STREAQ: Selective Tiered Routing for Effective and Affordable Contact Center Quality Assurance

Prajwal Sood **Rajdeep Agrawal** **Mayank Sati** **Digvijay Ingle**
Cijo George
Observe.AI, India
{prajwal.sood, rajdeep.agrawal, mayank.sati, digvijay.ingle}@observe.ai
{cijo.george}@observe.ai

Abstract

Contact centers process millions of customer conversations daily, requiring Quality Assurance (QA) teams to evaluate agent performance against compliance and service standards, often by answering agent evaluation questionnaires. Traditional manual QA cannot scale to growing volumes, while fully automated evaluation using large language models presents a cost-performance trade-off. High-performing models excel at detecting rare but business critical Answers of Interest (AoI) but incur prohibitive costs, while smaller fine-tuned models are economical but suffer from poor AoI precision, generating high false positive rates that erode agent trust and waste QA resources. We introduce STREAQ, a two tier selective routing framework to intelligently route queries between cost efficient and high-capability models. Based on benchmarking on a proprietary dataset across six large LMs, STREAQ achieves substantial cost reduction while preserving critical performance. Using *Nova-Pro*, STREAQ reduces daily costs by 48% from $34,162 to $17,842 while retaining 88.9% of full model AoI precision. Our ablation studies reveal that flawed reasoning from smaller models can degrade performance, emphasizing the importance of carefully designing routing systems, making enterprise scale automated QA both practical and economically viable.

1 Introduction

Contact centers are vital to customer experience, handling millions of conversations that influence satisfaction, retention, and compliance. Within this ecosystem, Quality Assurance (QA) teams play a crucial role in evaluating these conversations to ensure agents follow protocols, meet performance standards, and uphold regulatory requirements. While manual QA is effective, it struggles to scale across high volumes. The convergence of contact center management and artificial intelligence marks a transformative shift in how organizations manage customer experience and operational quality (Roy et al., 2016). Large Language Models (LMs) such as *GPT-4* (OpenAI, 2023), *Claude* (Anthropic, 2023), and *Gemini* (Anil et al., 2023) offer a compelling solution by automating QA assessment with greater consistency and depth, but their enterprise-scale deployment introduces new technical and operational challenges.

Skewed QA Outcomes: Contact center QA teams typically rely on standardized questionnaire, that must be answered by analyzing conversation and assessing effectiveness of the agent's interaction with the customer (Ingle et al., 2024). These conversations often yield imbalanced QA outcomes, where some responses are overwhelmingly more frequent than others. For example, the question *"Did the agent greet the customer properly?"* usually receives a *yes* in well-trained centers, as greetings are routine. Instances where agents fail to greet, referred to as *Answer of Interest (AoI)*, are rare but critical, resulting in a minority of negative labels must be accurately detected. This imbalance reflects an operational reality: routine behaviors are consistently performed well, while key failures are infrequent but carry high business impact. For example, incorrectly flagging an agent for *"missed customer greeting"* (a false positive) triggers unnecessary compliance reviews, coaching, and erosion of agent trust. In such cases, a high AoI precision is more valuable than high overall Macro F1.

The Cost-Scale Dilemma: While large LMs such as *Claude-3.5-Sonnet* have demonstrated strong performance QA assessment task, their deployment at enterprise scale remains economically prohibitive (Ingle et al., 2024). Assuming a contact

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: Industry Track, pages 1711–1726
November 4-9, 2025

center processes **1.5M** conversations per day, each requiring evaluation on 10–20 QA questions, this translates to a daily cost of approximately **$160,500** (refer to Appendix A), rendering full-scale adoption of large LMs infeasible for most organizations. Ingle et al., 2024 has explored plan-guided fine-tuning as a promising strategy to narrow the performance gap between large and smaller models, improving cost-effectiveness. However, our empirical analysis reveals that performance on the AoI remains significantly lower in such setups.

We hypothesize that this performance degradation arises from the inherently skewed nature of contact center data. While methods like SMOTE (Chawla et al., 2002), SMOTEBoost (Chawla et al., 2003), and Balanced Random Forest (Leevy et al., 2018) improve minority class performance in traditional classification tasks, they are less effective here. Such techniques assume a consistent task between training and inference, where a model learns to map input features to fixed output categories. In contrast, contact center QA requires a model to learn a **generalizable reasoning process** rather than a fixed classification mapping, as the specific QA questions encountered at inference time may be entirely different. The core challenge is to teach the model to systematically identify evidences, synthesize them, and arrive at logical conclusions for unseen questions. Simply balancing the *yes*/*no* label distribution does not address this fundamental reasoning deficit, which is the primary cause of failure in smaller models (Ingle et al., 2024).

Inspired by prior work on multi-tier model routing and selective computation in NLP (Sordoni et al., 2023; Pan et al., 2023; Ding et al., 2024; Chen et al., 2023), we present an empirical study investigating the feasibility and effectiveness of a two-step evaluation framework for contact center QA. Specifically, we explore if selectively routing predictions from lightweight LMs to more capable LMs at inference time can enable a cost-effective yet reliable QA pipeline. Specifically, our contributions are as follows:

1. We propose a RoBERTa-based binary classifier that learns to identify low-confidence predictions from a smaller model and selectively routes them to a more capable LM

2. We provide a detailed cost-benefit analysis comparing selective routing with full-model inference, demonstrating the practical viability of our approach at enterprise scale

3. We establish how reasoning traces from smaller LMs can negatively impact larger LMs in a two-tier setup, and show that omitting such context improves both performance and efficiency, thus reinforcing the importance of careful design choice for routing system

While multi-tier inference has been widely explored as a paradigm, to the best of our knowledge, this is the first study to apply selective routing to contact center QA with a specific focus on improving AoI performance, proposing an effective routing design tailored for this setting.

2 Benchmarking Large LMs on AoI

This section aims to evaluate out-of-the-box (OOTB) performance of a suite of large LMs on the contact center QA assessment task, with a specific focus on improving performance on the Answer of Interest (AoI). The methodology is detailed below.

2.1 Data Curation

To evaluate OOTB performance of large LMs on contact center QA tasks, we curate a specialized dataset, denoted as $\mathcal{D}_{QA}$, consisting of binary QA questions to be answered with either `yes` or `no` based on agent-customer conversations.

