EARS: Explanatory Abstention for Reliable Sub-Agent Modeling in Large-Scale Multi-Agent Systems

Shuang Xie [*1], **Yunan Lu** [*†1,2], **Han Li** [*1], **Lingyun Wang** [1]
[1]Shopify [2]Columbia University
Correspondence: lingyun.wang@shopify.com

Abstract

In large-scale enterprise settings, centralized multi-agent systems (MAS) are increasingly adopted, in which a coordinator delegates user requests to lightweight, domain-specialized sub-agents. While this architecture improves modularity, scalability, and cost efficiency, its reliability depends not only on accurate routing but also on sub-agents' ability to calibrate their responses to capability constraints. In particular, sub-agents built on smaller fine-tuned models often struggle with such calibration, leading them to over-answer ambiguous, underspecified, misrouted, or unsupported requests and produce hallucinated outputs instead of actionable feedback. To address this challenge, we present *EARS* (Explanatory Abstention for Reliable Sub-Agent Modeling), a production-oriented framework that reframes sub-agent abstention as an inter-agent communication protocol: a sub-agent does not merely abstain, but exposes an actionable failure state to the coordinator. *EARS* curates human-agent interaction data using an ensemble of calibrated LLM-as-a-Judge models, producing structured abstention labels and rationales under a taxonomy of sub-agent failure modes. These data are used to fine-tune sub-agents to detect failure conditions and return rationales for coordinator-level clarification, rerouting, or fallback. We evaluate *EARS* in a large-scale production e-commerce assistant supporting enterprise business intelligence workflows. *EARS* improves the overall response pass rate from 68.5% to 78.9%, demonstrating that sub-agent-side explanatory abstention improves MAS reliability.

1 Introduction

Multi-Agent Systems (MAS) have demonstrated impressive capabilities across complex tasks, including software engineering, scientific discovery, data analytics, web automation, and general-purpose problem solving (Qian et al., 2024; Wang et al., 2025; Gottweis et al., 2025; Swanson et al., 2024; Fourney et al., 2024). In industry, a widely adopted design is the centralized MAS architecture (Zhang et al., 2026; Wu et al., 2024; Fourney et al., 2024), in which a powerful coordinator interprets user requests, routes them to specialized sub-agents, and integrates their outputs into a final response. To balance task performance and operational cost, sub-agents often use smaller, domain-specific models fine-tuned for particular domains or databases (Abdin et al., 2024; Lepagnol et al., 2024). However, this architecture introduces reliability challenges: the system can fail not only when the coordinator misroutes a request, but also when the sub-agent returns unfaithful execution results.

Existing work has predominantly addressed MAS reliability at the coordinator layer by improving task planning, agent routing, and collaboration strategies (Yue et al., 2025; Wu et al., 2026; Zhang et al., 2025; Dang et al., 2025; Jin et al., 2025). Although these methods mitigate routing and coordination errors, they leave a complementary failure mode underexplored: even a correctly selected sub-agent may over-answer when a request is ambiguous or lacks required information. In these cases, an executable-looking output can propagate an error to the coordinator instead of revealing the condition that prevented faithful execution.

Recent analyses of MAS failures suggest that sub-agent execution and communication are central to system reliability. The failure taxonomy of Cemri et al. (2025) shows that many system-level failures arise from breakdowns in inter-agent information flow, including failures to seek clarification, information withholding, and ignoring inputs from other agents. These findings highlight that what a sub-agent returns to the coordinator is not merely an intermediate payload; it is a communication signal for downstream recovery. Prior work has studied when models should refuse unanswerable

[*] denotes equal contribution.
[†] This work was done as part of internship at Shopify.

The Proceedings of the Second Workshop for REsearch on Agent Language Models (REALM 2026), pages 832–845
October 29, 2026 ©2026 Association for Computational Linguistics

or unsupported questions (Kirichenko et al., 2025; Amayuelas et al., 2024). However, existing methods primarily study single-agent systems, where abstention serves as a final user-facing refusal, rather than a communication signal to the coordinator.

To address this gap, we propose *EARS*, a domain-adaptive approach for improving the robustness of centralized MAS. *EARS* uses calibrated LLM-as-a-Judge models with unanimous judge agreement to curate abstention supervision from human-agent interaction data. It then fine-tunes sub-agents to return category-specific abstention decisions and rationales as actionable coordinator signals, improving response reliability.

We make three contributions. First, we formulate explanatory abstention at the sub-agent level as a coordination signal for production MAS, turning abstention from a user-facing refusal into actionable inter-agent feedback. Second, we introduce a data curation pipeline to iteratively identify interaction states where sub-agents should abstain. Third, we evaluate *EARS* on a large-scale e-commerce platform that serves millions of customers globally and supports substantial annual commerce volume. *EARS* improves the response pass rate by 15.2% relatively, demonstrating that explanatory abstention provides a practical mechanism for improving reliability in large-scale centralized MAS.

2 Related Work

2.1 Centralized Multi-Agent Systems

Centralized MAS coordinate multiple specialized agents through a controller that decomposes requests, assigns tasks, and aggregates intermediate outputs. Frameworks such as AutoGen (Wu et al., 2024), Magentic-One (Fourney et al., 2024), and AgentOrchestra (Zhang et al., 2026) provide general mechanisms for organizing agent roles and communication workflows. Building on these architectures, recent work has improved how coordinators select and manage specialized agents. RIRS selects specialists according to their knowledge boundaries and combines routing with planning for complex cross-domain queries (Wu et al., 2026). MasRouter jointly determines the collaboration mode, agent roles, and underlying LLMs to balance task performance and inference cost (Yue et al., 2025). Other approaches learn routing policies or centralized control strategies under performance-cost trade-offs (Zhang et al., 2025; Dang et al., 2025; Jin et al., 2025). These methods primarily improve decisions made before or during task delegation. However, accurate routing does not guarantee faithful execution after a task reaches a sub-agent. A selected sub-agent may lack necessary context, tools, schema coverage, or domain capabilities while still producing a plausible output. *EARS* addresses this complementary problem at the sub-agent–coordinator interface. Rather than changing how the coordinator selects a sub-agent, it trains the selected sub-agent to expose its execution boundary through a structured and actionable response. Consequently, *EARS* can complement existing routing and orchestration methods.

2.2 Abstention in LLMs

Abstention enables LLMs to avoid unsupported responses when available knowledge or evidence is insufficient. One line of work evaluates whether LLMs abstain appropriately under knowledge uncertainty, unanswerable questions, incomplete evidence, ambiguity, and unsupported requests (Wen et al., 2025; Brahman et al., 2024; Amayuelas et al., 2024; Madhusudhan et al., 2024). AbstentionBench broadens this evaluation across diverse forms of unanswerability and shows that even strong reasoning models frequently answer when they should abstain (Kirichenko et al., 2025). RefusalBench constructs diagnostic cases by applying controlled linguistic perturbations to answerable questions, enabling systematic evaluation across different sources and degrees of informational uncertainty (Muhamed et al., 2025). ConfuseBench specifically focuses on cases involving uncertainty and reveals that LLMs struggle to accurately identify its root cause (Liu et al., 2025). These benchmarks provide controlled and reproducible evaluation signals, but they typically define answerability under a fixed task context. Moreover, synthetically perturbed queries may not fully reflect naturally occurring user requests or the system-specific constraints that make them unanswerable in practice.

