together with pooled results. Thresholds, calibration, cost parameters, and model-selection decisions are fixed using validation data only.

Natural prevalence. All splits are frozen at natural prevalence, and the reported experiments use no oversampling and no synthetic examples. If enrichment is used in later training, the predictor must be recalibrated on the untouched natural validation set before Brier/NLL/ECE or probability-based decision rules are evaluated.

L Discussion and Limitations

This section interprets the results of Sections 6 and I in the light of Section 4, states the limits of the evidence, and outlines extensions.

L.1 What the Results Show

The premises of PIR hold. The five experiments support the framework step by step. Receivers read the same messages differently: pooled Y^I ranges from 1.7% to 10.7%, the best and worst receiver differ by 4–13$\times$ within each dataset, and the lowest-risk receiver changes across datasets (Experiment 1). Interpretation failure is its own target: 32% of misread pairs still pass the answer check (Experiment 1), and a predictor trained on execution failure is near chance on Y^I (Experiment 2). The risk can be predicted before sending (AUROC 0.732 within receivers), and receiver information cuts ECE from 0.032 to 0.010, mostly by learning each receiver's risk level (Experiment 2). A short behavioural history sharpens the posterior over receiver types without naming the receiver (Experiment 3). On these premises, risk-guided revision lowers measured Y^I from 4.16% to 2.31% while execution failure stays within a 1-point non-inferiority margin of every reference (Experiment 4). At a 20% query budget, NetVoII closes 78% of the gap between no query and the true identity, against 5% for IG and 3% for random querying (Experiment 5).

When receiver information pays off. Heterogeneity alone does not make receiver information useful, as Theorem 4.1 states. Receivers differ several-fold in how often they misread, yet in Experiment 4 the receiver belief barely matters: a uniform belief changes 2 of 7,895 selection decisions, and the true identity gives almost the same Y^I (2.30% vs. 2.31%). The learned risk scores show why. An unstated output is the top risk feature for all five receivers, so the best message hardly depends on the receiver and $\Delta_E \approx 0$. Receiver information pays off where it changes a decision. In Experiment 2 it corrects the risk level of each receiver and, when one message is ranked across receivers, lifts AUROC from 0.701 to 0.800. A receiver-specific recalibration of A3 recovers most of both gains, and much of the ranking gain reflects Nemotron's position effect (Sections D.3 and I.2). In Experiment 5 it decides which episodes deserve a query.

Why IG is not enough. Experiment 5 shows the gap between information and value that Theorem 4.2 describes. The most informative query carries about the same information in every episode (0.19 nats), yet predicted value is positive in only 32.9% of episodes, from 1.4% on WebQSP and GrailQA to 73–75% on PopQA and Mintaka. IG identifies the receiver at least as well as NetVoII: at a 20% budget, its posterior mode is the true receiver in 28.5% of episodes, against 28.3% for NetVoII. Yet its answers rarely change the preferred message, so Y^I barely moves. A query has value only when its answer can move the posterior into the decision region of another message, and NetVoII spends its budget on the episodes where it predicts such a move.

Implications for LLM systems. Three lessons follow for systems in which models pass messages to one another. First, interpretation should be scored separately from execution: an answer-based check counts 32% of misreads as successes. Second, most of the revision gain comes from one fix shared by all receivers, stating the expected output, which the learned risk scores rank first for every receiver. A sender can apply such a fix before it knows who will read the message, and receiver-specific adaptation is needed only where receivers disagree about the best message. Third, the sender should keep its uncertainty about the receiver and query only when the expected gain exceeds the cost. Predicted value is concentrated in a minority of episodes, and querying every episode gives lower net utility than never querying (Table 3). In large agent networks, querying by default can also flood the system with queries.

L.2 Limitations

Measurement. The probe measures interpretation through behaviour, not through internal representations: it records which of three options the receiver picks, the intended task, one contrast task, or NONE OF THESE. A misreading other than the offered contrast can register only as NONE OF THESE, so each message tests one direction of misreading. Option order can bias the pick. Nemotron tends to avoid the last listed option: it misreads 24.5% of probes that list the intended task last, against 3.8% of the other probes. The gap appears under each of the six probe wordings, so it is an effect of position. Receiver heterogeneity survives when only probes with the intended task first or second are kept (3.8× between the highest and lowest receivers, Section D.3), but most of Nemotron's excess Y^I, and much of the across-receiver ranking gain of Experiment 2, reflect this effect. Execution is scored by gold-alias containment, which can miss paraphrased answers and accept answers that only mention the gold entity. Finally, misreadings are rare, about 4.4% of test pairs, so estimates for a single receiver, dataset, or question type have wide intervals.

Setting. The evidence comes from a narrow setting. All messages are knowledge-base questions rendered by one fixed template, so no sender model writes them, and each carries one of two contrast types (entity versus category, value versus attribute). Real handoffs between agents are longer, carry intermediate results, and can be misread in more ways. The receivers are five open-weight models run at temperature 0, and the type model is closed-set: a receiver from a new family would be assigned to one of the five known types. Identification is also weak in absolute terms. After twenty stored responses, top-1 accuracy is 29.0% against 20.0% for the uniform prior, so the posterior stays broad (Section I.3).

Control. Revision uses one rewriter, DeepSeek V4 Flash, and the semantic verifier is the same model version, called separately and without any optimisation signal. An audit of 12,402 sent messages found that 98.1% keep the task and 1.1% add a hint (Section I.4). The candidate set is finite, so a safer equivalent message can be missed. Our evaluation therefore concerns selection among the generated, verified candidates rather than global prompt optimality. Queries come from a fixed bank of 192 tasks, with at most one query per episode and no open-ended dialogue. The query gain is small in absolute terms: the true identity would lower Y^I by 0.06 points, and NetVoII lowers it by about 0.05. The gain is in interpretation rather than task success, and on the replication cohort its intervals include zero (Section I.5). The adapted VoI rule is a weak comparator: its LLM estimates are too coarse to let a query change the message, so it tests these estimates rather than the VoI rule. Section G.6 lists the studies that lie outside the reported experiments.

L.3 Future Directions

Three extensions follow from these limits. Open-set receiver modelling would detect a receiver that matches none of the known types instead of forcing it into one. Multi-round interaction would let the sender query, revise, and query again, and longer communication chains would show how interpretation errors spread across hops. Richer tasks, such as tool calls, code, and multi-step plans, would test whether PIR and VoII carry over when the set of possible readings is open-ended.