We begin by selecting 10 representative QA questions from a proprietary contact center corpus, which span a diverse range of evaluation categories, including customer experience, compliance, resolution competence, process adherence, and professionalism. For each question, we sample English dyadic conversations from phone and chat channels, relevant to the evaluation criteria, resulting in an initial dataset of approximately 3,000 question–conversation pairs. We further employ sampling heuristics to ensure a meaningful distribution of both `yes` and `no` responses per question (see Appendix B), thereby supporting reliable performance estimates, particularly for the minority AoI.

To obtain high-quality ground truth, each question–conversation pair is independently annotated by five domain experts with extensive experience in contact center QA. Annotators are guided by a detailed annotation protocol that encourages structured reasoning: identifying relevant conversational evidence, synthesizing it, and determining the final answer, similar to the methodology outlined by Ingle et al., 2024. This process emulates

the judgmental reasoning followed by professional QA analysts and ensures annotation traceability.

To ensure reliability of $\mathcal{D}_{QA}$, we retain only those samples in which at least three annotators agreed on the final answer, producing a refined dataset of 1,879 question–conversation pairs (approximately 62% of the initial set), each labeled with the majority agreed answer and its supporting rationale[1].

2.2 Experimental Setup

We evaluate the OOTB performance of six large LMs, spanning three major model families, on the QA dataset $\mathcal{D}_{QA}$. This evaluation suite is denoted as: $\mathcal{M} = \{M_i \mid i \in \mathcal{I}\}$ where:

$$\mathcal{I} = \begin{Bmatrix} \texttt{nova-pro}, \\ \texttt{nova-premier}, \texttt{o3-mini}, \\ \texttt{gpt-4o}, \texttt{claude-3.5-haiku}, \\ \texttt{claude-4-sonnet} \end{Bmatrix}$$

Each LM $\mathcal{L} \in \mathcal{M}$ is prompted with a question-plan-conversation triplet $(\mathcal{Q}, \mathcal{P}, \mathcal{C})$, where $(\mathcal{Q}, \mathcal{C}) \in \mathcal{D}_{QA}$ and $\mathcal{P}$ is an evaluation plan, obtained from annotators in Section 2.1 detailing how they assessed agent behavior in response to the question while annotating $\mathcal{D}_{QA}$. The model is instructed to assume the role of a contact center QA specialist (Kong et al., 2024). The model is tasked with performing conversation-guided chain-of-thought (CoT) reasoning (Wei et al., 2022), involving three sequential steps: (1) identifying relevant evidences from $\mathcal{C}$ that pertain to $\mathcal{Q}$, (2) synthesizing these evidences into a coherent reasoning, and (3) concluding a final answer $\mathcal{A} \in \{yes, no\}$ as proposed by Ingle et al., 2024.

While Ingle et al. (2024) leverages the evaluation plan to distill reasoning from a large LM into a smaller one during training, we hypothesize that even large models can benefit from being explicitly guided by the plan at inference time. This allows their reasoning to align more closely with business-specific QA requirements, rather than relying solely on general world knowledge. Accordingly, we slightly adapt the prompting methodology for larger LMs by integrating the plan into the input, encouraging them to ground their responses per QA expectations.

Additionally, we also evaluate an in-house model fine-tuned on contact center data based on the paradigm described in Ingle et al., 2024 on $\mathcal{D}_{QA}$. This CoT-based prompting approach mirrors the human annotation process (Section 2.1), enabling a fair comparison between model-generated and human-annotated responses.

2.3 Evaluation Strategy

We hypothesize that the proposed approach in Section 2.2 evaluates an LM's ability to understand contact center conversations and autonomously reason through them to answer the question $\mathcal{Q}$ in $\mathcal{D}_{QA}$ based on supporting evidence. To align with real-world QA use cases, where each question may have a different Answer of Interest (AoI) and distinct label skew, we evaluate performance independently per question by computing Precision, Recall, and F1 scores on the minority class, which is often the Answer of Interest (AoI), to assess model performance. This per-question evaluation emulates the practical requirement that models perform reliably across a diverse set of QA criteria, each with varying operational importance and statistical properties. We emphasize AoI metrics, as they capture performance on the business-critical minority class prevalent in skewed QA distributions. To ensure holistic assessment, we also report Macro F1 to account for potential degradation in performance on the majority class.

In addition to predictive ability, we report the computation cost of each model, estimated based on the total number of tokens processed and generated over the dataset $\mathcal{D}_{QA}$ and projected to 1.5M daily conversations (assumption). For *Nova* and *Claude* models, we use token pricing published by *Amazon Bedrock*; for *OpenAI* models, we refer to pricing from *Azure OpenAI* as of July 04, 2025. This allows us to quantify the financial implications of model selection in enterprise-scale deployments.

2.4 Utility at Scale

As shown in Table 1, while the in-house model offers an extremely economical inference cost of $3,070 per 1.5M conversations, it exhibits limited effectiveness, particularly on AoI precision, which is a key business requirement in QA scenarios. Particularly, an AoI precision of just 54.2% indicates a high rate of false positives that can lead to unnecessary coaching, agent distrust, and wasted QA bandwidth. This, coupled with an overall Macro F1 of 67.0% and AoI F1 of 56. 1%, suggests that it lacks the robustness of reasoning required to accurately identify and support high confidence rare

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

case detections.

In contrast, large LMs in $\mathcal{M}$ demonstrate consistently superior performance, particularly in terms of AoI precision. While *Claude-4-Sonnet* achieves the highest AoI precision at 84.6%, other models such as *Claude-3.5-Haiku* (82.4%), *Nova-Premier* (82.3%), and *o3-Mini* (80.5) perform lower. Crucially, each of these models still represents a substantial improvement over the in-house model. This indicates that large LMs offer a significant boost in precision, enhancing the reliability of QA assessment in high-stakes enterprise settings.

However, this precision gain on AoI comes at a significant economic cost. The most accurate model, *Claude-4-Sonnet*, incurs a cost of $204,956 per 1.5M conversations, nearly 70× higher than the in-house baseline. Even relatively lower-cost models like *Nova-Pro* ($34,162) or *Claude-3.5-Haiku* ($43,059) remain an order of magnitude more expensive. Additionally, we observe that marginal improvements in AoI precision often result in steep increases in cost, making direct deployment of these models prohibitively expensive at enterprise scale.

This stark trade-off between performance and economic feasibility motivates our exploration of *selective routing*, with a special focus on attaining high precision on AoI in a cost-efficient manner.

3 STREAQ: A Two-Tier Routing Framework

We propose a two-tier evaluation framework that optimizes cost-performance trade-offs through intelligent query routing. Specifically, we implement a system where computationally expensive LMs are selectively utilized only when necessary.