Beyond evaluation, recent methods also improve abstention through model training or inference-time intervention. Learning to Clarify introduces action-based contrastive self-training to teach LLMs appropriate dialogue policies, particularly when to ask clarifying questions rather than guess the user's intent (Chen et al., 2025). Abstain-R1 uses reinforcement learning with verifiable rewards to jointly optimize abstention and follow-up clarification for underspecified requests (Zhai et al., 2026). LatentRefusal formulates unan-

swerable Text-to-SQL requests as an answerability-gating problem and detects them from intermediate model representations (Ren et al., 2026). These approaches improve whether and how an individual model declines a request. Even when they generate clarification, however, they generally formulate recovery as an interaction between the model and the end user. *EARS* studies a different system role: the abstaining model is a specialized sub-agent, and its response becomes an internal coordination signal. The signal must therefore identify not only that execution should stop, but also why it should stop and what action the coordinator can take next.

3 Method

We introduce *EARS*, a framework for enhancing MAS resilience by enabling sub-agents to abstain and provide refusal rationales to the coordinator. As illustrated in Figure 1, *EARS* consists of two stages. The **offline calibration stage** builds a human-annotated seed dataset with an abstention taxonomy (Section 3.1), and uses it to calibrate an LLM-as-a-Judge via iterative refinement (Section 3.2). The **online data flywheel stage** applies the calibrated judges to curate interaction data (Section 3.3) for sub-agent fine-tuning (Section 3.4). The updated sub-agent is then deployed back into the MAS to collect new interactions and drive the next cycle, enabling iterative improvement.

3.1 Human Annotation

Abstention Taxonomy. To make sub-agent abstention feedback useful for the coordinator's final response, we design a taxonomy that captures why the selected sub-agent cannot handle a delegated request. Building on prior work (Wen et al., 2025), we classify abstention cases into four mutually exclusive categories: *Ambiguous Query* (Asai and Choi, 2021), where the user intent cannot be clearly determined; *Insufficient Input* (Li et al., 2024), where the intent is clear but execution requires additional information; *Missing Capability* (Amayuelas et al., 2024), where the sub-agent does not support the requested functionality; and *Misrouting*, where the request should be handled by another sub-agent. This taxonomy provides fine-grained failure categories that sub-agents return to the coordinator. We provide examples of each category in Table 4.

Annotation Process. To minimize annotation ambiguity and guide annotators through logically dependent label decisions, we structure the annotation process as a hierarchical decision procedure. Annotators first determine whether the user query is ambiguous. If the query admits multiple materially different interpretations due to underspecified key information, annotators label it as *Ambiguous Query*. Otherwise, the query is treated as non-ambiguous, meaning that it has a single intended operation. For non-ambiguous queries, annotators then assess whether the target sub-agent can fulfill the request given its available context, supported capabilities, and business policy constraints. If the sub-agent can complete the request, the instance is marked as solvable. If not, annotators assign it to one of the abstention categories: *Insufficient Input*, *Missing Capability*, or *Misrouting*.

Each annotation object includes both a category label and a written rationale. The rationale specifies the source of ambiguity for *Ambiguous Query*, the missing information for *Insufficient Input*, the unsupported functionality for *Missing Capability*, or the appropriate sub-agent for *Misrouting*.

Following standard annotation practice (Lia et al., 2025; Artstein and Poesio, 2008), we adopt an independent dual-annotation scheme with adjudication. Each instance is labeled independently by two annotators. Agreed labels are accepted directly; disagreements are resolved by a third senior annotator. Annotation was conducted by a professional data annotation vendor. The process yielded 600 solvable queries and 602 abstention instances, suggesting that a substantial proportion of user queries are not correctly handled by the sub-agent.

3.2 LLM-as-a-Judge Calibration

To curate abstention-aware training data from production data at scale, we use an automatic judge to determine whether a query is solvable and, if not, which abstention category it belongs to. Directly applying an off-the-shelf LLM-as-a-judge is unreliable in this setting because abstention decisions depend on task-specific rules and subtle distinctions among abstention categories. We therefore calibrate the judge on the human-annotated seed set before applying it to large-scale data.

Given the human-annotated seed set, we iteratively optimize the LLM-as-a-judge instructions to improve abstention-category prediction accuracy. We begin with an initial general instruction prompt and run the judge on the annotated dataset to measure its performance. If prediction accuracy falls below a predefined threshold, e.g., 80%, we use

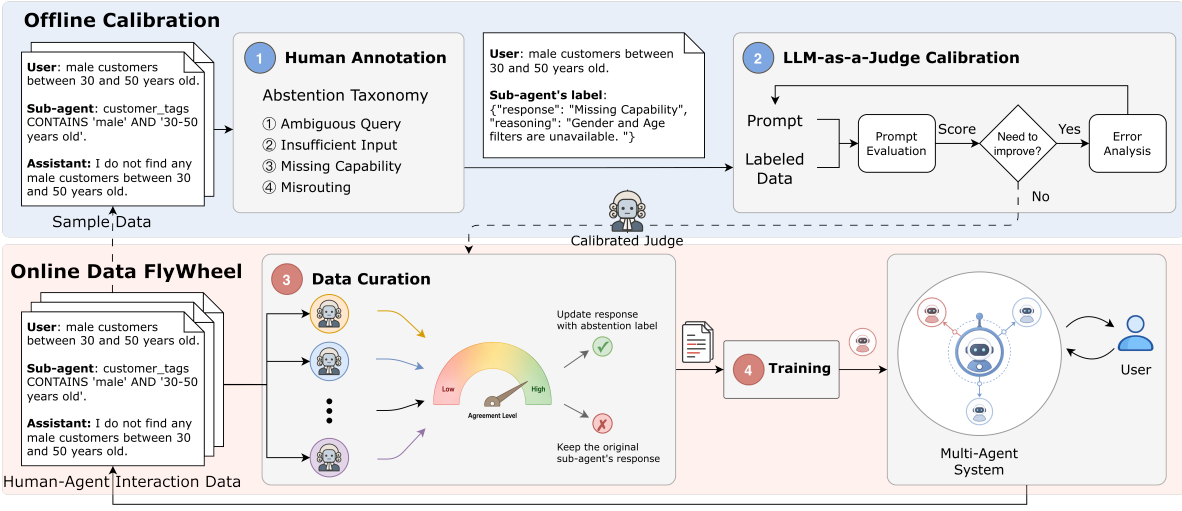

Figure 1: Overview of *EARS*. The framework starts by sampling a subset of human-agent interaction data for human annotation, then uses the annotated seed set to calibrate LLM-as-a-Judge models under an abstention taxonomy. The calibrated judges then curate large-scale interaction data for sub-agent fine-tuning. The updated sub-agent is finally redeployed into the multi-agent system, forming an online data flywheel for iterative improvement.

a second LLM to analyze the failure cases, summarize prominent confusion patterns, and generate actionable suggestions for revising the guidelines. This establishes a human-in-the-loop revision process that mitigates the risk of overfitting associated with fully automated prompt optimization. It also ensures that guideline updates remain general, interpretable, and consistent with the annotation taxonomy. We repeat this process until the score exceeds the threshold and shows no improvement for two consecutive revisions.

