What Question Did You Answer? Refining Contact Center Evaluation Plans via Backward Questions

Prajwal Sood **Rushikesh Pawar** **Digvijay Ingle** **Anup Pattnaik**
Observe.AI, India
{prajwal.sood, rushikesh.pawar, digvijay.ingle, anup.pattnaik}@observe.ai

Abstract

Capturing organization-specific domain knowledge remains a critical challenge for deploying cost-efficient language models in specialized tasks like contact center Quality Assurance (QA). While large LMs implicitly capture expert judgment, smaller LMs require explicit evaluation criteria that domain experts struggle to articulate. We introduce Backward Question-based Refinement (**BQR**), a diagnostic framework that generates backward questions, revealing what a model understood rather than what was asked, to systematically distill implicit reasoning from large LMs into explicit evaluation plans. Through experiments on 12 QA questions, BQR achieves performance improvements on 8 questions with absolute gains of up to 27.8% in Macro F1. Our analysis establishes empirical parallels to gradient-descent optimization and reveals a cross-family advantage where small LMs benefit more from large LMs of different families. These findings confirm BQR as an effective approach for bridging the gap between implicit expert knowledge and explicit evaluation criteria.

1 Introduction

Contact centers process millions of conversations daily that directly impact customer satisfaction and regulatory compliance. Within this ecosystem, Quality Assurance (QA) teams evaluate agent performance against organizational standards through standardized questionnaires (Roy et al., 2016). Recent advances in large Language Models (LMs) offer unprecedented potential for automating these evaluations (Ingle et al., 2024; Sood et al., 2025), yet face a fundamental challenge: effectively capturing organization-specific evaluation criteria.

Ingle et al. (2024) established that structured breakdown of QA questions into *evaluation plans* significantly enhances LM performance. While large LMs generate effective plans through extensive world knowledge, they reflect generic industry best practices rather than organization-specific policies. For instance, *"demonstrating active listening"* varies across organizations: one might prioritize verbal acknowledgments, another emphasizes proactive problem-solving. This misalignment becomes critical when using smaller, cost-effective models that rely on explicit criteria rather than implicit contextual interpretation. The challenge compounds further: QA analysts possess deep domain expertise but struggle to articulate it for AI systems, as this expertise lies in their *"muscle memory"*.

We hypothesize that the performance gap between large and small LMs (Ingle et al., 2024) reveals not just capability differences, but a specific deficiency in capturing implicit domain knowledge. Owing to their diverse world knowledge, large LMs excel at inferring unstated implications, while smaller models (Sood et al., 2025) struggle with implicit scenarios, often missing critical signals that lack explicit textual cues. This suggests a potential for using large LMs to extract signals to enrich evaluation plans, making implicit criteria explicit.

We introduce a methodology for automatic plan refinement by: (1) identifying systematic failure patterns where smaller LMs miss implicit signals, (2) enriching plans with explicit criteria codifying organizational standards. Through empirical evaluation, we demonstrate substantial performance improvements, making cost-effective automation viable without sacrificing organizational alignment. Our contributions include:

- We introduce **Backward Question-based Refinement (BQR)**, a diagnostic framework that corrects model misinterpretations by generating backward questions revealing what the model understood rather than what was asked.
- We demonstrate how BQR produces distinct refinement strategies adapted to question characteristics and uncover the advantage of using

cross-family large LMs for optimization.

- We draw intuitive analogies to gradient-based optimization, where outlier removal resembles regularization and generation temperature resembles learning-rate control.

While significant progress has been made in forward-backward reasoning (Chen et al., 2025) for LM response verification, to the best of our knowledge we are the first to leverage backward reasoning to distill implicit domain knowledge into explicit evaluation criteria.

2 Problem Formulation

Building upon the foundational QA task established by Ingle et al., 2024, we formalize the problem of iterative plan refinement for contact center QA. We begin by summarizing the QA evaluation task before introducing our extended formulation.

QA Evaluation Task: Given a conversation $\mathcal{C}$ between an agent and a customer, and an evaluation question $\mathcal{Q}$, the goal is to arrive at a reasoning $\mathcal{R}$ and answer $\mathcal{A} \in \{yes, no\}$, such that $\mathcal{R}$ logically leads to $\mathcal{A}$. This requires extraction and synthesis of information relevant to answer $\mathcal{Q}$ from $\mathcal{C}$.

To guide this reasoning process, an evaluation plan $\mathcal{P}$ is introduced, which decomposes $\mathcal{Q}$ into a structured set of assessment criteria. Formally, $\mathcal{P} = \{c_1, c_2, \ldots, c_k\}$ where each criterion c_i specifies a concrete aspect to evaluate. The model then processes the triplet $(\mathcal{Q}, \mathcal{P}, \mathcal{C})$ to produce $(\mathcal{R}, \mathcal{A})$.

Iterative Plan Refinement Task: We formulate plan refinement as an iterative process that improves model performance by distilling implicit reasoning from larger models into explicit criteria. Let $\mathcal{D} = \{(\mathcal{Q}_i, \mathcal{C}_i, \mathcal{A}_i^*)\}_{i=1}^N$ denote a dataset of evaluation instances with ground truth answers $\mathcal{A}_i^*$ aligned with organizational policy, and let $\mathcal{D}_Q \subset \mathcal{D}$ represent the subset relevant to question $\mathcal{Q}$.

Given an initial plan $\mathcal{P}^{(0)}$ for question $\mathcal{Q}$, a small LM $\mathcal{M}_s$ (cost-efficient but limited reasoning), a large LM $\mathcal{M}_l$ (capable but expensive) and historical evaluation dataset $\mathcal{D}_Q$, our goal is to find an optimal plan $\mathcal{P}^*$ that maximizes the performance of $\mathcal{M}_s$ on organization-specific evaluation criteria:

$$\mathcal{P}^* = \arg\max_{\mathcal{P}} \Psi(\mathcal{M}_s, \mathcal{P}, \mathcal{D}_Q)$$

where $\Psi(\cdot)$ represents an appropriate evaluation metric that is determined by business requirements and operational constraints.

3 Proposed Methodology

We introduce a Backward Question-based Refinement (BQR) framework, an automated algorithm to improve the evaluation plan by systematically distilling implicit reasoning capabilities from $\mathcal{M}_l$ into explicit evaluation criteria accessible to $\mathcal{M}_s$.

3.1 The BQR Framework

To ensure systematic evaluation during the refinement process, we first partition $\mathcal{D}_Q$ into three disjoint subsets: $\mathcal{D}_Q^{\text{train}}$, $\mathcal{D}_Q^{\text{val}}$, and $\mathcal{D}_Q^{\text{test}}$ such that $\mathcal{D}_Q = \mathcal{D}_Q^{\text{train}} \cup \mathcal{D}_Q^{\text{val}} \cup \mathcal{D}_Q^{\text{test}}$. Each iteration t of the BQR framework consists of the following steps:

Step 1: Forward Evaluation
For each instance $(\mathcal{Q}, \mathcal{C}_i, \mathcal{A}_i^*) \in \mathcal{D}_Q^{\text{train}}$, we generate responses using the current plan $\mathcal{P}^{(t)}$ with $\mathcal{M}_s$, each consisting of reasoning $\mathcal{R}$ and an answer $\mathcal{A}$.

$$(\mathcal{R}_{i,s}^{(t)}, \mathcal{A}_{i,s}^{(t)}) = \mathcal{M}_s(\mathcal{Q}, \mathcal{P}^{(t)}, \mathcal{C}_i)$$

Step 2: Generation of Backward Questions
While $\mathcal{R}_{i,s}^{(t)}$ represents chain-of-thought (CoT) (Wei et al., 2022) reasoning to arrive at answer $\mathcal{A}_{i,s}^{(t)}$ using $\mathcal{P}^{(t)}$, it provides limited insight into why a model fails in cases where $\mathcal{A}_{i,s}^{(t)} \neq \mathcal{A}_i^*$. Specifically, it fails to reveal which aspects of $\mathcal{Q}$ were misunderstood. To address this, we introduce backward question generation as a diagnostic mechanism to reveal model's internal interpretation of $\mathcal{Q}$.

Definition: Given a conversation $\mathcal{C}$ and a reasoning-answer pair $(\mathcal{R}, \mathcal{A})$, a *backward question* is a question $\mathcal{Q}'$ for which $\mathcal{A}$ would constitute a correct answer derived using $\mathcal{R}$ based on $\mathcal{C}$. Formally, the backward question generation step is:

$$\mathcal{Q}_i'^{(t)} = \text{BACKWARD}(\mathcal{C}_i, \mathcal{R}_{i,s}^{(t)}, \mathcal{A}_{i,s}^{(t)}, \mathcal{P}^{(t)})$$
$$\mathcal{Q}'^{(t)} = \bigcup_i \mathcal{Q}_i'^{(t)} \quad \forall \quad (\mathcal{Q}, \mathcal{C}_i, \mathcal{A}_i^*) \in \mathcal{D}_Q^{\text{train}}$$

Here, we prompt $\mathcal{M}_l$ to generate diverse questions about $\mathcal{C}_i$ that would yield the answer $\mathcal{A}$. Refer to Appendix A for illustrative examples of $\mathcal{Q}'$.

The rationale for this approach draws from recent work on forward-backward reasoning for verification (Jiang et al., 2024). While forward reasoning (question $\rightarrow$ answer) tests execution capability, backward reasoning (answer $\rightarrow$ question) reveals the model's conceptual understanding of the task space. If a model generates a backward question that is semantically distant from $\mathcal{Q}$, it indicates a fundamental misunderstanding, even if it can computationally process the instructions.