3.1 Problem Formulation

Let M_s denote a cost-efficient small LM and M_l denote a more capable, large LM with superior reasoning capabilities but higher computational cost. Given a QA assessment input instance $I = (q, p, c)$, where q is the evaluation question, p is the corresponding plan, and c is the conversation transcript, we follow the methodology proposed in Ingle et al. (2024). Each instance is first processed by M_s to produce a textual response $\hat{y}_s$ which includes evidence, synthesis, and a binary answer (either *yes* or *no*). A routing function $R(I, M_s(I)) \to \{0, 1\}$ then determines whether to accept the prediction from M_s ($R = 0$) or escalate to M_l ($R = 1$). The final system output is defined as:

$$\hat{y}_{\text{final}}(I) = \begin{cases} M_s(I), & \text{if } R(I, M_s(I)) = 0 \\ M_l(I, M_s(I)), & \text{if } R(I, M_s(I)) = 1 \end{cases} \tag{1}$$

The total cost of processing n instances is:

$$C_{\text{total}} = \sum_{i=1}^{n} \left[C_s^{(i)} + R(I_i, M_s(I_i)) \cdot C_l^{(i)} \right] \tag{2}$$

where $C_s^{(i)}$ and $C_l^{(i)}$ denote the costs of processing instance I_i with M_s and M_l respectively. We hypothesize that an intelligent routing strategy R can substantially reduce C_{total} while preserving performance comparable to full inference using M_l on all instances.

3.2 Experimental Setup

To evaluate the effectiveness of the proposed approach in Section 3.1, we fix $M_s = M_{\textit{in-house}}$, the most cost-efficient model available, and vary $M_l \in \mathcal{M}$, where $\mathcal{M}$ is a set of larger, more capable LMs as discussed in Section 2.2. We adopt an adversarial routing strategy $\mathcal{R}$, employing a fine-tuned RoBERTa-based binary classifier to make informed routing decisions. The routing function is defined as $R(I, M_s(I)) = \Phi(\text{concat}(I, M_s(I)))$, where Φ denotes a learned binary classifier that takes as input the concatenation of the original instance I and the small LM's output $M_s(I)$.

Our choice of routing mechanism is inspired by prior work on adversarial routing strategies (Sordoni et al., 2023; Chen et al., 2023), which demonstrate significant gains in classification accuracy through learned decision boundaries. We further analyze the impact of different routing functions and observe trends consistent with those reported in (Pan et al., 2023) and (Ding et al., 2024) While routing function design plays a critical role in overall system performance, a comprehensive investigation is beyond the scope of this paper. For all subsequent experiments, we use the best-performing routing configuration identified through preliminary evaluations. A detailed comparison of routing strategies is provided in Appendix C.

In our implementation, router $\mathcal{R}$ is instantiated as a fine-tuned RoBERTa-based binary classifier. It is trained on a curated dataset of historical QA assessment instances, where ground truth routing labels are assigned based on alignment between the small model prediction M_s and the gold standard annotation. Specifically, an instance is labeled

Group	Model Configuration	AoI Precision (%)	AoI Recall (%)	AoI F1 (%)	Macro F1 (%)	Daily Cost
Baseline M_s	$M_{in\text{-}house}$	54.2%	64.1%	56.1%	67.0%	**$3070**
Full Inference (M_l)	nova-pro (M_l)	79.0%	77.3%	73.9%	82.0%	$34,162
	nova-premier (M_l)	82.3%	78.0%	78.8%	85.6%	$114,122
	o3-mini (M_l)	80.5%	76.8%	72.8%	81.4%	$90,097
	gpt-4o (M_l)	79.9%	**84.2%**	80.1%	85.8%	$135,779
	claude-3.5-haiku (M_l)	72.4%	83.0%	75.5%	82.0%	$43,059
	claude-4-sonnet (M_l)	**84.6%**	83.5%	**81.7%**	**87.0%**	$204,956
Two-Tier Inference ($M_s \xrightarrow{\mathcal{R}} M_l$) ($M_s$ reasoning is passed to M_l)	$M_s \xrightarrow{\mathcal{R}}$ nova-pro (M_l)	70.2%	61.2%	63.7%	74.5%	$17,842
	$M_s \xrightarrow{\mathcal{R}}$ nova-premier (M_l)	74.3%	71.3%	71.3%	79.4%	$60,950
	$M_s \xrightarrow{\mathcal{R}}$ o3-mini (M_l)	73.6%	70.6%	69.7%	78.7%	$40,354
	$M_s \xrightarrow{\mathcal{R}}$ gpt-4o (M_l)	73.5%	72.6%	70.1%	76.2%	$67,775
	$M_s \xrightarrow{\mathcal{R}}$ claude-3.5-haiku (M_l)	65.9%	61.2%	61.3%	72.4%	$20,476
	$M_s \xrightarrow{\mathcal{R}}$ claude-4-sonnet (M_l)	76.0%	74.2%	72.3%	80.0%	$87,533

Table 1: Performance comparison across Full Inference (M_s and M_l) and Two-Tier Inference ($M_s \xrightarrow{\mathcal{R}} M_l$) inference configurations. Blue indicates the best within each group, **bold** highlights the overall best. Cost is calculated as daily token-based API cost (refer to Section A).

Setup	Inference Paradigm	Scratchpad	M_s Context	AoI P (%)	AoI R (%)	AoI F1 (%)	Macro F1 (%)	Cost
P0	M_s			54.20%	64.10%	56.10%	67.00%	$3,070
P1	$M_{l_{\text{nova-pro}}}$			**79.00%**	**77.30%**	**73.90%**	**82.00%**	$34,162
P2	$M_s \xrightarrow{\mathcal{R}} M_{l_{\text{nova-pro}}}$	✓	✓	70.20%	61.20%	63.70%	74.50%	$17,842
P3	$M_s \xrightarrow{\mathcal{R}} M_{l_{\text{nova-pro}}}$		✓	69.30%	60.40%	62.30%	73.10%	**$16,947**
P4	$M_s \xrightarrow{\mathcal{R}} M_{l_{\text{nova-pro}}}$			76.20%	71.90%	70.70%	79.30%	$17,842

Table 2: Ablation study: Performance of different inference paradigms across evaluation tasks. Best results for each column are highlighted in **bold**. Checkmarks (✓) indicate enabled features, empty cells indicate disabled features.

with a routing decision of 0 if $M_s(I)$ matches the ground truth (i.e., accepted), and 1 otherwise (i.e., routed). Full training details, including training data collection strategy, hyperparameters and fine-tuning configurations, are provided in Appendix D.