3.3 Data Curation

We apply the calibrated LLM-as-a-judge to annotate the human-agent interaction data. Prior work shows that different LLMs can exhibit complementary strengths, motivating ensemble-based approaches that combine predictions across models (Verga et al., 2024). Following this insight, we instantiate judges from three model families: `OpenAI` (OpenAI, 2025), `Claude` (Anthropic, 2025), and `Gemini` (Google, 2025). To avoid incorrectly converting answerable queries into abstention cases, we adopt a conservative ensemble strategy. For each sample, we run all judges and accept the label only when they unanimously agree on the same abstention category.

The number of required agreeing judges provides tunable precision–recall control. In our setting, we use unanimous agreement to construct a high-precision curated dataset for training. In applications where recall is more important, such as surfacing potential abstention cases for human review, this threshold can be lowered according to deployment requirements and risk tolerance.

3.4 Training

After curating abstention-aware interaction data, we fine-tune the sub-agent with supervised fine-tuning (SFT). Each training instance consists of the user query, the sub-agent context, and the expected sub-agent response: a task-specific output for solvable queries, or a structured abstention response with an abstention category and a concise rationale. We jointly train on solvable and abstention examples so that the sub-agent learns both to execute supported requests and to recognize its execution boundary, rather than defaulting to either over-answering or excessive abstention.

Subsequently, the sub-agents are deployed back into the MAS to serve live interactions. Newly collected interactions are subsequently returned to the data curation stage, where the calibrated judges identify additional abstention cases for future training cycles, completing the flywheel loop.

4 Experiments and Results

4.1 Experimental Setup

We evaluate *EARS* on an enterprise-level AI assistant deployed in a production e-commerce environment, focusing on its ability to answer merchant queries related to shop operations and management. The assistant follows a centralized coordinator–sub-

agent architecture, where a powerful coordinator built on `Claude-sonnet-4-6` (Anthropic, 2026) routes user queries to a set of in-house fine-tuned sub-agents built on `Qwen3-32B-Instruct` (Qwen Team, 2025). In this work, we focus on two sub-agents in the *business intelligence (BI)* domain: the **customer segmentation sub-agent**, which constructs customer cohorts based on merchant-specified conditions (e.g., Find customers who visited my shop in the past week but did not make a purchase), and the **business analytics sub-agent**, which answers questions about shop performance and operational metrics by generating responses in a Domain-Specific Language (DSL) with SQL-like syntax (e.g., What was my shop's conversion rate last month?).

Dataset. The training dataset consists of 13.8K user-agent interactions randomly sampled from the production environment, including 7.0K customer segmentation queries and 6.8K business analytics queries. We then apply the data curation workflow described in Section 3.3 to identify unsolvable queries and their corresponding abstention categories: *Ambiguous Query*, *Insufficient Input*, *Missing Capability*, and *Misrouting*. For evaluation, we construct a balanced dataset covering both customer segmentation and business analytics queries. Table 1 summarizes the statistics of the training and evaluation datasets.

Category	Training Set	Eval Set
Segmentation Solvable	5,967	1,524
Analytics Solvable	6,800	2,000
Abstention	1,086	476
Total	13,853	4,000

Table 1: Dataset statistics for training and evaluation.

Training Setup. We fine-tune the sub-agents, built upon `Qwen3-32B-Instruct` (Qwen Team, 2025), using supervised fine-tuning (SFT). As the task requires learning nuanced abstention boundaries across multiple categories, we perform full-parameter fine-tuning. We adopt AdamW with a cosine learning rate schedule, setting the peak learning rate to 1×10^{-5} with a warmup ratio of 5%. We train for three epochs with a global batch size of 8 and a maximum sequence length of 32,768 with data packing. All experiments are conducted on 8 NVIDIA H200 GPUs.

4.2 Evaluation Metrics

Abstention Ability Following prior work (Feng et al., 2024), we evaluate the LLM-as-a-Judge's quality in identifying abstention categories using standard metrics: accuracy, precision, and recall.

Cost We measure computational cost by token usage and normalize it relative to a single `gemini-2.5-flash` LLM-as-a-Judge call.

Pass Rate We measure sub-agent performance by Pass Rate, defined as the percentage of queries that receive correct responses. The **Overall Pass Rate** is determined through two sequential verification steps: (1) **Syntax Validity (Syntax)**, where a deterministic, code-based verifier checks whether the structured output is valid and executable when applicable, including either a syntactically correct query in the corresponding sub-agent's Domain-Specific Language (DSL) or a properly formatted JSON abstention response; and (2) **Semantic Correctness (Semantic)**, where an LLM-as-a-Judge assesses whether the output satisfies the user query along four dimensions: intent alignment, domain constraint satisfaction, query logic correctness, and abstention correctness when applicable. Outputs that fail the syntax check are automatically marked as overall failures and are not passed to the semantic judge. A response passes overall only if it satisfies both verification steps. We use the same semantic judge across all methods; Appendix A provides the prompt details.

Because Syntax Validity is determined by a fully reproducible code-based verifier, human agreement is relevant only to the Semantic Correctness judge. To validate this judge, we asked four domain experts with extensive knowledge of the deployed sub-agents to annotate 1,050 production samples using the same rubric. Two experts independently labeled each sample, achieving substantial inter-annotator agreement (Cohen's $\kappa = 0.718$), and resolved disagreements through adjudication. Against the resulting consensus labels, the semantic judge achieved 88.3% accuracy, 82.4% precision, and 77.8% recall.

Automated Conversation Evaluation (ACE) While Pass Rate evaluates sub-agent outputs, we additionally use ACE, our in-house LLM-as-a-Judge system, to assess whether these improvements translate into better final response in the MAS. ACE evaluates each response on a 0-2 scale along five dimensions: (1) Claim Groundedness,

LLM-as-a-Judge	Accuracy	Precision	Recall	Cost
`gpt-5`	88.5%	89.2%	88.5%	x4.1
`o3`	89.6%	90.3%	89.5%	x6
`claude-sonnet-4-5`	88.4%	88.4%	88.4%	x9.3
`gemini-2.5-flash`	<u>90.4%</u>	<u>90.5%</u>	**90.4%**	**x1**
4-judge consensus	**93.1%**	**96.2%**	<u>89.8%</u>	x20.4

Table 2: Abstention classification performance of calibrated LLM-as-a-Judge models. The best result is in **bold** and the second-best is <u>underlined</u>.