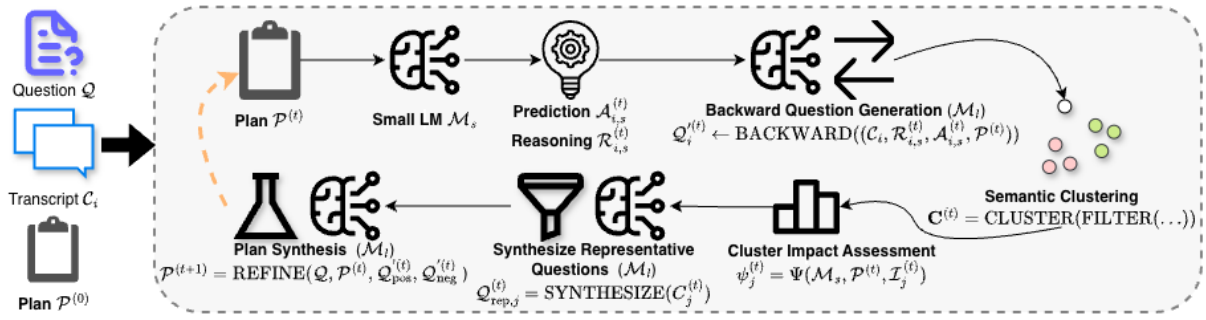

Figure 1: **The Backward Question-based Refinement (BQR) framework.** The framework exploits the reasoning disparity between $\mathcal{M}_s$ and $\mathcal{M}_l$ to iteratively optimize evaluation plans. In each iteration, $\mathcal{M}_l$ diagnoses $\mathcal{M}_s$'s responses by generating backward questions. These questions are clustered to identify systematic patterns, allowing $\mathcal{M}_l$ to synthesize a updated plan $\mathcal{P}^{(t+1)}$, which is progressively refined until an optimal plan $\mathcal{P}^*$ is achieved.

Step 3: Diagnostic Signal Aggregation

The raw backward questions in $\mathcal{Q}'^{(t)}$ often contain noise, formatting inconsistencies, and semantic redundancies. Hence, we process the backward questions as follows:

Normalization and Filtration: We first apply text normalization to each backward question $\mathcal{Q}_i'^{(t)} \in \mathcal{Q}'^{(t)}$ to remove formatting noise and standard phrasing, extracting its core semantic content $\tilde{\mathcal{Q}}_i'^{(t)}$. To eliminate hallucinated or irrelevant queries, we apply an outlier removal algorithm with threshold r (the proportion removed), producing a clean, filtered set $\tilde{\mathcal{Q}}_f'^{(t)}$ (refer Appendix B).

Semantic Clustering: To identify semantically similar patterns, we group the remaining normalized questions into a set of clusters $\mathbf{C}^{(t)}$:

$$\tilde{\mathcal{Q}}_f'^{(t)} = \text{FILTER}\big(\{\tilde{\mathcal{Q}}_i'^{(t)} \mid \mathcal{Q}_i'^{(t)} \in \mathcal{Q}'^{(t)}\}, r\big)$$
$$\mathbf{C}^{(t)} = \text{CLUSTER}\big(\tilde{\mathcal{Q}}_f'^{(t)}, k\big)$$

where k represents the number of clusters and $\mathbf{C}^{(t)} = \{C_1^{(t)}, C_2^{(t)}, \ldots, C_k^{(t)}\}$.

Step 4: Plan Refinement

We introduce a principled approach to prioritize refinement efforts based on cluster impact.

Cluster Impact Assessment and Selection: For each cluster $C_j^{(t)} \in \mathbf{C}^{(t)}$, we retrieve its associated training instances $\mathcal{I}_j^{(t)}$ and compute a cluster-specific performance score:

$$\psi_j^{(t)} = \Psi(\mathcal{M}_s, \mathcal{P}^{(t)}, \mathcal{I}_j^{(t)})$$

Ranking by $\psi_j^{(t)}$, we select the top n highest-performing (positive) clusters $\mathbf{C}_{pos}^{(t)}$ and the bottom m lowest-performing (negative) clusters $\mathbf{C}_{neg}^{(t)}$.

Representative Question Synthesis: For each selected cluster, we prompt $\mathcal{M}_l$ to analyze all backward questions within it and synthesize a single, coherent representative question:

$$\mathcal{Q}_{\text{rep},j}'^{(t)} = \text{SYNTHESIZE}(C_j^{(t)})$$

This yields two distinct sets: $\mathcal{Q}_{\text{pos}}'^{(t)}$ capturing near-correct interpretations, and $\mathcal{Q}_{\text{neg}}'^{(t)}$ capturing severe misalignments.

Plan Synthesis: We prompt $\mathcal{M}_l$ with a meta-prompt containing the evaluation question $\mathcal{Q}$, the current plan $\mathcal{P}^{(t)}$, $\mathcal{Q}_{\text{pos}}'^{(t)}$, and $\mathcal{Q}_{\text{neg}}'^{(t)}$. $\mathcal{M}_l$ analyzes this spectrum to directly synthesize an updated plan that corrects the identified misalignments while reinforcing the correct logic:

$$\mathcal{P}^{(t+1)} = \text{REFINE}(\mathcal{Q}, \mathcal{P}^{(t)}, \mathcal{Q}_{\text{pos}}'^{(t)}, \mathcal{Q}_{\text{neg}}'^{(t)})$$

Step 5: Validation and Iteration Control

We evaluate the refined plan $\mathcal{P}^{(t+1)}$ on $\mathcal{D}_Q^{\text{val}}$ using Ψ and denote this as Ψ^{val}. To prevent overfitting, we enforce an early stopping with patience p and call the plan achieving the highest Ψ^{val} across all iterations as the optimal plan $\mathcal{P}^*$ (refer Appendix C).

Final Evaluation: Finally, upon reaching the stopping criteria (via early stopping or reaching T_{max}), we evaluate $\mathcal{M}_s$ on the held-out test set $\mathcal{D}_Q^{\text{test}}$ using the optimal plan $\mathcal{P}^*$.

4 Experimental Setup

4.1 Data Curation

Building upon methodologies from prior works (Ingle et al., 2024; Sood et al., 2025), we construct a specialized dataset of real-world English dyadic conversations from a proprietary contact center corpus spanning banking, automotive, and healthcare

domains. We select 12 representative QA questions evaluating agent performance across procedural compliance, soft skills, and call-handling mechanics. We sample 300 conversations per question yielding an initial pool of approximately 3,600 question-conversation pairs.

A panel of five domain experts independently annotates each pair with binary labels (*yes*/*no*) following structured protocols from Ingle et al. (2024). We apply rigorous consensus filtering, retaining only instances with strong agreement ($\geq$4/5 annotators). This yields a refined dataset of 2,297 question-conversation pairs (63.8% of initial sample), each assigned its respective consensus label. We denote this dataset as $\mathcal{D}$[1], consistent with the formulation in Section 2. Detailed label distributions are provided in Appendix D.

4.2 Implementation Details

We partition $\mathcal{D}_Q$ into train, validation, and test sets following Section 3.1, employing stratified sampling in a 50:25:25 ratio to preserve label distribution. We fix $\mathcal{M}_s$ as `nova-lite` for its suitability in high-volume deployment due to lower inference costs (Sood et al., 2025), and $\mathcal{M}_l$ as `nova-pro` for superior reasoning abilities (Sood et al., 2025) (see Appendix E for classification rationale). By fixing both models, we first establish the potential of BQR before extending to other LMs in Section 5.3.

We employ `all-mpnet-base-v2` (Reimers and Gurevych, 2019) for generating sentence embeddings of backward questions[2], K-Means (MacQueen, 1967) for clustering, and Isolation Forest (Liu et al., 2008) for outlier detection, chosen for computational efficiency. Although exploring alternative clustering and outlier detection algorithms presents interesting future work, our framework is agnostic to these choices, and we defer such investigations to future work to maintain experimental focus. All LM inferences use greedy decoding ($\tau = 0$) for reproducibility. All prompts are provided in Appendix F.

Without loss of generality, we fix Ψ as Macro F1, though the framework generalizes to any metric tailored to business requirements. We conduct hyperparameter search over: contamination parameter $r \in \{0, 0.1, 0.25, 0.5\}$, clusters $k \in \{5, 10, 15\}$, top/bottom clusters $n, m \in \{0, 1, 2\}$, maximum iterations $T_{\max} \in \{5, 10, 15\}$, and patience $p \in \{2, 3, 4, 5\}$. We select the optimal configuration based on Macro F1 on $\mathcal{D}_Q^{\text{val}}$ and report the best-performing plan $\mathcal{P}^*$ evaluated on $\mathcal{D}_Q^{\text{test}}$ across all 12 questions in Table 1.

We additionally compare BQR against two optimization-based baselines, OPRO (Yang et al., 2023) and PRESTO (Chu et al., 2025), under identical train/validation/test splits, the same initial plan $\mathcal{P}^{(0)}$, and the same target small model $\mathcal{M}_s$ = `nova-lite`. Each method optimizes the textual evaluation plan using validation-set Macro F1 and reports the best-performing plan on the held-out test set. To assess deployment practicality, we also evaluate BQR with LLM-generated labels in place of expert labels during refinement, and estimate optimization cost using the costing methodology introduced in STREAQ (Sood et al., 2025).