3.3 Evaluation Strategy

We evaluate the proposed two-tier inference framework on $\mathcal{D}_{QA}$ following the evaluation strategy outlined in Section 2.3. The results across all configurations are summarized in Table 1.

4 Results and Discussion

4.1 Cost-Performance Trade-Off

As shown in Table 1, *Claude-4-Sonnet* continues to achieve the highest *AoI Precision* within the proposed two-tier framework, consistent with our earlier observations in Section 2.4. While full inference using *Claude-4-Sonnet* remains the top-performing configuration across all baselines, it incurs a prohibitive daily cost exceeding $200,000, rendering it impractical for most enterprise-scale deployments. In contrast, our proposed two-tier approach strikes a favorable balance, achieving 76% AoI Precision—preserving 89.8% of the performance of full inference with *Claude-4-Sonnet* while reducing daily inference cost by approximately 58%. This also translates to approximately 40% relative gain in AoI Precision compared to using M_s alone.

Interestingly, we find that full inference with *Nova-Pro* not only outperforms two-tier inference with *Claude-4-Sonnet* across all metrics but also reduces cost by 61%, making it a more practical alternative for enterprise-scale QA automation. This suggests that *Claude-4-Sonnet*, despite its accuracy, may not be suitable for cost-constrained deployments. On the other hand, simply resorting to full inference with *Nova-Pro* provides high performance on AoI while simultaneously maintaining cost efficiency. However, for organizations operating under tighter budget constraints, a two-tier inference setup using *Nova-Pro* still emerges as a strong contender—offering a further approximately 50% reduction in cost (to $17,842 per day) while retaining 88.9% of AoI Precision compared to full inference with *Nova-Pro*, and delivering a 29.5% relative gain over the M_s baseline.

A critical consideration is whether the observed drop in AoI precision (e.g., 8.6 percentage points for the two-tier Claude-4-Sonnet) is acceptable in practice. The answer depends on the operational context. For highly critical compliance tasks where errors can lead to significant regulatory or legal

penalties, the best-performing model may be non-negotiable, regardless of cost. However, for many standard QA use-cases, particularly those operating with a human-in-the-loop to validate flagged interactions, an 8.6% precision drop is a reasonable trade-off for a 58% reduction in daily costs (a saving of over $117,000). This allows organizations to reallocate substantial budget toward expanding QA coverage or other quality initiatives.

4.2 Ablation Study

We conduct a comprehensive ablation study examining how different inference paradigms affect performance on AoI metrics. As established in Section 4.1, the *Nova-Pro* model emerges as a strong candidate for two-tier routing due to its favorable balance between cost-efficiency and performance. Accordingly, we select *Nova-Pro* as the focus of this study. However, our ablation methodology is model-agnostic and can be readily extended to other models in the set $\mathcal{M}$.

Table 2 presents a comparative analysis in four key inference setups. As discussed in Section 2.4, full inference using M_l (setup P1), achieves the highest performance across all metrics and serves as an upper bound for what is achievable by selective routing strategies. The prompts used for these settings can be found in the Appendix section E.1

In Setup P2, we hypothesize that prompting M_l to independently generate a detailed reasoning trace (scratchpad) before contrasting it with M_s's output could allow M_l to both identify weaknesses in M_s's reasoning and reflect more deeply upon its reasoning trace using evidences surfaced by M_s. In line with our expectation, removing the scratchpad component from the reasoning generated by M_l (setup P3) in turn leads to a noticeable drop in performance across all metrics. This is likely due to the absence of M_l's independent reasoning process, which diminishes its ability to critically assess and revise conclusions.

Surprisingly, Setup P4, where both the scratchpad and M_s's reasoning are omitted, results in a substantial improvement in AoI metrics. This indicates that flawed reasoning from M_s can negatively bias M_l's inference trajectory, even when M_l is a significantly more capable model. From the analysis of this behavior, an example is chosen from a consistent set of failures from P2 and P3, highlighting the possible cause behind this behavior in Section E.3. These findings reinforce the idea that, in two-tier routing systems, including reasoning traces from low-precision models may degrade overall evaluation quality. Furthermore, omitting this context not only improves performance but also reduces token consumption, thus offering additional cost benefits. Therefore, this becomes our ideal STREAQ Routing $\mathcal{R}$ set-up.

Our ablation studies reveal that careful model design, particularly avoiding the propagation of flawed reasoning from weaker models, can further enhance performance. Overall, our findings underscore the potential of selective routing as a scalable and cost-efficient strategy for high-stakes QA assessment in real-world deployments.

4.3 Model Sensitivity and Scaling Laws

To further analyze model sensitivity and the scaling relationship between cost and performance, we calculated the marginal cost required to achieve incremental gains in Macro F1. As shown in Table 3, there is a clear trend of diminishing returns for both full and two-tier inference configurations.

In the "Full Inference" setup, upgrading from *Nova-Pro* to *Nova-Premier* costs approximately $22,211 for each percentage point of F1 gain. This cost escalates dramatically for the top-performing models; the final step from *gpt-4o* to *claude-4-sonnet* costs over $57,000 per F1 point. Our STREAQ framework significantly lowers these marginal costs in the lower and mid-tiers, demonstrating superior cost-effectiveness. For instance, the upgrade from *nova-lite* to *nova-pro* costs only $3,957 per F1 point in the two-tier setup, making performance gains much more accessible for budget-constrained operations.

5 Prior Work

Traditional contact center QA relies on manual, rule-based evaluations using scorecards, where analysts assess a sample of conversations for compliance, empathy, and conversation flow. This approach is labor-intensive, inconsistent, and does not scale to high-volume environments (Lee, 2023).