Metric	Baseline	Prompt-Only	*EARS*
Overall Pass Rate	68.5%	70.5%	**78.9%**
Seg. Syntax	92.7%	93.5%	**97.2%**
Seg. Semantic	61.9%	65.8%	**82.7%**
Analytics Syntax	96.3%	96.2%	**96.9%**
Analytics Semantic	**75.1%**	**75.1%**	75.0%

Table 3: Evaluation results for the customer segmentation and business analytics sub-agents. Seg. and Analytics denote customer segmentation and business analytics, respectively. Syntax and Semantic denote syntax validity and semantic correctness pass rates. Prompt-Only provides the abstention taxonomy in context without abstention-aware fine-tuning.

whether domain-specific claims are supported by information available to the model at the corresponding turn; (2) Goal Fulfillment, whether the system demonstrably completes the user's objective; (3) User Satisfaction, inferred exclusively from the user's utterances; (4) Action Execution Success, whether attempted actions complete successfully, independent of whether the system selects the appropriate actions; (5) Response Usability, whether responses are relevant, comprehensible, and concise.

To validate ACE as an end-to-end evaluator, we compare its scores against a held-out set of 311 conversations independently annotated by domain experts using the same 0–2 rubric. Averaged across the five evaluation dimensions, ACE achieves 82.7% accuracy, a quadratic weighted Cohen's κ of 0.42, and a Kendall's τ of 0.43. The rank correlations are statistically significant for all dimensions ($p < 10^{-3}$), supporting the use of ACE to measure end-to-end system quality. Table 5 reports the dimension-level results.

4.3 Evaluation Results

Table 2 reports the calibrated abstention classification performance and relative cost of different LLM-as-a-Judge models on 1,202 human-annotated samples. All four single-model judges achieve above 88% accuracy, precision, and recall. Among them, `gemini-2.5-flash` provides the strongest cost-performance trade-off, achieving 90.4% accuracy at the lowest cost. The 4-judge consensus further improves precision by 6.3% at higher cost. These results show a practical trade-off between efficiency and accuracy: single-judge curation is preferable when throughput is the primary constraint, while multi-judge consensus provides higher-confidence abstention labels when label quality is the priority. Appendix C provides a analysis for each abstention category.

Table 3 reports the response quality of the sub-agents. The baseline corresponds to the MAS with sub-agents trained without abstention data. The prompt-only variant uses the same models and provides the abstention taxonomy in context without additional fine-tuning, whereas *EARS* fine-tunes the sub-agents on data curated by an ensemble of four LLM-as-a-Judge models: `gpt-5`, `o3`, `claude-sonnet-4-5`, and `gemini-2.5-flash`. Prompt-only abstention increases the Overall Pass Rate from 68.5% to 70.5%, while *EARS* further improves it to 78.9%, outperforming the prompt-only variant by 8.4 percentage points (11.9% relatively). The gains primarily arise from customer segmentation sub-agent, for which semantic correctness increases from 61.9% to 82.7% relative to the baseline. This asymmetry is expected because 23.8% of queries from customer segmentation sub-agent are predicted to require abstention, compared with fewer than 2% of analytics queries. Syntax validity remains high across both sub-agents and all variants, suggesting that *EARS* improves abstention decisions without degrading DSL query generation. Besides, *EARS* improves the overall ACE score by 2.97%, indicating that explanatory abstention improves not only sub-agent correctness but also final system behavior.

Human Evaluation. To support side-by-side comparison with the incumbent system, we conducted an end-to-end shadow deployment with human validation. We randomly sampled 2,000 queries from all production sessions routed to either BI-domain sub-agent in June 2026. Third-party domain experts evaluated each example at two granularities. Each example received two independent labels at each granularity, with disagreements adjudicated by a third expert.

At the end-to-end MAS level, experts assessed

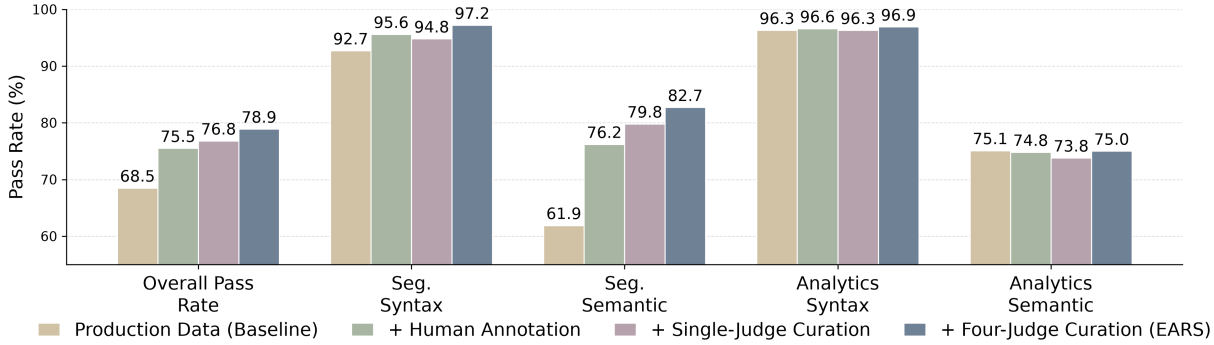

Figure 2: Ablation results on the effect of abstention data under different curation strategies. Adding human-annotated abstention data improves the overall pass rate and substantially increases segmentation semantic correctness. *EARS* uses four-judge curation, which achieves the strongest overall performance.

session-level response quality using a binary rubric, with 1 denoting success and 0 denoting failure. *EARS* achieved a human-rated success rate of 84%, substantially outperforming the incumbent system's 71.5%. At the sub-agent level, its abstention decisions achieved 93% accuracy, 93% precision, 70% recall, and 80% F1, demonstrating strong alignment with expert judgments. Appendix B provides the complete evaluation rubrics.

4.4 Ablations

To assess the effect of abstention data and LLM-as-a-Judge ensemble strategy, we run ablations across dataset variants with different curation settings. The results are summarized in Figure 2. Overall, abstention data consistently improves segmentation performance without degrading analytics performance. Human-annotated data increases the overall pass rate by 10.2% relatively. The gain is most pronounced in segmentation semantic correctness, with a 23.1% relative improvement, demonstrating that abstention examples can improve model reliability.

Although human annotation provides high-quality abstention supervision, mixing it with raw production interactions introduces label inconsistencies. Semantically similar queries may receive conflicting supervision: one is labeled with an abstention category, while another retains the original incorrect sub-agent response. *EARS* curation reduces this conflict by correcting production examples, yielding further gains.

Among single-judge variants, the best-performing judge (`gemini-2.5-flash`) improves the overall pass rate by 1.3% over human annotation alone. In contrast, four-judge consensus achieves a 4.5% gain and improves all four metrics. This pattern suggests that single-judge curation can introduce label noise, whereas consensus-based curation provides a more reliable training signal by retaining only high-agreement corrections.