Question	Macro F1 Score (%)			Δ Len
	$\mathcal{M}_s$ (Base)	$\mathcal{M}_l$ (Base)	$\mathcal{M}_s$ (Opt)	
Address Name	59.9	67.5	**65.7**	+33.3%
Disclose PII	45.5	81.5	45.5	0.0%
Positive Tone	45.2	67.8	**56.9**	-1.2%
Willing Assist	42.9	86.6	**62.9**	+179.3%
Natural Style	47.9	60.3	**50.0**	+388.9%
Ack. Issue	93.0	95.3	90.7	+50.8%
Clear Disconnect	65.3	83.0	**83.7**	+37.1%
Demo Underst.	67.1	71.1	**69.3**	+56.3%
Avoid Assist	50.9	65.3	**62.6**	+87.8%
Gather Info	66.7	42.9	66.7	0.0%
Focus Resol.	59.3	82.0	**87.1**	+290.3%
Ask Repeat	69.3	65.7	63.3	+171.8%

Table 1: Performance comparison across evaluation questions. (Base) denotes performance using the initial plan $\mathcal{P}^{(0)}$, while (Opt) denotes performance using the optimized plan $\mathcal{P}^*$. **Bold** indicates improvement over the $\mathcal{M}_s$ baseline. Blue indicates the optimized plan outperforms both baselines. Δ Len represents the percentage change in plan verbosity from $\mathcal{P}^{(0)}$ to $\mathcal{P}^*$.

5 Results and Discussion

The results in Table 1 show that BQR improves the optimized small-model plan over the baseline plan in **8 out of 12 questions** with $\mathcal{M}_s$ = `nova-lite` and $\mathcal{M}_l$ = `nova-pro`. In the subsequent sections, we analyze where these gains arise, compare BQR to alternative optimization strategies, and discuss settings where refinement is less likely to help.

5.1 Comparison to Optimization Baselines

Table 2 compares BQR against OPRO (Yang et al., 2023) and PRESTO (Chu et al., 2025) under identical settings. BQR outperforms OPRO on **8/12**

[1]We cannot release the dataset due to proprietary reasons.

[2]https://huggingface.co/sentence-transformers/all-mpnet-base-v2

Question	$\mathcal{M}_s$ (Base)	$\mathcal{M}_l$ (Base)	OPRO	PRESTO	BQR
Address Name	59.9	67.5	60.4	**67.1**	65.7
Disclose PII	45.5	81.5	46.2	**50.3**	45.5
Positive Tone	45.2	67.8	52.1	55.2	**56.9**
Willing Assist	42.9	86.6	55.3	60.4	**62.9**
Natural Style	47.9	60.3	49.1	49.8	**50.0**
Ack. Issue	**93.0**	95.3	92.1	91.2	90.7
Clear Disconnect	65.3	83.0	74.8	82.1	**83.7**
Demo Underst.	67.1	71.1	68.4	68.7	**69.3**
Avoid Assist	50.9	65.3	57.2	60.8	**62.6**
Gather Info	66.7	42.9	66.7	**68.4**	66.7
Focus Resol.	59.3	82.0	72.6	84.3	**87.1**
Ask Repeat	**69.3**	65.7	64.1	67.1	63.3

Table 2: Comparison against optimization-based baselines under identical settings. OPRO (Yang et al., 2023), PRESTO (Chu et al., 2025) and BQR (this work) optimize the same initial plan for the same small model. **Bold** indicates the best score among the base small-model result and the optimized small-model variants.

questions and PRESTO on **7/12** questions. The advantage is most pronounced on subjective questions with underspecified base plans. For *"Willing Assist,"* BQR reaches **62.9%** Macro F1 versus **55.3%** for OPRO and **60.4%** for PRESTO. For *"Focus Resol.,"* BQR achieves **87.1%** compared with **72.6%** and **84.3%**, respectively. These cases suggest that backward questions provide a more targeted diagnostic signal by revealing what $\mathcal{M}_s$ actually understood, rather than relying only on candidate scores or score prediction, indicating that the diagnostic signals from BQR extend the gains achieved through iterative optimization.

5.2 Impact of Plan Verbosity

We analyze the relationship between plan verbosity (measured by word count) and performance improvements, revealing three distinct refinement strategies: *Additive*, *Subtractive*, and *Substitutive*, each tailored to question characteristics.

Additive Refinement: For subjective evaluation criteria, we observe strong positive correlation between plan expansion and performance gains. The question *"Willing Assist"* exemplifies this pattern, where Macro F1 saw an absolute gain of 20.0% as the plan evolved from vague sentiment assessment into a structured checklist of observable behavioral markers (Appendix J). This demonstrates that smaller models require explicit articulation of abstract concepts. Rather than inferring "willingness to assist" from contextual cues, the model benefits from concrete behavioral anchors such as "personally commit" and "outline steps" verifiable against the transcript.

Subtractive Refinement: Conversely, verbosity is not strictly necessary for improvement. For question *"Positive Tone"*, BQR increased the score by an absolute 11.7% while maintaining nearly identical plan length. As shown in Appendix J, the framework performed *subtractive* refinement by removing extraneous formatting elements (headers, bullet points) and consolidating core logic into dense, semantic instructions.

Substitutive Refinement: For questions requiring adherence to organization-specific procedures, BQR applies *substitutive refinement*, replacing generic guidelines with targeted operational constraints encoding tacit organizational knowledge. Question *"Address Name"* exemplifies this with an absolute 5.8% Macro F1 increase via precise rephrasing (Appendix J). The framework transforms generic phrases ("look into the complete call") into precise temporal constraints ("from the beginning of the call transcript") and explicit conclusion criteria ("as a condition to conclude with a yes or no answer"). This reinforces that smaller models require direct articulation of evaluation logic rather than relying on contextual inference.

Small LM ($\mathcal{M}_s$)	$\mathcal{M}_s$ (Base)	nova-pro	nova-premier	claude-4.5-sonnet	gpt-5
nova-lite	65.1%	74.1%	70.0%	**76.0%**	74.0%
gpt-5-nano	66.0%	73.0%	**73.2%**	72.0%	70.2%
claude-4.5-haiku	72.0%	-	**81.0%**	75.0%	76.0%

Table 3: Cross-family $\mathcal{M}_s - \mathcal{M}_l$ pairings yield the highest performance in all 3 cases (highlighted in **bold**). We do not compute the result for `claude-4.5-haiku` and `nova-pro`, as `claude-4.5-haiku` is costlier among the two. Refer to Appendix E for more details.

5.3 Cross-Family Relationship

We extend Section 4.2 to investigate interactions between $\mathcal{M}_s$ and $\mathcal{M}_l$ from different model families. We vary $\mathcal{M}_s \in \{$`nova-lite`, `gpt-5-nano`, `claude-4.5-haiku`$\}$ and $\mathcal{M}_l \in \{$`nova-pro`, `nova-premier`, `gpt-5`, `claude-4.5-sonnet`$\}$, execute BQR as formalized in Section 3.1 and report Macro F1 averaged across all questions on the test set. Table 3 reveals a cross-family advantage: the large LM within a family does not necessarily yield optimal performance when instructing its smaller sibling.

The "Echo Chamber" Effect: Counter-intuitively, $\mathcal{M}_s$ often converges higher with cross-family $\mathcal{M}_l$. For instance, `claude-4.5-haiku` achieved **75.0%** when optimized by `claude-4.5-sonnet` but improved to **81.0%** with `nova-premier`. Similarly, `gpt-5-nano` scored **70.2%** with `gpt-5` versus

73.2% with `nova-premier`. We hypothesize that this stems from shared architectural biases: when $\mathcal{M}_l$ and $\mathcal{M}_s$ share training data and RLHF alignment, $\mathcal{M}_l$ reinforces $\mathcal{M}_s$'s latent priors and blind spots. In contrast, cross-family $\mathcal{M}_l$ introduces novel reasoning patterns, forcing generalization beyond inherent distributions.

6 Failure Modes and Practical Guidance

The same benchmark also highlights settings in which refinement provides limited benefit. We observe two recurring failure modes. First, already-explicit procedural plans such as *"Disclose PII"* and *"Gather Info"* leave little room for improvement because the initial instructions already specify concrete, operational criteria. In these cases, BQR produces either no change or only marginal differences, and OPRO/PRESTO behave similarly. Second, refinement is less effective in low-headroom settings where either the base small-model score is already high or the large model does not meaningfully outperform the small model. For *"Ack. Issue,"* the base score is already **93.0%**, leaving little room for gains. For *"Ask Repeat,"* $\mathcal{M}_l$ itself only reaches **65.7%**, indicating a weak diagnostic gap for BQR to exploit.

These observations yield two practitioner-facing heuristics. Refinement is less likely to help when (1) the baseline Macro F1 is already near ceiling, or (2) $\mathcal{M}_l$ does not meaningfully outperform $\mathcal{M}_s$ on the target question. Under these conditions, the safer choice may be to retain the original plan or prioritize data and label improvements instead. Appendix J, Appendix K, and Appendix L provide representative examples of how refinement helps when those conditions are not met.

7 Ablation Studies

To isolate contributions of individual components within BQR, we conduct systematic ablation with controlled variations across three levers: backward question generation temperature $\tau \in \{0, 1\}$, inclusion of chain-of-thought (CoT) reasoning traces during backward question generation, and outlier removal threshold $r \in \{0, 0.1, 0.25, 0.5\}$. Table 4 presents results, where values represent Macro F1 averaged across all 12 questions, with $\mathcal{M}_s$ as `nova-lite` and $\mathcal{M}_l$ as `nova-pro`.

7.1 Necessity of Chain-of-Thought Reasoning

We analyze the impact of including CoT reasoning during backward question generation. Comparing setups with CoT (A1, A3) against those without (A0, A2), we observe that CoT consistently enhances performance across all temperature and outlier configurations. We attribute this to intermediate reasoning steps, which encourage $\mathcal{M}_l$ to objectively decompose complex implicit reasoning before formulating questions. This ensures generated backward questions ground in specific logical gaps rather than superficial label mismatches.