Domain-specific QA requirements in finance, telecom, and healthcare include financial disclosures (Altinok, 2018), HIPAA compliance (Neupane et al., 2025; Rahman et al., 2024), and technical troubleshooting accuracy (Kaplan, 2020). These tasks often demand customized lexicons, domain-tuned entity recognition, and sentiment analysis aligned with regulatory standards. Such specialization leads to severe class imbalance

Configuration	Model Upgrade (M_l)	Add. Daily Cost	Macro F1 Gain	Marginal Cost per 1% F1
Full Inference	nova-pro → nova-premier	+$79,960	+3.6%	~$22,211
	gpt-4o → claude-4-sonnet	+$69,177	+1.2%	~$57,648
	nova-premier → claude-4-sonnet	+$90,834	+1.4%	~$64,881
Two-Tier	nova-lite → nova-pro	+$12,662	+3.2%	~$3,957
	claude-3.5-haiku → gpt-4o	+$38,137	+9.0%	~$4,237
	nova-pro → nova-premier	+$31,743	+1.6%	~$19,839
	nova-premier → claude-4-sonnet	+$34,179	+0.4%	~$85,448

Table 3: Analysis of marginal cost per 1% Macro F1 gain for model upgrades. The table highlights the diminishing returns as models become more powerful, a trend STREAQ helps mitigate.

across QA labels. Henning et al. (2023) categorizes deep learning approaches into sampling, loss design, staged learning, and model-level strategies. Sampling methods like ROS, RUS, adaptive, and class-aware sampling (Carvalho et al., 2025) offer trade-offs between performance and the risk of overfitting or discarding informative examples.

Early model cascading aimed to balance accuracy and computational cost by invoking complex models only when simpler ones failed. While traditional cascades were static, recent work has shifted toward dynamic inference. Behera et al. (2025) propose a deployment-aware taxonomy to guide model selection under cost and latency constraints. Wang et al. (2025) introduce COSMOS, a system for predictable and cost-effective LLM adaptation that anticipates deployment costs while maintaining target performance.

Bai et al., 2024 provide a systematic survey of resource-efficient LLMs, highlighting strategies such as quantization, pruning, knowledge distillation, and early exit mechanisms. Arefeen, 2024 explores cost-efficiency in vision AI pipelines, detailing how query-aware optimization can be leveraged for LLM deployment in edge devices. These findings extend to generative models, proposing dynamic batching and caching techniques to further enhance performance.

6 Conclusion

We present STREAQ, a two-tier selective routing framework that balances cost and evaluation quality for enterprise-scale contact center QA. Our experiments show that STREAQ achieves up to 48% cost reduction while preserving over 88% of full-model AoI precision using *Nova-Pro*, making it a practical solution for large-scale deployments. Through a RoBERTa-based routing classifier and detailed cost-benefit analysis, we demonstrate the viability of selective routing in high-volume QA workflows. Our ablation studies further validate that flawed reasoning from smaller models can degrade performance, highlighting the importance of routing design. Together, these findings confirm STREAQ as an effective and scalable approach for economically viable, high-precision automated QA.

Limitations

This work presents several important limitations that should be considered when interpreting our results and applying our methodology in practice.

1. **Dataset and Domain Constraints:** Our evaluation is conducted on a proprietary contact center dataset that cannot be released publicly, limiting reproducibility and independent validation. Additionally, our dataset is limited to English-language conversations, predominantly featuring American English speech patterns, which constrains the generalizability of our findings to multilingual or multicultural contact center environments.

2. **Model and Architecture Limitations:** The evaluation focuses on six specific LLM families (*Nova*, *GPT*, *Claude*), and performance characteristics may vary significantly with newer model releases or different model architectures.

3. **Evaluation Scope and Metrics:** Our study is limited to binary QA tasks with yes/no responses, which may not reflect the complexity of contact center evaluations that involve multifaceted assessments. Additionally, our cost analysis is based on current API pricing models, which are subject to change and may not reflect the true operational costs of enterprise deployments.

4. **Annotation and Ground Truth Limitations:** Our ground truth labels are based on majority agreement among five expert annotators, achieving 62% agreement rates. This relatively low agreement suggests inherent ambiguity in some QA tasks, which may limit the reliability of our evaluation benchmark. Furthermore, the static nature of our annotations does not account for evolving business requirements or changing evaluation criteria over time.

5. **Generalizability Concerns:** Our findings are specific to the contact center QA domain and may not generalize to other text classification or routing tasks. The effectiveness of our routing strategy is likely dependent on the specific characteristics of contact center conversations, including their length, structure, and the types of QA questions typically asked. Organizations with different conversation patterns, evaluation criteria, or operational constraints may experience different cost-performance tradeoffs.

Ethical Considerations

The integration of artificial intelligence in employee performance evaluation demands adherence to established ethical principles and proactive risk management. We present our ethical framework organized around three core pillars: stakeholder protection, system integrity, and organizational responsibility.

Stakeholder Protection and Rights

1. **Data Rights and Privacy Protection:**
   - All voice recordings undergo anonymization procedures, removing personally identifiable information before computational processing.
   - Data retention policies must align with regulatory requirements and organizational needs, with clear deletion timelines.
   - Employees should have visibility into what data is collected, how it is used, and the duration of storage.

2. **Equitable Treatment Across Demographics:**
   - The system's current calibration for American English speech patterns limits its applicability to diverse linguistic backgrounds.
   - Deployment should be restricted to contexts where the training data adequately represents the target population.
   - Future adaptations must incorporate diverse speech patterns and undergo rigorous bias testing before implementation.

System Integrity and Reliability

1. **Technical Robustness and Validation:**
   - Continuous monitoring protocols detect performance degradation, unexpected behavioral changes, or systematic errors.
   - Regular validation against human expert evaluations ensures alignment with organizational quality standards.

- Fallback mechanisms activate when system confidence levels fall below acceptable thresholds.

2. **Interpretability and Auditability:**
   - Evaluation criteria and scoring methodologies are documented and accessible to relevant stakeholders.
   - The system provides explanations for its assessments, highlighting key factors that influenced scoring decisions.
   - Audit trails maintain records of system decisions, updates, and interventions for accountability purposes.

Organizational Responsibility and Governance

1. **Leadership Accountability:**
   - Management maintains ultimate responsibility for evaluation decisions and their consequences on employee welfare.
   - Clear escalation procedures exist for addressing system malfunctions, bias incidents, or ethical concerns.
   - Regular ethical impact assessments evaluate the system's effects on workplace culture and employee well-being.

2. **Adaptive Governance Framework:**
   - A multidisciplinary ethics committee oversees system deployment, including representatives from HR, legal, technical, and employee advocacy groups.
   - Periodic reviews assess the system's alignment with evolving ethical standards and regulatory requirements.
   - Stakeholder feedback mechanisms ensure continuous improvement and responsiveness to emerging concerns.

3. **Organizational Impact Considerations:**
   - Training programs prepare supervisors to effectively integrate AI insights with their professional judgment.
   - Change management processes support employees in adapting to AI-augmented evaluation practices.

4. **Implementation Commitment:** This ethical framework guides our approach to deploying AI-powered evaluation systems responsibly. We recognize that ethical AI implementation is an ongoing process requiring vigilance, adaptation, and commitment to prioritizing human welfare alongside operational efficiency. Regular reassessment ensures our practices evolve with technological capabilities and societal expectations.