4.5 Cost analysis

We further compare the deployment costs of *EARS* and inference-time gating using the calibrated judges. For 10 million queries, inference-time gating incurs a recurring monthly cost of approximately \$90K for a single `gemini-2.5-flash` judge at \$0.009 per query, or \$1.83M for the four-judge consensus configuration. In contrast, *EARS* incurs a one-time offline cost of \$3,293, comprising \$2,525 for curating 13.8K training examples and \$768 for 24 hours of training on eight H200 GPUs. The one-time cost of *EARS* is equivalent to processing approximately 366K queries with the single-judge gate and only 18K queries with the four-judge gate. Therefore, although inference-time gating may remain economical at low query volumes, offline fine-tuning substantially reduces costs for a production system serving millions of queries each month.

5 Case Study

To better understand how abstention training affects deployed MAS behavior, we conduct a qualitative analysis over representative cases. Table 4 compares the baseline with *EARS* across four abstention categories. The examples show that the baseline sub-agent often forces underspecified, unsupported, or misrouted requests into executable segmentation predicates. This behavior propagates to the assistant, which returns a hallucinated or misleading response to the user. In contrast, *EARS* produces category-specific abstention rationales that

Type	User request	Baseline	EARS
Ambiguous Query	"find customers who might be interested in buying shoes wholesale"	**Sub-agent:** `storefront.product_viewed` `MATCHES (tag contains 'wholesale')` **Assistant:** "**I do not find any customers** who might be interested in buying shoes wholesale..." **Issue:** Uses an unverified proxy for wholesale buyer interest instead of asking for clarification.	**Sub-agent:** `{"Response":"Ambiguous Query", "Reasoning": "What is wholesale-interest signal?"}` **Assistant:** "**The segmentation tool needs a clearer signal for wholesale interest.** Could you specify what indicates potential wholesale buyers, such as repeat shoe purchases, large orders, or views of wholesale products?"
Insufficient Input	"customers who purchased products with type French Horns, Trombones, or Trumpets"	**Sub-agent:** `product_purchased MATCHES (tag contains 'Frentch Horns' OR contains 'Trombines' OR 'Trumpets')` **Assistant:** "**I do not find any customers** who purchased products with these types." **Issue:** Attempts segmentation without resolved product inputs, leading to invalid product filters.	**Sub-agent:** `{"Response": "Insufficient Input", "Reasoning": "Need intended product."}` **Assistant:** "Could you clarify which products you mean by French Horns, Trombones, or Trumpets? **I need the intended product** before creating the segment."
Missing Capability	"male customers between 30 and 50 years old"	**Sub-agent:** `customer_tags CONTAINS 'male' AND customer_tags CONTAINS '30-50 years old'` **Assistant:** "**I do not find any male customers between 30 and 50 years old.**" **Issue:** Maps unsupported demographic attributes to tags, falsely implying searchable gender and age fields.	**Sub-agent:** `{"Response": "Missing Capability", "Reasoning": "Gender and Age filters are unavailable."}` **Assistant:** "**I cannot create a segment directly from gender or age because these filters are unavailable.** I can help segment customers using supported criteria, such as purchase behavior, location, customer tags, or engagement."
Misrouting	"create segment for customers who visited the store multiple times but never purchased"	**Sub-agent:** `number_of_orders = 0` **Assistant:** "Here are the customer segments for **customers who never purchased**: ..." **Issue:** Misroutes workflow-dependent request to segmentation, dropping repeated-visit condition.	**Sub-agent:** `{"Response": "Misrouting", "Reasoning": "Need workflow sub-agent."}` **Assistant:** "I routed this request to the workflow sub-agent. **It will first tag customers who visit multiple times without purchasing**; then I will create a segment from that tag."

Table 4: Representative abstention examples comparing the baseline and *EARS*. The baseline over-answers ambiguous, underspecified, unsupported, or misrouted requests, whereas *EARS* abstains with category-specific rationales that support more accurate assistant responses. **Red** marks errors, and **green** marks accurate outputs.

help the assistant choose a better next action.

For example, in the *Missing Capability* case, the baseline treats demographic attributes as ordinary customer tags, even though gender and age filters are unavailable to the customer segmentation sub-agent due to privacy and compliance policies. This response is problematic because it implies access to sensitive information that the system should not expose. *EARS* instead recognizes the capability boundary and guides the assistant to explain the unsupported filters while suggesting feasible alternatives. Similarly, in the *Ambiguous Query* case, the baseline assumes a proxy for wholesale interest without confirming the user's intended definition. *EARS* identifies this ambiguity and asks the assistant to clarify the signal before segmentation. The *Misrouting* case further shows that abstention can preserve user intent across agent boundaries. By identifying the required workflow sub-agent, *EARS* enables the coordinator to construct the necessary intermediate tag before invoking segmentation sub-agent. This illustrates how explanatory abstention supports multi-step recovery rather than merely preventing an incorrect local execution.

These cases show that abstention rationales not only prevent invalid outputs, but also help the coordinator clarify, redirect, or constrain the task before execution.

6 Conclusion

We presented *EARS*, a production-oriented framework for improving centralized MAS reliability through explanatory sub-agent abstention. Rather than treating abstention as a terminal user-facing refusal, *EARS* reframes it as an inter-agent communication protocol, enabling sub-agents to expose ac-

tionable failure states to the coordinator. Using calibrated LLM-as-a-Judge curation and supervised fine-tuning, *EARS* improves the overall response pass rate by 15.2% relatively in a production e-commerce assistant. These results demonstrate that explanatory abstention provides an effective coordination signal for improving MAS reliability.

7 Limitations

The proposed method is designed to be domain-adaptive, while our empirical evaluation focuses on an e-commerce business-intelligence assistant. Further validation is needed in other domains, where abstention scenarios and evaluation costs may differ. Future work should apply *EARS*'s curation and fine-tuning pipeline to MAS deployed in domains such as finance, education, and healthcare.

To obtain reliable abstention labels, we adopt a human-in-the-loop workflow for calibrating LLM-as-a-Judge models. While this design improves interpretability and reduces the risk of overfitting to the seed set, it also motivates more automated calibration strategies. Recent advances in automated judge revision and prompt refinement (Wang et al., 2024; Agrawal et al., 2025) should be explored to improve scalability while preserving label quality.

Finally, RL-based coordinator training or recovery-policy optimization could complement *EARS* by improving MAS-level response strategies following sub-agent abstention. We leave such system-level optimization to future work.

8 Acknowledgments

We thank Prof. Zhou Yu for valuable guidance. We are also grateful to Jason Liu, Nicolas Bertagnolli, and Aly Gemae for their helpful support and feedback throughout this work. We further thank Kaitlyn Chenier and the annotation team for their assistance with data annotation.