7.2 Temperature as Learning Rate

Comparing Setups A1 ($\tau = 0$) and A3 ($\tau = 1$), backward question generation temperature behaves analogously to learning rate in gradient descent. Greedy decoding ($\tau = 0$) provides stable but slow improvement, achieving **73.1%** over **6.35 iterations**. This conservative approach resembles a small learning rate, reliable but prone to local minima with limited exploration.

Increasing temperature to $\tau = 1$ (Setup A3) introduces exploratory behavior, allowing $\mathcal{M}_l$ to generate diverse reasoning perspectives rather than similar backward questions. This diversity enables broader exploration of plan refinements, achieving convergence over **2.5 iterations**.

7.3 Outlier Removal as Regularization

The outlier removal threshold (r) exhibits behavior analogous to regularization in gradient-descent optimization. Removing outliers ($r > 0$) consistently improves performance over the unfiltered baseline ($r = 0$, **63.6%**). Optimal performance occurs at $r = 0.1$ with Macro F1 of **74.1%** (Setup A3), suggesting a critical balance is required while perfomring outlier removal.

Under-regularization ($r = 0$): Without filtering, $\mathcal{M}_s$ learns overly specific patterns from noisy $\mathcal{M}_l$ outputs, reducing generalization (**63.6%** vs **74.1%** at $r = 0.1$). For the *"Ack. Issue"* question, the framework includes specific phrases appearing in few training examples, causing $\mathcal{M}_s$ to reject valid acknowledgments with different wording (Appendix K). Setting $r = 0.1$ removes such phrases while preserving core evaluation criteria.

Over-regularization ($r \geq 0.25$): Excessive filtering removes useful information along with with noise. Performance drops to **69.1%** (vs **74.1%** at $r = 0.1$) as the model loses valid but uncom-

Setup	CoT	Temp	Outlier Removal (r)				Avg. Iter.
			0	0.1	0.25	0.5	
A0	No	0	63.4%	69.4%	68.4%	68.8%	6.35
A1	Yes	0	65.1%	73.1%	70.1%	69.6%	
A2	No	1	61.9%	71.8%	64.5%	67.8%	2.50
A3	Yes	1	63.6%	**74.1%**	69.1%	71.5%	

Table 4: Systematic ablation reveals that Chain-of-Thought **(CoT)** reasoning consistently enhances performance, while **Temperature** $\tau = 1$ accelerates convergence by **60%** (2.50 vs. 6.35 iterations). The optimal configuration (Setup A3, $r = 0.1$) leverages both to achieve the highest Macro F1 of **74.1%**.

mon reasoning patterns. For "Willingness to Assist" (Appendix L), by removing too much information, the framework lost useful refinements and reverted to generic instructions similar to the baseline.

8 Related Work

Recent advancements in large LM utilization have shifted from manual engineering to algorithmic optimization. Foundational frameworks like *Instruction Induction* (Honovich et al., 2023) and APE (Zhou et al., 2023) demonstrated LLMs' ability to generate and score instructions, while subsequent iterative methods like OPRO (Yang et al., 2023), Promptbreeder (Fernando et al., 2024), and PRESTO (Chu et al., 2025) introduced trajectory-based, evolutionary, and preimage-informed optimization strategies. To address the high computational cost of execution-based validation, recent approaches have pivoted to geometric and gradient-free methods. Notably, Chowdhury et al. (2026) proposed a validation-free centroid ranking in embedding space, while Pryzant et al. (2023) and Choi (2025) introduced textual gradients and confusion-matrix-based feedback (APO-CF) to refine prompts without overfitting to majority classes. Our work extends this by employing "Backward Questions" to explicitly reconstruct latent misunderstandings rather than relying on statistical aggregates.

Applying these methods to Contact Center Quality Assurance introduces unique challenges regarding class imbalance and subjective criteria (Roy et al., 2016; Henning et al., 2023). While Ingle et al. (2024) demonstrated that "Plan-Guided" prompting can bridge reasoning gaps in smaller models, and STREAQ (Sood et al., 2025) proposed tiered routing for cost-efficiency, deployment remains constrained by privacy standards. Research by Neupane et al. (2025) and Rahman et al. (2025) emphasizes the necessity of HIPAA-compliant architectures and PII-aware embeddings. This motivates our focus on thought distillation methods that can work with nuances of both objective and subjective quality assurance questions.

9 Conclusion

We present Backward Question-based Refinement (BQR), a diagnostic framework that leverages backward reasoning to systematically distill implicit domain knowledge into explicit evaluation criteria. Our experiments demonstrate that BQR achieves substantial performance improvements across 8 out of 12 contact center QA questions, with gains up to 27.8% (refer question *"Focus Resol."*) absolute Macro F1 while exhibiting additive, subtractive, and substitutive refinement strategies. We establish empirical parallels to gradient-descent optimization, showing outlier removal functions as regularization and generation temperature as learning rate. Our findings reveal a cross-family advantage, where small LMs benefit more from optimization by large LMs of different families, suggesting architectural diversity helps correct rather than reinforce existing biases. These results confirm BQR as an effective framework for operationalizing tacit expert knowledge in automated evaluation systems while fostering sustainable NLP research by reducing carbon footprint through deployment of computationally-efficient LMs at scale.

Ethical Considerations

The development and deployment of the Backward Question-based Refinement (BQR) framework involve several ethical dimensions and inherent limitations that warrant careful consideration.

- **Data Privacy and Reproducibility:** This research utilizes a proprietary corpus of real-world English dyadic conversations from multiple business domains. While these datasets are essential for capturing domain-specific nuances, the strictly proprietary nature of the data, combined with the presence of sensitive customer information, explicitly prevents its public release. For real-world deployment, ethical implementation requires adherence to strict privacy protocols, such as HIPAA-compliant architectures and PII redaction, to ensure that sensitive data remains protected throughout the iterative refinement process.

- **Human-in-the-Loop and Subjectivity:** BQR is designed as a tool to augment, rather than autonomously replace, human quality assurance. While the framework successfully formalizes the tacit knowledge of experts into explicit criteria, it remains inherently constrained by the subjectivity of its training signal. Since our ground-truth labels originate from a consensus of five domain experts, the optimized plans ($\mathcal{P}^*$) potentially codify the collective biases or specific organizational interpretations of that panel. Consequently, we emphasize that BQR outputs should be treated as high-fidelity recommendations subject to periodic human audit to ensure criteria remain fair, inclusive, and aligned with evolving organizational standards.

- **Computational and Environmental Footprint:** The iterative refinement process requires repeated inferences from high-parameter large language models ($\mathcal{M}_l$) such as `nova-pro` or `gpt-5`, which carry a significantly higher carbon footprint than standard, single-pass inference. While BQR promotes long-term sustainability by enabling high-volume deployment on energy-efficient small models ($\mathcal{M}_s$), the environmental benefit is amortized. The substantial energy consumption of the initial optimization phase is only ethically and practically justified for large-scale applications where downstream efficiency gains outweigh the upfront computational costs. In smaller deployments, even a modest one-time optimization cost may not amortize as favorably.

- **Interpretability:** BQR provides an ethical advantage by converting implicit model judgments into explicit, auditable evaluation plans, enhancing the interpretability of automated assessments. However this is subject to the generalization ability of the larger model. If the larger model is incapable of surfacing these nuances, organizations can run into the risk of relying on sub optimal plans and interpretations

- **Diagnostic Reliability and Domain Scope:** BQR relies on statistical outlier removal to filter diagnostic signals. While this prevents the distillation of idiosyncratic hallucinations into automated evaluation criteria, it assumes that systematic errors are more frequent than rare edge cases. Consequently, the framework's reliance on majority-driven refinement may overlook critical but infrequent reasoning patterns. Furthermore, while effective for contact center soft skills, further study is required to evaluate BQR's ethical and technical reliability in highly technical domains like medical or legal auditing.

References

Justin Chih-Yao Chen, Zifeng Wang, Hamid Palangi, Rujun Han, Sayna Ebrahimi, Long T. Le, Vincent Perot, Swaroop Mishra, Mohit Bansal, Chen-Yu Lee, and Tomas Pfister. 2025. Reverse thinking makes llms stronger reasoners. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2025 - Volume 1: Long Papers, Albuquerque, New Mexico, USA, April 29 - May 4, 2025*, pages 8611–8630. Association for Computational Linguistics.

Jaekeol Choi. 2025. Efficient prompt optimization for relevance evaluation via llm-based confusion matrix feedback. *Applied Sciences*, 15(9).

Md. Faisal Mahbub Chowdhury, Nandana Mihindukulasooriya, Niharika S. D'Souza, Horst Samulowitz, Neeru Gupta, Tomasz Hanusiak, and Michal Kapitonow. 2026. Automatic prompt engineering with no task cues and no tuning. *CoRR*, abs/2601.03130.

Jaewon Chu, Seunghun Lee, and Hyunwoo J. Kim. 2025. PRESTO: preimage-informed instruction optimization for prompting black-box LLMs. *CoRR*, abs/2510.25808.

Chrisantha Fernando, Dylan Banarse, Henryk Michalewski, Simon Osindero, and Tim Rocktäschel. 2024. Promptbreeder: Self-referential self-improvement via prompt evolution. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, volume 235 of *Proceedings of Machine Learning Research*, pages 13481–13544. PMLR / OpenReview.net.