References

Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, Alon Benhaim, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Martin Cai, Qin Cai, Vishrav Chaudhary, Dong Chen, Dongdong Chen, and 110 others. 2024. Phi-3 technical report: A highly capable language model locally on your phone. *Preprint*, arXiv:2404.14219.

Duygu Altinok. 2018. An ontology-based dialogue management system for banking and finance dialogue systems. *Preprint*, arXiv:1804.04838.

Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M. Dai, Anja Hauth, Katie Millican, David Silver, Slav Petrov, Melvin Johnson, Ioannis Antonoglou, Julian Schrittwieser, Amelia Glaese, Jilin Chen, Emily Pitler, Timothy P. Lillicrap, and 33 others. 2023. Gemini: A family of highly capable multimodal models. *CoRR*, abs/2312.11805.

Anthropic. 2023. Model card: Claude 3 technical report. [Online; accessed 18-July-2024].

Md Adnan Arefeen. 2024. *Cost-Efficient Vision AI: Challenges and Solutions for Real-Time and Stored Video Analytics With Classical and Generative AI*. Ph.D. thesis, University of Missouri–Columbia.

G Bai, Z Chai, C Ling, S Wang, J Lu, and N Zhang. 2024. Beyond efficiency: A systematic survey of resource-efficient large language models. *arXiv preprint arXiv:2401.00625*.

AP Behera, JP Champati, and R Morabito. 2025. Towards efficient multi-llm inference: Characterization and analysis of llm routing and hierarchical techniques. *arXiv preprint arXiv:2506.06579*.

Miguel Carvalho, Armando J. Pinho, and Susana Brás. 2025. Resampling approaches to handle class imbalance: a review from a data perspective. *Journal of Big Data*, 12(1):71.

N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. 2002. Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357.

Nitesh V. Chawla, Aleksandar Lazarevic, Lawrence O. Hall, and Kevin W. Bowyer. 2003. Smoteboost: Improving prediction of the minority class in boosting. In *Knowledge Discovery in Databases: PKDD 2003*, pages 107–119, Berlin, Heidelberg. Springer Berlin Heidelberg.

Lingjiao Chen, Matei Zaharia, and James Zou. 2023. Frugalgpt: How to use large language

models while reducing cost and improving performance. *Preprint*, arXiv:2305.05176.

Dujian Ding, Ankur Mallick, Chi Wang, Robert Sim, Subhabrata Mukherjee, Victor Ruhle, Laks V. S. Lakshmanan, and Ahmed Hassan Awadallah. 2024. Hybrid llm: Cost-efficient and quality-aware query routing. *Preprint*, arXiv:2404.14618.

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. *Preprint*, arXiv:2210.04675.

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.

Micaela Kaplan. 2020. May i ask who's calling? named entity recognition on call center transcripts for privacy law compliance. *Preprint*, arXiv:2010.15598.

Aobo Kong, Shiwan Zhao, Hao Chen, Qicheng Li, Yong Qin, Ruiqi Sun, Xin Zhou, Enzhi Wang, and Xiaohang Dong. 2024. Better zero-shot reasoning with role-play prompting. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, pages 4099–4113. Association for Computational Linguistics.

Christopher M. Lee. 2023. Formulated quality assurance (qa) and customer satisfaction (csat) scorecards indexing and inference research information from the business process outsource (bpo) workplace. *SSRN Electronic Journal*.

Joffrey L. Leevy, Taghi M. Khoshgoftaar, Richard A. Bauder, and Naeem Seliya. 2018. A survey on addressing high-class imbalance in big data. *Journal of Big Data*, 5(1):42.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. *Preprint*, arXiv:1711.05101.

Subash Neupane, Sudip Mittal, and Shahram Rahimi. 2025. Towards a hipaa compliant agentic ai system in healthcare. *Preprint*, arXiv:2504.17669.

OpenAI. 2023. GPT-4 technical report. *CoRR*, abs/2303.08774.

Liangming Pan, Michael Stephen Saxon, Wenda Xu, Deepak Nathani, Xinyi Wang, and William Yang Wang. 2023. Automatically correcting large language models: Surveying the landscape of diverse self-correction strategies. *ArXiv*, abs/2308.03188.

Md Abdur Rahman, Md Abdul Barek, ABM Kamrul Islam Riad, Md Mostafizur Rahman, Md Bajlur Rashid, Smita Ambedkar, Md Raihan Miaa, Fan Wu, Alfredo Cuzzocrea, and Sheikh Iqbal Ahamed. 2024. Embedding with large language models for classification of hipaa safeguard compliance rules. *Preprint*, arXiv:2410.20664.

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.

Alessandro Sordoni, Xingdi Yuan, Marc-Alexandre Côté, Matheus Pereira, Adam Trischler, Ziang Xiao, Arian Hosseini, Friederike Niedtner, and Nicolas Le Roux. 2023. Joint prompt optimization of stacked llms using variational inference. *Preprint*, arXiv:2306.12509.

J Wang, A Albarghouthi, and F Sala. 2025. COSMOS: Predictable and cost-effective adaptation of llms. *arXiv preprint arXiv:2505.01449*.

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*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, and 3 others. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Manzil Zaheer, Guru Guruganesh, Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. 2021. Big bird: Transformers for longer sequences. *Preprint*, arXiv:2007.14062.

A LLMs API Cost Estimation

We assume the system handles approximately **1.5** million user conversations per day. Each conversation triggers a single API call to an LLM, which is evaluated against an average of 15 question-answering (QA) prompts. We assume continuous speech without pauses or turn-taking delays.

For the spoken content, we use a typical speech rate of **140** words per minute, consistent with natural conversational English. Based on this rate, we estimate that a single conversation corresponds to approximately **1,688** words of spoken text. Using a standard word-to-token conversion ratio (approximately **0.75** tokens per word), this translates to **2,250** input tokens per conversation. The output from the LLM, representing a concise answer or response, is estimated to be about **256** tokens.

According to the *Claude-3.5-Sonnet* pricing model, input tokens are billed at **$3** per million tokens, while output tokens are billed at **$15** per million tokens.

Given these assumptions, the system would issue approximately **22.5** million API calls per day (1.5 million conversations × 15 prompts). This results in a daily token usage of about **50.6** billion input tokens (2,250 × 22.5M) and **5.7** billion output tokens (256 × 22.5M). Applying the pricing structure, the estimated cost amounts to $151,875 for input tokens and $8,625 for output tokens, yielding a total projected daily LLM API cost of approximately **$160,500**.