References

Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J. Hewett, Mojan Javaheripi, Piero Kauffmann, James R. Lee, Yin Tat Lee, Yuanzhi Li, Weishung Liu, Caio C. T. Mendes, Anh Nguyen, Eric Price, Gustavo de Rosa, Olli Saarikivi, and 8 others. 2024. Phi-4 Technical Report. *arXiv preprint*. ArXiv:2412.08905 [cs.CL].

Lakshya A Agrawal, Shangyin Tan, Dilara Soylu, Noah Ziems, Rishi Khare, Krista Opsahl-Ong, Arnav Singhvi, Herumb Shandilya, Michael J Ryan, Meng Jiang, and 1 others. 2025. Gepa: Reflective prompt evolution can outperform reinforcement learning. *arXiv preprint arXiv:2507.19457*.

Alfonso Amayuelas, Kyle Wong, Liangming Pan, Wenhu Chen, and William Wang. 2024. Knowledge of Knowledge: Exploring Known-Unknowns Uncertainty with Large Language Models. *arXiv preprint*. ArXiv:2305.13712 [cs.CL].

Anthropic. 2025. Introducing claude sonnet 4.5.

Anthropic. 2026. Claude sonnet 4.6. Model card and product page for `claude-sonnet-4-6`.

Ron Artstein and Massimo Poesio. 2008. Survey Article: Inter-Coder Agreement for Computational Linguistics. *Computational Linguistics*, 34(4):555–596.

Akari Asai and Eunsol Choi. 2021. Challenges in Information-Seeking QA: Unanswerable Questions and Paragraph Retrieval. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1492–1504, Online. Association for Computational Linguistics.

Faeze Brahman, Sachin Kumar, Vidhisha Balachandran, Pradeep Dasigi, Valentina Pyatkin, Abhilasha Ravichander, Sarah Wiegreffe, Nouha Dziri, Khyathi Chandu, Jack Hessel, Yulia Tsvetkov, Noah A. Smith, Yejin Choi, and Hannaneh Hajishirzi. 2024. The Art of Saying No: Contextual Noncompliance in Language Models. In *Advances in Neural Information Processing Systems*, volume 37, pages 49706–49748. Curran Associates, Inc.

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. 2025. Why Do Multi-Agent LLM Systems Fail? *arXiv preprint*. ArXiv:2503.13657 [cs.AI].

Maximillian Chen, Ruoxi Sun, Tomas Pfister, and Sercan Ö Arık. 2025. Learning to Clarify: Multi-turn Conversations with Action-Based Contrastive Self-Training. *arXiv preprint*. ArXiv:2406.00222 [cs.CL].

Yufan Dang, Chen Qian, Xueheng Luo, Jingru Fan, Zihao Xie, Ruijie Shi, Weize Chen, Cheng Yang, Xiaoyin Che, Ye Tian, Xuantang Xiong, Lei Han, Zhiyuan Liu, and Maosong Sun. 2025. Multi-Agent Collaboration via Evolving Orchestration. *arXiv preprint*. ArXiv:2505.19591 [cs.CL].

Shangbin Feng, Weijia Shi, Yike Wang, Wenxuan Ding, Vidhisha Balachandran, and Yulia Tsvetkov. 2024. Don't Hallucinate, Abstain: Identifying LLM Knowledge Gaps via Multi-LLM Collaboration. *arXiv preprint*. ArXiv:2402.00367 [cs.CL].

Adam Fourney, Gagan Bansal, Hussein Mozannar, Cheng Tan, Eduardo Salinas, Erkang, Zhu, Friederike Niedtner, Grace Proebsting, Griffin Bassman, Jack Gerrits, Jacob Alber, Peter Chang, Ricky Loynd, Robert West, Victor Dibia, Ahmed Awadallah, Ece Kamar, Rafah Hosn, and Saleema Amershi. 2024. Magentic-One: A Generalist Multi-Agent System for Solving Complex Tasks. *arXiv preprint*. ArXiv:2411.04468 [cs.AI].

Google. 2025. Gemini 2.5 flash.

Juraj Gottweis, Wei-Hung Weng, Alexander Daryin, Tao Tu, Anil Palepu, Petar Sirkovic, Artiom Myaskovsky, Felix Weissenberger, Keran Rong, Ryutaro Tanno, Khaled Saab, Dan Popovici, Jacob Blum, Fan Zhang, Katherine Chou, Avinatan Hassidim, Burak Gokturk, Amin Vahdat, Pushmeet Kohli, and 15 others. 2025. Towards an AI co-scientist. *arXiv preprint*. ArXiv:2502.18864 [cs.AI].

Bowen Jin, T. J. Collins, Donghan Yu, Mert Cemri, Shenao Zhang, Mengyu Li, Jay Tang, Tian Qin, Zhiyang Xu, Jiarui Lu, Guoli Yin, Jiawei Han, and Zirui Wang. 2025. Controlling Performance and Budget of a Centralized Multi-agent LLM System with Reinforcement Learning. *arXiv preprint*. ArXiv:2511.02755 [cs.CL].

Polina Kirichenko, Mark Ibrahim, Kamalika Chaudhuri, and Samuel J. Bell. 2025. AbstentionBench: Reasoning LLMs Fail on Unanswerable Questions. *arXiv preprint*. ArXiv:2506.09038 [cs.AI].

Pierre Lepagnol, Thomas Gerald, Sahar Ghannay, Christophe Servan, and Sophie Rosset. 2024. Small Language Models Are Good Too: An Empirical Study of Zero-Shot Classification. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 14923–14936, Torino, Italia. ELRA and ICCL.

Shuyue Stella Li, Vidhisha Balachandran, Shangbin Feng, Jonathan S. Ilgen, Emma Pierson, Pang Wei Koh, and Yulia Tsvetkov. 2024. MediQ: Question-Asking LLMs and a Benchmark for Reliable Interactive Clinical Reasoning. *arXiv preprint*. ArXiv:2406.00922 [cs.CL].

Nusrat Jahan Lia, Shubhashis Roy Dipta, Abdullah Khan Zehady, Naymul Islam, Madhusodan Chakraborty, and Abdullah Al Wasif. 2025. Read Between the Lines: A Benchmark for Uncovering Political Bias in Bangla News Articles. In *Proceedings of the Second Workshop on Bangla Language Processing (BLP-2025)*, pages 61–79, Mumbai, India. Association for Computational Linguistics.

Jingyu Liu, JingquanPeng JingquanPeng, Xiaopeng Wu, Xubin Li, Tiezheng Ge, Bo Zheng, and Yong Liu. 2025. Do not abstain! identify and solve the uncertainty. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 17177–17197.

Nishanth Madhusudhan, Sathwik Tejaswi Madhusudhan, Vikas Yadav, and Masoud Hashemi. 2024. Do LLMs Know When to NOT Answer? Investigating Abstention Abilities of Large Language Models. *arXiv preprint*. ArXiv:2407.16221 [cs.CL].