Sophie Henning, William Beluch, Alexander Fraser, and Annemarie Friedrich. 2023. A survey of methods for addressing class imbalance in deep-learning based natural language processing. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2023, Dubrovnik, Croatia, May 2-6, 2023*, pages 523–540. Association for Computational Linguistics.

Or Honovich, Uri Shaham, Samuel R Bowman, and Omer Levy. 2023. Instruction induction: From few examples to natural language task descriptions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, pages 1935–1952.

Digvijay Ingle, Aashraya Sachdeva, Surya Prakash Sahu, Mayank Sati, Cijo George, and Jithendra Vepa. 2024. Probing the depths of language models' contact-center knowledge for quality assurance. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: EMNLP 2024 - Industry Track, Miami, Florida, USA, November 12-16, 2024*, pages 790–804. Association for Computational Linguistics.

Weisen Jiang, Han Shi, Longhui Yu, Zhengying Liu, Yu Zhang, Zhenguo Li, and James T. Kwok. 2024. Forward-backward reasoning in large language models for mathematical verification. In *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, volume ACL 2024 of *Findings of ACL*, pages 6647–6661. Association for Computational Linguistics.

Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. 2008. Isolation forest. In *Proceedings of the 8th IEEE International Conference on Data Mining (ICDM 2008), December 15-19, 2008, Pisa, Italy*, pages 413–422. IEEE Computer Society.

J. B. MacQueen. 1967. Some methods for classification and analysis of multivariate observations. *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, 1(14):281–297.

Subash Neupane, Shaswata Mitra, Sudip Mittal, and Shahram Rahimi. 2025. Towards a HIPAA compliant agentic AI system in healthcare. *CoRR*, abs/2504.17669.

Reid Pryzant, Dan Iter, Jerry Li, Yin Tat Lee, Chenguang Zhu, and Michael Zeng. 2023. Automatic prompt optimization with "gradient descent" and beam search. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 7957–7968. Association for Computational Linguistics.

Md. Abdur Rahman, Abdul Barek, ABM Kamrul Islam Riad, Md. Mostafizur Rahman, Md Bajlur Rashid, Md Raihan Mia, Hossain Shahriar, Guillermo Francia III, Fan Wu, Alfredo Cuzzocrea, and Sheikh Iqbal Ahamed. 2025. Embedding with large language models for classification of HIPAA safeguard compliance rules. In *49th IEEE Annual Computers, Software, and Applications Conference, COMPSAC 2025, Toronto, ON, Canada, July 8-11, 2025*, pages 1040–1046. IEEE.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 3980–3990. Association for Computational Linguistics.

Shourya Roy, Ragunathan Mariappan, Sandipan Dandapat, Saurabh Srivastava, Sainyam Galhotra, and Balaji Peddamuthu. 2016. Qa^{rt}: A system for real-time holistic quality assurance for contact center dialogues. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA*, pages 3768–3775. AAAI Press.

Prajwal Sood, Rajdeep Agrawal, Mayank Sati, Digvijay Ingle, and Cijo George. 2025. STREAQ: selective tiered routing for effective and affordable contact center quality assurance. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, EMNLP 2025 - Industry Track, Suzhou, China, November 4-9, 2025*, pages 1711–1726. Association for Computational Linguistics.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V. Le, Denny Zhou, and Xinyun Chen. 2023. Large language models as optimizers. *CoRR*, abs/2309.03409.

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

Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023. OpenReview.net.

Appendix

A Qualitative Examples for $\mathcal{Q}'$

Good Backwards Questions:

- **High-Specificity**: "Does the agent use a technical code instead of a natural language explanation?"
- **Behavioral Anchor Identification**: "Did the agent explicitly state that they will look into the customer's issue?"
- **Objective**: "Does the agent identify the customer account by their address"

Bad Backwards Questions:

- **Redundant/In-Plan**: "Did the agent acknowledge the customer's issue?"
- **Noisy/Hallucinated**: "Did the agent mention the blue sky during the billing discussion?"
- **Non-Atomistic/Compound**: "Did the agent greet the customer and then solve their problem and then ask if there was anything else?"
- **Overly Subjective**: "Was the agent nice?"

B Text Normalization and Clustering Pre-processing

As detailed in Step 3 of the BQR framework (Section 3.1), the raw backward questions generated by the large language model ($\mathcal{M}_l$) often contain repetitive conversational artifacts or structural boilerplate (e.g., *"Based on the provided transcript, did the agent..."* or *"...please answer with yes or no"*). These common prefixes and suffixes act as noise that can dominate the semantic embedding space, leading to clusters based on syntax rather than intent.

To mitigate this, we employ a *Common Substring Reduction* algorithm as part of the normalization function NORMALIZE$(\cdot)$. This procedure identifies and strips the longest common prefix and suffix shared across the batch of generated questions $\mathcal{Q}'$ prior to vectorization.

B.1 Steps for Text Normalization(NORMALIZE$(\cdot)$)

Let $\mathcal{Q}' = \{q'_1, q'_2, \ldots, q'_N\}$ be the set of raw backward questions generated in iteration t. The normalization process proceeds as follows:

- **Prefix Identification:** We compute the longest common prefix ρ shared by all sequences in $\mathcal{Q}'$.
- **Suffix Identification:** We compute the longest common suffix σ shared by all sequences in $\mathcal{Q}'$.
- **Heuristic Filtering:** To prevent the removal of short, meaningful linguistic markers (e.g., "Is", "Did"), we strictly enforce a length threshold δ (set to 2 characters).
- **Reduction:** For each question q'_i, we remove ρ and σ only if they satisfy the length threshold.
- **Safety Fallback:** If the resulting normalized string $\tilde{q}'_i$ is overly short (length < 3), suggesting over-truncation, we revert to the original q'_i.

The formal procedure is outlined in Algorithm 1.

Algorithm 1: Common Sub-string Reduction (Text Norm)

Input: Batch of raw backward questions $\mathcal{Q}' = \{q'_1, \ldots, q'_N\}$
Output: Normalized questions $\tilde{\mathcal{Q}}'$

$\rho \leftarrow \texttt{LongestCommonPrefix}(\mathcal{Q}')$; `// Identify shared opening phrasing`
$\sigma \leftarrow \texttt{LongestCommonSuffix}(\mathcal{Q}')$; `// Identify shared closing phrasing`
$\tilde{\mathcal{Q}}' \leftarrow \emptyset$;
foreach $q' \in \mathcal{Q}'$ **do**
  $text \leftarrow q'$;
  `// Remove Prefix if significant`
  if $\texttt{Len}(\texttt{Strip}(\rho)) > 2$ ***and*** *text starts with* ρ **then**
    $text \leftarrow text[\texttt{Len}(\rho) :]$;
  `// Remove Suffix if significant`
  if $\texttt{Len}(\texttt{Strip}(\sigma)) > 2$ ***and*** *text ends with* σ **then**
    $text \leftarrow text[: -\texttt{Len}(\sigma)]$;
  `// Safety Fallback to prevent information loss`
  if $\texttt{Len}(\texttt{Strip}(text)) < 3$ **then**
    $text \leftarrow q'$;
  $\tilde{\mathcal{Q}}' \leftarrow \tilde{\mathcal{Q}}' \cup \{\texttt{Strip}(text)\}$;
return $\tilde{\mathcal{Q}}'$;

C BQR Algorithm Details

This appendix provides a formal breakdown of the Backward Question-based Refinement (BQR) framework presented in Algorithm 2. The framework operates as an iterative optimization loop designed to distill implicit reasoning from a Large

Algorithm 2: Backward Question-based Refinement (BQR)

In : $\mathcal{Q}, \mathcal{P}^{(0)}, \mathcal{D}^{tr/vl/ts}, \mathcal{M}_s, \mathcal{M}_l$
Out : Optimized Plan $\mathcal{P}^*$

$\mathcal{P}^* \leftarrow \mathcal{P}^{(0)}; \Psi_{best} \leftarrow -\infty; t \leftarrow 0;$
while $t < T_{max}$ ***and*** *not converged* **do**
  `// Step 1: Forward Evaluation`
  $\forall(\mathcal{C}_i, \mathcal{A}_i^*) \in \mathcal{D}^{tr} : (\mathcal{R}_i^{(t)}, \mathcal{A}_i^{(t)}) \leftarrow$ $\texttt{SmallLMEval}(\mathcal{Q}, \mathcal{P}^{(t)}, \mathcal{C}_i);$
  `// Step 2: Backward Question Gen`
  $\mathcal{Q}' \leftarrow \emptyset;$
  foreach *sample* i **do**
    $q_i' \leftarrow \texttt{BackwardGen}(\mathcal{C}_i, \mathcal{R}_i^{(t)}, \mathcal{A}_i^{(t)}, \mathcal{P}^{(t)});$
    $\mathcal{Q}' \leftarrow \mathcal{Q}' \cup \{\texttt{Norm}(q_i')\};$
  `// Steps 3 & 4: Aggregation & Refinement`
  $C^{(t)} \leftarrow \texttt{Cluster}(\texttt{Filter}(\mathcal{Q}', r), k);$
  foreach $C_j \in C^{(t)}$ **do**
    $\psi_j^{(t)} \leftarrow \texttt{Impact}(\mathcal{M}_s, \mathcal{P}^{(t)}, \text{Inst}(C_j));$
  $C_{pos}, C_{neg} \leftarrow \texttt{Select}(C^{(t)}, \{\psi_j^{(t)}\}, n, m);$
  $\mathcal{P}^{(t+1)} \leftarrow$ $\texttt{Refine}(\mathcal{Q}, \mathcal{P}^{(t)}, \texttt{Synth}(C_{pos}), \texttt{Synth}(C_{neg}));$
  `// Step 5: Validation`
  $\Psi_{vl} \leftarrow \texttt{Eval}(\mathcal{M}_s, \mathcal{P}^{(t+1)}, \mathcal{D}^{vl});$
  if $\Psi_{vl} > \Psi_{best}$ **then**
    $\Psi_{best} \leftarrow \Psi_{vl}; \mathcal{P}^* \leftarrow \mathcal{P}^{(t+1)};$
  if *Patience Exceeded* **then break**;
  $t \leftarrow t + 1;$
return $\mathcal{P}^*$ *evaluated on* $\mathcal{D}^{ts}$;

Language Model ($\mathcal{M}_l$) into explicit evaluation criteria for a Small Language Model ($\mathcal{M}_s$).