These estimates highlight the substantial cost implications of large-scale LLM integration and underscore the need for optimization strategies in production environments.

B Data Distribution and Sampling Heuristics

This section provides a detailed overview of the 10 QA questions constituting our benchmarking dataset $\mathcal{D}_{QA}$, along with the distribution of their binary labels in Table 4. We also shed light on the general sampling heuristics employed in carefully sampling conversations for these questions to maintain a meaningful distribution across both class labels.

B.1 Conversation Sampling Heuristics

We leverage a fundamental understanding of contact center conversation transcripts, based on turns and natural language phraseology, to arrive at elementary heuristics for assigning provisional answer labels to question-conversation pairs. While these heuristics are bound by their limitations and human annotation is subsequently required for high-quality ground truth labels, these do provide some direction in ensuring a good sample size across both class labels for each question. A few such heuristics for different questions are outlined below:

- For the question - *Was the customer's complete email address mentioned by the agent prior to the customer saying it?* - we use a regex pattern to detect email strings in the transcript, and if the first such instance occurs in an agent turn, we conclude the answer as *yes*. We also add some customer name checks to mitigate false positives, as the agent may occasionally provide the email of one of their support colleagues to assist the customer.
- A similar approach is used for the question, *Was the customer's complete phone number stated unprompted by the agent?*, where the first positive match of a phone number regex string in an agent turn results in a *yes*. That said, accounting for "complete" phone number patterns of geographically diverse locations becomes a challenge, besides again, the agent sharing a support phone number with the customer.
- For the question - *Did the agent reflect an understanding of the customer's query or concern?* - We employ a temporally constrained combination of fuzzy and semantic matching of the first few, say 2 or 3, customer and agent turns. This is usually where the customer elaborates on their issue, and the agent responds with a paraphrased version or asks clarifying questions around the same. A strong match points to the answer *yes*,

Question	No	Yes	Total
Does the agent respond empathetically?	108	130	238
Did the agent reveal customer email unsolicitedly?	62	61	123
Did the agent follow proper unresponsiveness protocol?	273	69	342
Did the agent correctly identify customer's need to change mode of order?	180	99	279
Did the agent verify the customer's identity?	48	40	88
Did the agent breach PII compliance by sharing sensitive customer data?	52	37	89
Did the agent demonstrate active listening?	92	141	233
Did the agent address the customer with proper salutations?	48	45	93
Did the agent share appropriate documentation with the customer?	99	57	156
Did the agent take nessecary steps to lead the call to conclusion?	169	69	238

Table 4: Label Distributions for Questions in Benchmark Dataset

although this heuristic can suffer if the agent doesn't verbalize their understanding of the customer's concern.

- For the question - *Was the customer's delivery location stated by the agent before being mentioned by the customer?*, we use out-of-the-box Named Entity Recognition (NER) to identify location type entities in transcript turns, and if the first occurrence is in an agent turn, we flag this as *yes*.

C Routing Strategies

We evaluate three distinct routing strategies, each targeting different aspects of the cost-performance optimization problem:

C.1 Probabilistic Routing (Baseline)

The probabilistic routing strategy serves as our baseline, implementing query-agnostic routing decisions through a Bernoulli distribution:

$$R_{\text{prob}}(I, M_s(I)) \sim \text{Bernoulli}(p) \quad (3)$$

where $p \in [0, 1]$ is a fixed routing probability. This strategy provides a cost-controlled baseline that is independent of query characteristics or model outputs, enabling assessment of whether intelligent routing mechanisms provide meaningful improvements over random selection.

C.2 Deterministic Routing (Business-Aware)

The deterministic routing strategy leverages domain knowledge about business-critical performance requirements by routing based on predicted answer classes and their business importance:

$$R_{\text{det}}(I, M_s(I)) = \begin{cases} 1, & \text{if } \hat{y}_s = y_{\text{target}} \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

where y_{target} represents the answer of interest determined by business requirements for each question and $\hat{y}s$ is the binary answer extracted from $M_s(I)$. The routing rule can be business-defined based on specific operational requirements and cost considerations. For this study, we routed on the minority class label ($y_{\text{target}} = \text{minority}$) since most QA use cases require higher precision on the minority class to achieve business-relevant outcomes such as agent coaching.

C.3 Routing $\mathcal{R}$ (Learned)

The routing strategy employs a fine-tuned RoBERTa-based binary classifier to make intelligent routing decisions:

$$R_{\text{learned}}(I, M_s(I)) = \Phi(\text{concat}(I, M_s(I))) \quad (5)$$

where Φ represents a learned binary classifier that takes as input the concatenation of the original instance I and the small model's textual response $M_s(I)$. In the current setup, we implement Φ as a fine-tuned RoBERTa-based binary classifier. The classifier is fine-tuned on a dataset of historical examples where ground truth routing decisions are determined by comparing M_s and M_l predictions, with instances routed when M_s predictions differ from M_l ground truth labels. Detailed fine-tuning procedures, hyperparameters, and training configurations for the RoBERTa-based classifier are provided in Section D.

D Training Configurations for RoBERTa-based binary classifier

This section details the experimental setup for training the RoBERTa-based binary classifier, including data preprocessing, model architecture, hyperparameter selection, loss function, and optimization strategies.

D.1 Data Preprocessing and Label Binarization

The dataset comprises prompts and corresponding responses from two large language models (LLMs): an in-house LLM (fine-tuned *phi-3-mini* (Abdin et al., 2024)) and a large LLM M_l (Zaheer et al., 2021). Each data point is labeled based on the correctness of these responses. For the routing task, we binarize the labels to focus on whether a prompt should be routed to the large LLM M_l:

- **Label 1**: Cases where the large LLM M_l is correct and the in-house LLM is incorrect, or both are incorrect.
- **Label 0**: All other cases.

This binarization ensures the classifier is optimized for the routing decision.

The training dataset was curated following the methodology described in Section 2.1, scaled to include a larger volume of diverse prompts and their corresponding responses. The original multiclass labels were binarized for the routing task, where samples requiring routing (labels -1 and 1) were assigned to class 1, and all others to class 0. The dataset was partitioned using stratified sampling with an **80-10-10** split for training, validation, and testing, respectively, ensuring balanced class distribution across all splits.