Aashiq Muhamed, Leonardo F. R. Ribeiro, Markus Dreyer, Virginia Smith, and Mona T. Diab. 2025. RefusalBench: Generative Evaluation of Selective Refusal in Grounded Language Models. *arXiv preprint*. ArXiv:2510.10390 [cs.CL].

OpenAI. 2025. Introducing gpt-5.

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. 2024. ChatDev: Communicative Agents for Software Development. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15174–15186, Bangkok, Thailand. Association for Computational Linguistics.

Qwen Team. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Xuancheng Ren, Shijing Hu, Zhihui Lu, Jiangqi Huang, and Qiang Duan. 2026. LatentRefusal: Latent-Signal Refusal for Unanswerable Text-to-SQL Queries. *arXiv preprint*. ArXiv:2601.10398 [cs.AI] version: 3.

Kyle Swanson, Wesley Wu, Nash L. Bulaong, John E. Pak, and James Zou. 2024. The Virtual Lab: AI Agents Design New SARS-CoV-2 Nanobodies with Experimental Validation. Pages: 2024.11.11.623004 Section: New Results.

Pat Verga, Sebastian Hofstatter, Sophia Althammer, Yixuan Su, Aleksandra Piktus, Arkady Arkhangorodsky, Minjie Xu, Naomi White, and Patrick Lewis. 2024. Replacing Judges with Juries: Evaluating LLM Generations with a Panel of Diverse Models. *arXiv preprint*. ArXiv:2404.18796 [cs.CL].

Xingyao Wang, Boxuan Li, Yufan Song, Frank F Xu, Xiangru Tang, Mingchen Zhuge, Jiayi Pan, Yueqi Song, Bowen Li, Jaskirat Singh, Hoang H Tran, Fuqiang Li, Ren Ma, Mingzhang Zheng, Bill Qian, Yanjun Shao, Niklas Muennighoff, Yizhe Zhang, Binyuan Hui, and 5 others. 2025. OPENHANDS: AN OPEN PLATFORM FOR AI SOFTWARE DEVELOPERS AS GENERALIST AGENTS. *arXiv preprint arXiv:2407.16741*.

Xinyuan Wang, Chenxi Li, Zhen Wang, Fan Bai, Haotian Luo, Jiayou Zhang, Nebojsa Jojic, Eric Xing, and Zhiting Hu. 2024. Promptagent: Strategic planning with language models enables expert-level prompt optimization. In *International Conference on Learning Representations*, volume 2024, pages 23967–24001.

Bingbing Wen, Jihan Yao, Shangbin Feng, Chenjun Xu, Yulia Tsvetkov, Bill Howe, and Lucy Lu Wang.

2025. Know Your Limits: A Survey of Abstention in Large Language Models. *arXiv preprint*. ArXiv:2407.18418 [cs.CL].

Feijie Wu, Zitao Li, Fei Wei, Yaliang Li, Bolin Ding, and Jing Gao. 2026. Talk to Right Specialists: Iterative Routing in Multi-agent Systems for Question Answering. *arXiv preprint*. ArXiv:2501.07813 [cs.MA].

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, Ahmed Awadallah, Ryen W White, Doug Burger, and Chi Wang. 2024. AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversations. *arXiv preprint arXiv:2308.08155*.

Yanwei Yue, Guibin Zhang, Boyang Liu, Guancheng Wan, Kun Wang, Dawei Cheng, and Yiyan Qi. 2025. MasRouter: Learning to Route LLMs for Multi-Agent Systems. *arXiv preprint*. ArXiv:2502.11133 [cs.LG].

Skylar Zhai, Jingcheng Liang, and Dongyeop Kang. 2026. Abstain-R1: Calibrated Abstention and Post-Refusal Clarification via Verifiable RL. *arXiv preprint*. ArXiv:2604.17073 [cs.CL] version: 1.

Wentao Zhang, Liang Zeng, Yuzhen Xiao, Yongcong Li, Ce Cui, Yilei Zhao, Rui Hu, Yang Liu, Yahui Zhou, and Bo An. 2026. AgentOrchestra: Orchestrating Multi-Agent Intelligence with the Tool-Environment-Agent(TEA) Protocol. *arXiv preprint*. ArXiv:2506.12508 [cs.AI].

Zheyuan Zhang, Kaiwen Shi, Zhengqing Yuan, Zehong Wang, Tianyi Ma, Keerthiram Murugesan, Vincent Galassi, Chuxu Zhang, and Yanfang Ye. 2025. AgentRouter: A Knowledge-Graph-Guided LLM Router for Collaborative Multi-Agent Question Answering. *arXiv preprint*. ArXiv:2510.05445 [cs.CL].

A Semantic Correctness Judge Prompts

We use task-specific semantic correctness prompts for analytics outputs and segmentation outputs. These prompts are used by the final evaluator, `openai:gpt-5` with `temperature=0`, and are separate from the abstention-labeling prompts used by the data-curation judges. The evaluator shares the `gpt-5` base model with one curation judge, but it is run as a separate evaluation pipeline and is not used to label SFT data. Before final evaluation, the prompt is calibrated on a separate 1,050 example human-annotated split. The production prompts include platform-specific schemas, domain terminology, and internal unsupported-request lists. For anonymized disclosure, we report sanitized excerpts that preserve the evaluation protocol while masking platform names, schema fields, and domain-specific capabilities.

A.1 Analytics Judge Prompt

```
Input:
  user_message: the original user request
  context: request metadata, including request time when available
  model_output: the generated [ANALYTICS_QUERY_LANGUAGE] query

Task:
  Determine whether model_output answers the user's request.
  Return only a JSON object with:
    explanation: detailed reasoning about intent alignment and validity
    result: true if the query is correct, false otherwise

Core checks:
  1. Validate that all metrics, dimensions, and models are allowed by
     [ANALYTICS_SCHEMA_RESTRICTIONS].
  2. Check that the query matches the user's intent, including filters,
     aggregation level, requested entity type, and requested time range.
  3. If the user asks for a visualization, require a supported visualization
     directive from [SUPPORTED_VISUALIZATION_TYPES].
  4. For trend requests, require an ordering or grouping that makes the trend interpretable.
  5. For top, highest, maximum, or best requests, require descending ordering
     by an appropriate metric.
  6. For bottom, lowest, minimum, or least requests, require ascending ordering
     by an appropriate metric.
  7. If no date is specified, treat all-time data as the default.
  8. Interpret relative date ranges using the request timestamp in context.
  9. Apply domain-specific semantic rules for reversal metrics, traffic-source
     dimensions, entity-name matching, and exclusion filters, with all
     platform-specific field names masked.
  10. Reject outputs that attempt unsupported reports listed in
      [UNSUPPORTED_REQUEST_TYPES].
```