D Label Distribution

Table 5 shows the distribution of labels for each question.

E Classification of Small vs Large LM

For all LMs used in this work, we lack visibility into their parameter counts. Consequently, we use API pricing from Microsoft Azure[3] and Amazon Bedrock[4] as of February 14, 2026 as a proxy to classify models within each family as small versus large. We emphasize that this classification is *relative within model families* rather than absolute. For instance, we designate `claude-4.5-haiku` as small despite its higher pricing than `nova-pro`, as it represents the smaller variant within the Claude-4.5 family.

[3]https://azure.microsoft.com/en-us/pricing/details/azure-openai/

[4]https://aws.amazon.com/bedrock/pricing/

F Prompt Templates

In this section, we provide prompt templates for various tasks mentioned in the work.

F.1 QA Evaluation Task

We use the same prompt as given in (Sood et al., 2025) for the QA evaluation task and is given in Figure 2.

```
You are a seasoned expert in quality assurance
for customer support conversations.

You are presented with an evaluation question
intended to assess an agent's performance
during a customer conversation. This
question is broken down into sub-criteria to
ensure a thorough and structured analysis.
Alongside, you are also provided with the
full dialogue between the customer and the
agent. Extract relevant evidence for each
sub-point, synthesize these observations
into a clear rationale, and conclude with
your final answer. Below are the required
information pieces for your task

1. Main question: {{question}}
2. Sub-criteria: {{plan}}
3. Conversation transcript: {{transcript}}
4. Answer options: ['yes','no']

To answer the given question, let's think
step by step:

Evidences:
(List evidences for each sub-criterion)
Synthesis:
(Summarize your reasoning)
Hence, the final answer is: (Your chosen
answer)
```

Figure 2: Prompt used for QA Evaluation Task defined in Section 2.

F.2 Generation of Backward Questions

To diagnose the model's internal interpretation, our primary framework utilizes the explicit Chain-of-Thought reasoning trace while generating backward questions, as shown in Figure 3. For our ablation studies evaluating the impact of this reasoning trace (detailed in Section 7.1), we employ an alternative prompt that relies solely on the final predicted answer, displayed in Figure 4.

F.3 Representative Question Synthesis

Following semantic clustering, we distill each cluster of backward questions into a single, cohesive query. The prompt instructing the model to syn-

Short Name	Evaluation Question	*no*	*yes*	Total
Address Name	Did the agent call the customer by their name at least thrice during the conversation?	155	90	245
Disclose PII	During the call, did the agent take the initiative to state the customer's phone number, name, date of birth, address, social secruity number?	155	25	180
Positive Tone	Assess whether the customer's overall experience with the agent was positive, considering their tone, the quality of the interaction, and how the call ended independent of the actual resolution outcome.	90	109	199
Willing Assist	Did the agent indicate their readiness to help the customer?	127	47	174
Natural Style	Did the agent communicate in a way that felt natural and engaging?	135	25	160
Ack. Issue	Did the agent acknowledge the customer's issue?	94	75	169
Clear Disconnect	Assess whether the agent clearly informed the customer that the call would be ended after they became unresponsive, specifically looking for statements that signal the agent's intention to disconnect due to no reply.	105	106	211
Demo Underst.	Did the agent demonstrate that they understood the customer's issue?	36	136	172
Avoid Assist	Did the agent refrain from providing assistance to the customer?	116	81	197
Gather Info	Did the agent gather information required to investigate?	70	98	168
Focus Resol.	Did the agent keep control of the conversation and maintain a clear focus on the resolution?	86	100	186
Ask Repeat	Did the agent request the customer to repeat themselves?	177	59	236

Table 5: Distribution of labels for the dataset. Columns represent the count of *no* and *yes* labels for each question and total examples.

```
You are an expert diagnostic assistant. You
are provided with a conversation transcript
and a model's reasoning trace that concludes
in a specific answer.

Your goal is to reverse-engineer the model's
interpretation by generating a "backward
question". Generate a diverse set of
questions that are NOT currently covered
by the existing evaluation plan, for which
the provided reasoning and answer would be
considered a logically correct response.

Below are the inputs for your task:

1. Conversation Transcript: {{transcript}}
2. Model Reasoning: {{reasoning}}
3. Model Answer: {{answer}}
4. Current Evaluation Plan: {{plan}}

Based on the logic shown in the reasoning,
what specific question was the model actually
answering?

Requirements:
- The generated questions must yield the
provided answer based on transcript.
- Ensure the questions are semantically
distinct from those in plan.
- Provide a diverse set of interpretations.
- Each generated question should be atomistic,
objective, and self-contained.

Backward Questions:
(Generate the list of backward questions here)
```

Figure 3: The prompt used for the Backward Question Generation to probe small model's internal interpretation and conceptual understanding of question by leveraging it's Chain-of-Thought reasoning trace along with predicted answer.

```
You are an expert diagnostic assistant. You
are provided with a conversation transcript
and a model's final predicted answer.

Your goal is to reverse-engineer the model's
interpretation by generating a "backward
question". Generate a diverse set of
questions that are NOT currently covered
by the existing evaluation plan, for which
the provided answer would be considered
a logically correct response based on the
transcript.

Below are the inputs for your task:

1. Conversation Transcript: {{transcript}}
3. Model Answer: {{answer}}
4. Current Evaluation Plan: {{plan}}

Based on the provided answer, what specific
question was the model actually answering?

Requirements:
- The generated questions must yield the
provided answer based on transcript.
- Ensure the questions are semantically
distinct from those in plan.
- Provide a diverse set of interpretations.
- Each generated question should be atomistic,
objective, and self-contained.

Backward Questions:
(Generate the list of out-of-plan questions
here)
```

Figure 4: The prompt used for the Backward Question Generation to probe small model's internal interpretation and conceptual understanding of question using only the predicted answer. Used for setups A0, A2 mentioned in Section 7.1

Model	Price ($) per 1M Tokens	
	Output	Input
nova-lite	0.06	0.24
nova-pro	0.80	3.20
nova-premier	2.50	12.50
gpt-5-nano	0.05	0.40
gpt-5	1.25	10.00
claude-4.5-haiku	1	5
claude-4.5-sonnet	3	15

Table 6: Pricing for API usage.

thesize this representative question is provided in Figure 5.

```
You are given a list of semantically similar questions.
Your task is to write a single clear and concise question that best represents their shared meaning.

Questions:{{question_set}}

Rules:
- Avoid repetition.
- Do not add new information.
- Output ONLY the final question.

Representative Question: (Synthesize the representative question here)
```

Figure 5: The prompt used for Representative Question Synthesis.

F.4 Plan Synthesis

Finally, the large language model refines the evaluation guidelines by contrasting the successful and failed interpretation clusters. The meta-prompt used to generate this updated evaluation plan is detailed in Figure 6.

G Refinement Walkthroughs

This appendix provides compact end-to-end walkthroughs for the three refinement patterns discussed in Section 5.2. Each walkthrough traces the transition from the initial plan to the optimized plan through the dominant diagnostic pattern surfaced by backward questions.

Additive Walkthrough: Willing Assist

The initial *"Willing Assist"* plan in Appendix J contained only two short, high-level criteria about helping and empathy. The dominant backward-question pattern instead centered on concrete commitment signals, such as whether the agent outlined

```
You are an expert in designing evaluation guidelines. Your task is to refine an evaluation plan into precise instructions for a QA analyst.

Original Question: {{question}}

Current Plan: {{current_plan}}

Positive Interpretations: {{pos_questions}}

Negative Interpretations: {{neg_questions}}

Instructions:
- Analyze the negative interpretations to identify ambiguities, and rewrite the plan to explicitly prevent these misalignments.
- Preserve and reinforce the logic that guided the positive interpretations.
- Structure the plan as actionable evaluation criteria, rather than a simple list of sub-questions.
- Explicitly define when to use these criteria as a set of logical AND / OR conditions to conclude with a definitive YES or NO answer to the Original Question.
- Output the updated plan strictly as a JSON list where each element represents a criterion. Do not output any other text or markdown.

Refined Plan: (Generate the refined plan in JSON format here)
```

Figure 6: The meta-prompt used for Plan Synthesis. Large LM analyzes positive and negative interpretations to generate an updated plan.

next steps, personally committed to an action, or gave the customer confidence that assistance would follow. BQR therefore expanded the plan with explicit behavioral anchors that operationalize willingness through observable commitments rather than vague sentiment. This additive refinement increased Macro F1 from **42.9%** to **62.9%**.

Subtractive Walkthrough: Positive Tone

For *"Positive Tone,"* the original plan in Appendix J was already detailed, but it mixed core evaluation logic with formatting and reader-oriented scaffolding. The dominant diagnostic pattern emphasized the customer's final affective outcome and the agent behaviors most relevant to that outcome, without needing the surrounding instructional overhead. BQR therefore consolidated the plan into denser semantic criteria while preserving its core decision rule. This subtractive refinement improved Macro F1 from **45.2%** to **56.9%** without materially increasing plan length.

Substitutive Walkthrough: Address Name

For *"Address Name,"* the initial plan in Appendix J captured the broad intent but left the temporal scope and decision rule underspecified. The diagnostic signal consistently pointed toward questions about identifying the caller's name from the start of the interaction and using repeated name usage as a precise yes/no criterion. BQR responded by replacing generic phrasing with operational constraints about where to identify the name and how many uses suffice for a positive label. This substitutive refinement improved Macro F1 from **59.9%** to **65.7%**.

H Additional Deployment Results

LLM-Generated Labels: When we replace expert labels with LLM-generated labels during refinement, BQR improves average Macro F1 from **64.41%** to **73.03%**, with gains on **7/12** questions. The resulting failure patterns remain similar to those observed under expert supervision, suggesting that the framework can still identify useful refinement signals in more automated pipelines.

Cost Perspective: Using the costing methodology of STREAQ (Sood et al., 2025), upgrading `nova-lite` to `nova-pro` for full inference yields a much larger daily cost increase than refining the plan once offline and continuing to serve with `nova-lite`. In our setting, the one-time offline optimization cost is approximately $2 per plan, which amortizes quickly in high-volume deployments.

I Scalability, Generalizability, and Automated Labels

BQR is designed so that the expensive use of $\mathcal{M}_l$ occurs offline during refinement and the optimized plan $\mathcal{P}^*$ is then deployed with the cheaper $\mathcal{M}_s$ at inference time. Using the costing methodology of STREAQ (Sood et al., 2025), the one-time optimization cost in our setting is approximately $2 per plan, while the resulting deployment overhead remains far smaller than replacing $\mathcal{M}_s$ with a larger model at full inference scale. We also observe that BQR remains effective with LLM-generated labels, improving average Macro F1 from **64.41%** to **73.03%** and improving **7/12** questions, which suggests the method can operate in more automated pipelines.

J Full Plan Refinement Comparisons

This section details the evolution of instructions for each question for the results presented in the main body of the paper.$P^{(0)}$ represents the initial plan provided to the small LM while the P^* represents the converged plan after BQR optimization.

1. Address Name

Original Plan ($P^{(0)}$):

- The agent used the caller's name at least thrice any time during the call. Look into the complete call.
- Determine if the agent addressed the caller by their name at least two times. Provide a 'Yes' or 'No' answer, and list the exact phrases where the agent used the caller's name."

(Length: 51 words)

Optimized Plan (P^*):

- Identify the caller's name from the beginning of the call transcript.
- Determine if the agent addressed the caller by their name at least two times throughout the call.
- Use the criteria of the agent addressing the caller by name at least two times as a condition to conclude with a 'Yes' or 'No' answer.
- Did the agent address the caller by name minimum 2 times on the call?

(Length: 68 words)

2. Disclose PII

Original Plan ($P^{(0)}$):

- Determine whether the agent proactively disclosed any of the following customer information before the caller provided or verified it: Phone Number Full Name Date of Birth (DOB) Address Social Security Number
- The agent must not read out or state this information first. Instead, the agent should: Ask the customer to provide the information voluntarily (e.g., 'Can you confirm your date of birth?'), Only confirm it after the customer states it, and Never initiate the disclosure of this data.
- Provide: A 'Yes' or 'No' answer: Did the agent improperly disclose any of the listed information proactively? A brief justification, including the exact phrases used and timestamps (if available). A list of any violations found, matched with the type of information disclosed."*

(Length: 120 words)

Optimized Plan (P^*):
[No Change] The optimizer retained the original plan instructions.
(Length: 120 words)

3. Positive Tone

Original Plan ($P^{(0)}$):

- Scoring Logic (Binary Output) Yes (1): Customer had a positive or neutral experience, with no signs of dissatisfaction or rejection at the end of the call. No (0): Customer showed clear dissatisfaction, frustration, or rejection of the interaction or outcome.
- Agent actions that may contribute to a positive experience (use as supporting factors, not mandatory): The agent deescalates emotional situations The agent modifies tone to be supportive or empathetic The agent goes the extra mile (e.g., provides thorough explanations, reassures the caller) The agent uses courteous or empathetic language
- Customer shows signs of satisfaction, relief, or gratitude Neutral tone with no objection to the outcome or closing Explicit acceptance of the resolution, regardless of whether the issue was solved Absence of complaints, pushback, or dissatisfaction
- The customer does not show any sign of dissatisfaction or rejection at the end of the call, do not mark it against the agent, even if the agent does not explicitly demonstrate positive behaviors such as deescalating, tone-shifting, or using courtesy phrases.

(Length: 170 words)

Optimized Plan (P^*):

- Evaluate the customer's tone throughout the call for signs of satisfaction, relief, or gratitude, and there is an absence of complaints, pushback, or dissatisfaction.
- Assess the quality of interaction by identifying if the agent deescalated emotional situations, modified tone to be supportive, and used courteous or empathetic language.
- Determine if the customer showed explicit acceptance of the resolution or had a neutral tone with no objection to the outcome, regardless of whether the issue was resolved.
- Conclude with a yes or no answer based on the criteria that the customer did not show any sign of dissatisfaction or rejection at the end of the call, even if the agent did not explicitly demonstrate positive behaviors.
- Based on the customer's tone and expressions during the call, did the interaction quality align with the criteria for a positive and cooperative experience?
- Analyze the customer's tone and specific phrases throughout the call to determine whether they expressed satisfaction, relief, or gratitude towards the resolution, despite any initial dissatisfaction or negative emotions.

(Length: 168 words)

4. Willing Assist

Original Plan ($P^{(0)}$):

- The agent gives the user confidence that they will assist them to the best of their ability.
- If the customer is angry, the agent demonstrates empathy with their situation.

(Length: 29 words)

Optimized Plan (P^*):

- Does the agent's language and tone convey a positive and reassuring commitment to helping the customer?
- Did the agent outline specific actions or steps they would take to assist the customer?
- Did the agent either personally commit to taking specific actions or provide a detailed plan to resolve the customer's issue?
- How did the agent initially acknowledge the customer's inquiry or concern?
- Considering the above criteria as a set of 'and' conditions, did the agent express willingness to assist the customer?

(Length: 81 words)

5. Natural Style

Original Plan ($P^{(0)}$):

- Agent's communication felt natural and engaging rather than scripted, or overly formal.
- Look for personalized responses, conversational flow, and appropriate tone adjustments based on the customer's response.

(Length: 27 words)

Optimized Plan (P^*):

- Evaluate if the agent's communication felt natural and engaging, as opposed to being scripted, robotic, generic, or overly formal.
- Assess whether the agent's responses were personalized, demonstrated a conversational flow, and appropriately adjusted tone based on the customer's cues.
- Determine if there were any specific characteristics missing from the agent's communication that prevented it from sounding natural and engaging, and analyze how this impacted the overall effectiveness of their interaction with the customer.
- Identify specific elements in the agent's response that either enhance or detract from a natural and engaging communication style, and provide examples to support your assessment.
- Conclude with a yes or no answer to the question: Did the agent's communication style sound natural and engaging? (Use the above criteria as a set of and/or conditions to reach your conclusion.)

(Length: 132 words)

6. Ack. Issue

Original Plan ($P^{(0)}$):

- The agent clearly acknowledged and restated the customer's issue or concern during the conversation.
- The agent actively listened and restated the customer's issue in their own words to confirm understanding using phrases like " the situation you are experiencing with" , "we regret that you were", " you are experiencing", "so you have reached out because", "so you are having issue with" etc

(Length: 63 words)

Optimized Plan (P^*):

- Identify if the agent restated the customer's issue or concern during the conversation.
- Determine if the agent used specific phrases to acknowledge the customer's issue, such as 'the situation you are experiencing with', 'we regret that you were', 'you are experiencing','so you have reached out because', or'so you are having issue with'.
- Assess if the agent's response indicates active listening and understanding of the customer's issue.
- Conclude with a yes or no answer based on the criteria that the agent clearly acknowledged and restated the customer's issue or concern, using specific phrases to confirm understanding.

(Length: 95 words)

7. Clear Disconnect

Original Plan ($P^{(0)}$):

- The agent states that because they are unable to hear from customer they are disconnecting closing the call. Example "ending the chat, closing the chat, end our chat, end the call, disconnect the call, disconnect the chat etc
- The agent states that because of no response from customer they are disconnecting the call or ending the chat or close the chat. Example "ending the chat, closing the chat, end our chat, end the call, disconnect the call, end this call, disconnect the chat, disconnect our chat etc
- This should not include instances where in the agent has transferred the call or suggests that they are going to transfer if they are placing the customer on hold

(Length: 116 words)

Optimized Plan (P^*):

- Assess whether the agent utilized a distinct phrase to signal the termination of the communication (call or chat) as a result of the customer's non-responsiveness.
- Evaluate if the agent clearly articulated that the disconnection of the call or chat was a direct consequence of the customer's failure to respond.
- Verify if the agent's communication incorporates expressions such as 'ending the chat', 'closing the chat', 'end our chat', 'end the call', 'disconnect the call', 'end this call', 'disconnect the chat', or 'disconnect our chat'.
- Ensure that the agent's decision to conclude the call or chat is solely attributed to the customer's lack of response, and not influenced by alternative factors such as call transfer suggestions, three-way call arrangements, or placing the customer on hold.
- Conclude with a yes or no answer based on the criteria that the agent provided a clear call disconnection disclosure when the customer stopped responding, using the above conditions as a set of and/ or requirements.