A.2 Segmentation Judge Prompt

```
Input:
  user_message: the original user request
  context: request metadata, including request time when available
  model_output: a generated [SEGMENTATION_FILTER_LANGUAGE] expression or
    structured abstention signal

Task:
  Evaluate whether model_output correctly handles the user's request.
  Return only a JSON object with:
    explanation: detailed reasoning about the selected case and final verdict
    result: true if the output is correct, false otherwise

Case selection:
  A. Standard query:
     Use this case when model_output does not contain an abstention signal.
     Check whether the filters, operators, values, and date constraints match
     the user's intent under [SUPPORTED_SEGMENTATION_FILTERS].

  B. Abstention query:
     Use this case when model_output contains an abstention signal.
     Independently decide whether the request is:
       solvable: supported with information already provided
       insufficient_input: supported but missing concrete user-provided input
       not_solvable: unsupported or too ambiguous to execute
     If not_solvable, assign exactly one reason:
       missing_capability
       misrouting
       ambiguous_query
     The model output is correct only when its abstention type and reason match
     the independent assessment.

Core checks:
  1. Confirm intent alignment before judging syntax-level details.
  2. Treat segmentation as audience-list construction, not aggregate analytics.
  3. Reject requests that require aggregate reports, ranking, visualization,
     mutation, or unsupported capabilities in [UNSUPPORTED_REQUEST_TYPES].
  4. Use request time from context for relative dates.
  5. Prefer lifecycle/value-group filters for vague high-value or loyalty
     requests when no numeric threshold is provided.
  6. Do not accept invented tags, unsupported attributes, or proxy filters that
     create unsupported behavior.
```

Dimension	Accuracy	Quadratic κ	Kendall's τ
Claim Groundedness	70.4%	0.30	0.38
Goal Fulfillment	73.0%	0.53	0.53
User Satisfaction	82.0%	0.44	0.45
Action Execution Success	89.1%	0.24	0.20
Response Usability	99.0%	0.57	0.57

Table 5: Agreement between ACE and human expert annotations on 311 held-out conversations.

B Human Evaluation Details

This appendix reports the response-quality rubric used for the full MAS human evaluation in Section 4. Domain annotators assessed session-level assistant responses according to the rubric in Table 6. A rating of 1 is counted as a pass, whereas a rating of 0 is counted as a failure.

Score	Label	Rubric
0	Poor Quality	The response has one or more serious issues: confusing or illogical explanations, excessive verbosity, unnecessary or incorrect tool calls, criticism of platform features or recommendations for competitors, inappropriate tone, or failure to provide actionable guidance.
1	Satisfactory	The response is coherent, professionally toned, and sufficiently addresses the user's query. It may contain minor structure, flow, or conciseness issues, but uses tools appropriately and provides actionable guidance.

Table 6: Human E2E response-quality rubric.

C LLM-as-a-Judge Calibration Analysis

Table 7 reports the prediction accuracy of each LLM-as-a-Judge on the 602 human-annotated abstention examples only. This abstention-only sensitivity analysis is separate from the all-sample calibration results in Table 2, which also includes solvable examples. Each judge evaluates the same outputs independently. `claude-sonnet-4-5` achieves the highest abstention-only overall prediction accuracy at 87.6%, followed by `gemini-2.5-flash`. Judges perform most reliably on *Missing Capability* and *Misrouting* cases, where the decision often depends on whether the requested function or target worker is supported. In contrast, *Ambiguous Query* and *Insufficient Input* remain harder and show larger variation across judges. The confusion matrices in Figure 3 help explain this pattern: ambiguous queries are often confused with insufficient-input cases, since both involve underspecified user requests. Insufficient-input cases are also frequently confused with missing-capability or solvable cases, suggesting that judges differ in whether they interpret missing execution details as a request for unsupported functionality, an answerable query, or a query that requires additional user input.

LLM-as-a-Judge	Overall Accuracy	Ambiguous Query	Insufficient Input	Missing Capability	Misrouting
`gpt-5`	79.6%	45.5%	58.3%	87.6%	86.8%
`o3`	79.6%	33.3%	66.0%	85.6%	83.1%
`claude-sonnet-4-5`	**87.6%**	**83.3%**	**76.7%**	90.8%	**90.9%**
`gemini-2.5-flash`	84.7%	41.7%	70.9%	**91.5%**	85.7%

Table 7: Abstention-only judge sensitivity analysis. Each judge evaluates the same 602 human-annotated abstention examples. Category columns report per-category prediction accuracy under the final taxonomy.

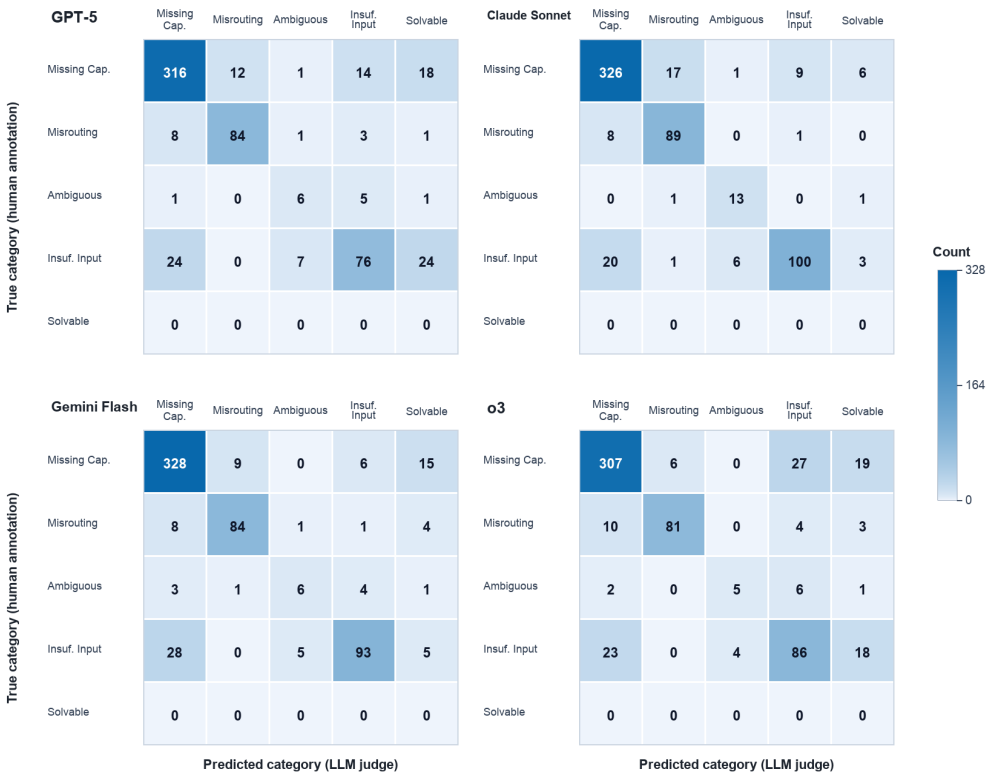

Figure 3: Abstention judge confusion matrices for final-model abstention-category predictions. The y-axis gives the true human-annotated category, and the x-axis gives the category predicted by each LLM judge. Darker cells indicate larger counts.