(Length: 159 words)

8. Demo Underst.

Original Plan ($P^{(0)}$):

- Review the initial customer inquiry and agent's response. If the issue is straightforward, the agent may proceed directly to addressing it without clarifying questions.
- Check if the agent's response directly addresses the specific customer concern, even if brief or templated. Explicit restatement of the issue is not required if the response shows relevant understanding
- Evaluate if the agent acknowledges the core issue, regardless of their ability to resolve it. Understanding can be demonstrated even when facing policy constraints, system limitations, or need for escalation.
- Mark YES if the agent shows understanding, even if: they can't resolve the issue, must redirect it, fail to provide satisfactory resolution, or cannot suggest solutions.
- Make final YES/NO determination based solely on demonstrated understanding, regardless of customer satisfaction or resolution outcome.

(Length: 126 words)

Optimized Plan (P^*):

- Review the initial customer inquiry to identify the specific issue or concern raised by the customer.
- Examine the agent's response to determine if it acknowledges or addresses the customer's specific issue, even if the response is brief or uses templated language.
- Assess whether the agent's response demonstrates an understanding of the core issue, irrespective of their ability to resolve it, redirect it, or suggest solutions.
- Consider the context of the interaction: if the issue is straightforward, direct addressing without clarifying questions may still indicate understanding.
- Use the following criteria as a set of AND conditions to conclude with a YES answer: the agent's response must show acknowledgment of the customer's issue, demonstrate understanding of the core concern, and address the issue directly or through relevant action, regardless of the resolution's satisfactoriness or the agent's ability to fully resolve the issue.
- Conclude with a NO answer if the agent's response does not demonstrate an understanding of the customer's issue, fails to acknowledge the core concern, or if the response is unrelated to the customer's inquiry.
- Make the final YES/NO determination based solely on the demonstrated understanding of the customer's issue, independent of customer satisfaction or the resolution outcome.

(Length: 197 words)

9. Avoid Assist

Original Plan ($P^{(0)}$):

- Analyze the conversation and determine if the agent avoided assisting the customer
- Look for indication where the agent may have failed to address the customer's concerns, dismissed their inquiries, or provided no straightforward help.
- Exclude interactions where customer is not present.

(Length: 41 words)

Optimized Plan (P^*):

- Did the agent consistently provide meaningful assistance and engage actively throughout the conversation to help resolve the customer's issues and inquiries?
- Were there any instances where the agent failed to address the customer's concerns or dismissed their inquiries?
- Did the agent exhibit behaviors such as providing no meaningful help, avoiding the customer's questions, or showing reluctance in resolving the issue?
- Using the criteria above as a set of and/or conditions, did the agent avoid assisting the customer?

(Length: 77 words)

10. Gather Info

Original Plan ($P^{(0)}$):

- Analyze the conversation and determine if the agent gathered necessary information required to resolve the issue
- Agent gathered detailed and relevant information from the customer to understand their issue.
- Agent made an effort to gather more information from the customer by probing them to provide additional details

(Length: 47 words)

Optimized Plan (P^*):
[No Change] The optimizer retained the original plan instructions.
(Length: 47 words)

11. Focus Resol.

Original Plan ($P^{(0)}$):

- Agent maintained control of the conversation and remained focused on leading the conversation toward a resolution
- Do not penalise the agent for probing, gathering account details or checking for customer's presence

(Length: 31 words)

Optimized Plan (P^*):

- Assess whether the agent consistently maintained control of the conversation throughout the call.
- Evaluate if the agent demonstrated focus on driving the interaction toward a resolution by probing for necessary information.
- Determine if the agent gathered relevant account details to assist in resolving the customer's issue.
- Check if the agent ensured the customer's presence and engagement during the call for effective communication.
- Identify any instances where the agent lost focus on the resolution or failed to maintain control of the conversation.
- Conclude with a yes or no answer based on the agent meeting all the above criteria as a set of and conditions to determine if they stayed on top of the call and did not lose focus of the resolution.

(Length: 121 words)

12. Ask Repeat

Original Plan ($P^{(0)}$):

- Analyze the interaction to determine if the agent asked the customer to repeat any information.
- Exclude instances wherein the agent explicitly requested clarification due to inaudibility, or unclear responses.
- Exclude instances wherein customer asking the agent to repeat themselves

(Length: 39 words)

Optimized Plan (P^*):

- Analyze the interaction transcript to determine if the agent explicitly requested the customer to repeat any information.
- Identify instances where the agent explicitly requested clarification due to inaudibility, unclear responses, or missing details, and use these as evidence to conclude if the agent asked the customer to repeat themselves.
- Exclude instances where the customer asked the agent to repeat themselves, as these do not contribute to determining if the agent asked the customer to repeat.
- Conclude with a yes or no answer based on the presence or absence of explicit requests from the agent for the customer to repeat information, using the identified instances as criteria.

(Length: 106 words)

K Effect of Under-Regularization

This appendix illustrates the detrimental effects of under-regularization ($r = 0$) on plan generalization, as discussed in Section 7. Without outlier removal, the BQR framework tends to overfit to the larger model's noisy feedback, incorporating overly specific linguistic artifacts and rigid logical constraints into the optimized plan.

Case Study: Acknowledge Customer's Issue

In the "Ack. Issue" question, the baseline plan provided a general instruction to look for acknowledgment. The under-regularized optimization process ($r = 0$) generated a plan that rigidly enforced the use of specific phrases and included idiosyncratic negative constraints observed in a small subset of the training data.

Under-Regularized Plan ($r = 0$):

- Identify if the agent clearly acknowledged and restated the customer's issue or concern during the conversation.
- Evaluate if the agent actively listened and restated the customer's issue in their own words to confirm understanding, using phrases like 'the situation you are experiencing with', 'we regret that you were', 'you are experiencing', 'so you have reached out because', 'so you are having issue with', etc.
- Determine if the agent's response included any specific phrases indicating acknowledgment of the customer's issue.
- **Check for any specific technical detail or reference provided by the agent instead of acknowledging the customer's issue.**
- Conclude with a yes or no answer based on the criteria that the agent both clearly acknowledged the issue and actively listened and restated the customer's concern (using the set of AND conditions), and there were no specific technical details or references provided instead of acknowledgment (using the set of OR conditions).

This restrictive plan caused the smaller model to penalize valid acknowledgments that utilized natural but distinct phrasing, or instances where an agent provided a technical solution immediately (which the plan penalized via the "technical detail" check). This led to a drop in Macro F1 score to **63.6%**.

Optimized Plan ($r = 0.1$): By applying a moderate outlier removal threshold ($r = 0.1$), the framework successfully filtered out the noisy negative constraint regarding "technical details" and simplified the boolean logic. The resulting plan maintained the useful examples of acknowledgment phrases but framed them as indicators rather than rigid requirements, allowing the smaller model to generalize better and achieving a Macro F1 of **74.1%**.

Optimized Plan ($r = 0.1$):

- Identify if the agent restated the customer's issue or concern during the conversation.
- Determine if the agent used specific phrases to acknowledge the customer's issue, such as 'the situation you are experiencing with', 'we regret that you were', 'you are experiencing','so you have reached out because', or 'so you are having issue with'.
- Assess if the agent's response indicates active listening and understanding of the customer's issue.
- Conclude with a yes or no answer based on the criteria that the agent clearly acknowledged and restated the customer's issue or concern, using specific phrases to confirm understanding.

L Effect of Over-Regularization

This appendix illustrates the impact of over-regularization ($r \geq 0.25$) on plan granularity. As discussed in Section 7, setting the outlier removal threshold too high eliminates valid reasoning patterns alongside noise. This results in over-simplified plans that lack the specific behavioral anchors required for small models to reason effectively.

Case Study: Willingness to Assist

In the "Willing Assist" question, the optimized framework ($r = 0.1$) discovered that the smaller model improved significantly when given a checklist of specific behaviors (e.g., greetings, proactive offers). However, when the outlier threshold was increased to $r = 0.5$, these specific refinements were statistically filtered out because they appeared in diverse but not identical forms across larger model traces. The resulting plan reverted to a generic, high-level instruction similar to the baseline.

Over-Regularized Plan ($r = 0.5$):

- Identify if the agent provided a clear statement or action indicating their commitment to help the customer, which gives the user confidence that they will assist them to the best of their ability.
- Determine if the agent showed empathy towards the customer's situation, especially in scenarios where the customer expressed anger or frustration.
- Conclude with a yes or no answer based on the fulfillment of both criteria above as a set of AND conditions to affirm the agent's expressed willingness to assist the customer.

This generic plan failed to guide the smaller model on *how* to identify "commitment" or "empathy," resulting in a performance drop similar to the original baseline (**47.4%**). The model struggled to map these abstract concepts to the text without explicit examples.

Optimized Plan ($r = 0.1$): With a moderate threshold ($r = 0.1$), the framework retained the granular behavioral markers. The plan explicitly defined "willingness" through observable actions, such as a "welcoming greeting" and "proactive solutions", providing the necessary scaffolding for the smaller model to achieve a high Macro F1 of **87.9%**.

Optimized Plan ($r = 0.1$):

- Identify if the agent used a welcoming and positive greeting to initiate the conversation, setting a tone of readiness to assist.
- Determine if the agent explicitly stated their intention to help resolve the customer's issue or answer their questions.
- Assess whether the agent used phrases that convey understanding and empathy towards the customer's situation or concerns.
- Evaluate if the agent offered specific assistance or solutions to the customer's problem, indicating a proactive approach to help.
- Conclude with a 'Yes' if the agent's communication throughout the conversation consistently demonstrated a clear willingness and readiness to assist the customer.