D.2 Model Architecture

We employ Google's bigbird-roberta-base model (Zaheer et al., 2021), a transformer-based architecture capable of handling long sequences efficiently. The model is fine-tuned for binary sequence classification using the HuggingFace Transformers library (Wolf et al., 2020). The training hyperparameters were selected based on empirical analysis of the dataset and memory constraints. Key settings include:

- **Maximum Sequence Length**: 4096 tokens (covers 100% of data with a safety margin).
- **Batch Size**: 16 for training, 32 for evaluation.
- **Number of Epochs**: 20.
- **Learning Rate**: 1×10^{-5}.
- **Gradient Accumulation Steps**: 2 (effective batch size: 32).
- **Warmup Steps**: 150.
- **Weight Decay**: 0.01.
- **Gradient Clipping**: Maximum norm of 1.0.
- **Mixed Precision Training**: Enabled (FP16).
- **Gradient Checkpoints**: Enabled for memory efficiency.
- **Evaluation and Checkpoints**: Evaluation and checkpoints occur every 40 steps, with a maximum of 5 checkpoints retained.

Given the class imbalance in the routing task, we employ a weighted cross-entropy loss. Class weights are computed using the `compute_class_weight` utility from scikit-learn, ensuring that minority classes are not neglected during training. The loss function is implemented as:

$$\mathcal{L} = -\sum_{i=1}^{N} w_{y_i} \log p_{y_i} \tag{6}$$

Where:

- N is the number of samples,
- w_{y_i} is the weight for the true class y_i of sample i,
- p_{y_i} is the predicted probability for the true class y_i of sample i.

The model was optimized using the AdamW optimizer (Loshchilov and Hutter, 2019). A linear learning rate schedule with warmup was employed, where the learning rate was initially increased linearly for the first **150** steps before decaying. To prevent overfitting and improve generalization, weight decay regularization with a coefficient of **0.01** was applied. Gradient clipping was used to stabilize training, with the maximum gradient norm set to **1.0**.

Mixed precision training (FP16) and gradient checkpointing were enabled to reduce memory consumption, allowing for longer input sequences and larger batch sizes.

Balanced accuracy is used as the primary metric for model selection, given the imbalanced

nature of the task. To further ensure robustness, the training pipeline included frequent evaluation and checkpointing, with up to **5** checkpoints retained during training.

E Baseline $M_s \rightarrow M_l$ Experiments

E.1 Prompt Templates

In this section, we provide various prompts used in the experiments.

```
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
2. Sub-criteria
3. Conversation transcript
4. Answer options

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 1: Implicit CoT Reasoning prompt template for M_s and M_l. Used in P0, P1 and P4 ablation experiments.

```
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
.

You are supported by an AI assistant referred to as
the "L1 Layer," but it is essential that you
critically assess its reasoning and form your own
judgment where necessary.

Below are the required information pieces fror your
task

1. Main question
2. Sub-criteria
3. Conversation transcript
4. Answer options
5. L1 Layer Reasoning
6. L1 layer Response


### Instructions:
1. Read the conversation and the evaluation question
 carefully.
2. Assess each sub-criterion on its own merit.
3. Develop your own detailed analysis and determine
the correct answer to the main question.
4. If  L 1 s  reasoning is thorough and accurate, you
 may refer to it. If you detect errors, omissions,
or weak logic, prioritize your own independent
assessment.
5. Your response must end with a definitive "Yes" or
 "No" answer to the evaluation question, grounded in
 your analysis.


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: Implicit CoT Reasoning prompt template for M_s and M_l. Used in P3 ablation experiments where reasoning is passed but scratchpad (independent reasoning) is not provided.

E.2 Performance across experimental settings

This section reports the performance of the baseline stacked model inference settings where a smaller model M_s is used to route inputs probabilistically or deterministically to a larger model M_l. We consider five primary configurations:

- **Probabilistic Routing (No Context)**: A simple stacked setup where the routing from M_s to M_l is learned probabilistically without incorporating reasoning information from the output of M_s.
- **Deterministic Routing (No Context)**: A rule-based stacking approach that deterministically routes inputs based on M_s predictions, still without reasoning from M_s.
- **Deterministic Routing with M_s Context**: The same stacking-based routing strategy as above, now with reasoning from M_s to inform the decision.
- **Machine-Learned Routing (No Context)**: A data-driven routing function trained to map the M_s output to M_l, without reasoning from M_s
- **Machine-Learned Routing with Context**: A data-driven routing function

```
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
.

You will receive support from an AI assistant called
 the "L1 Layer," but you must independently
scrutinize its reasoning and override it if needed.

Below are the required information pieces for your
task

1. Main question
2. Sub-criteria
3. Conversation transcript
4. Answer options
5. L1 Layer Reasoning
6. L1 layer Response


### Instructions:
1. Review the conversation and question in detail.
2. Evaluate each sub-criterion individually.
3. Conduct your own analysis and determine the most
appropriate answer to the main question.
4. You may use  L 1 s  reasoning if it is thorough
and accurate. However, if you detect gaps, errors,
or incomplete logic, rely on your independent
judgment.
5. Conclude your evaluation with a definitive "Yes"
or "No" answer, supported by your reasoning.

Please structure your response using the sections
below:

**List your analysis here, without considering the
L1's reasoning:**

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 3: Implicit CoT Reasoning prompt template for M_s and M_l. Used in P2 ablation experiments where reasoning is passed, but scratchpad (independent reasoning) is also provided.

trained to map the M_s output to M_l, using both predictions and reasoning M_s as characteristics.

Each configuration is evaluated in multiple model pairings, measuring performance via **Macro F1**, **AoI F1**, **AoI Precision**, **AoI Recall**, and associated **Inference Cost**. The bold values indicate the highest score for a given metric in all configurations.

E.3 Why not passing M_s reasoning to M_l improves the performance

Upon manual analysis of examples where we resisted passing the context to M_l from M_s, we identified a consistent behavior. M_s **reasoning often dominates the general reasoning of the inference**. We hypothesize that the reasoning of M_s is a very strong single form for M_l, which often dominates the reasoning process of M_l. Pair this with the fact that an ideal routing framework should be routing all the wrong answers from M_l exclusively, passing the context ends up feeding wrong information to M_l. And given M_l is susceptible to being fooled by reasoning from M_s, this creates a feedback loop in the routing framework, where we send examples which are wrong along with their reasoning, which leaves M_l vulnerable to making the wrong decision again. We chose a representative example that highlights this behavior over a question about *Did the agent follow proper unresponsiveness protocol?*, where the expected answer is **no**

- M_s synthesis: While the agent does not use the exact phrases mentioned in the sub-criteria, their statement about waiting for the refund process to complete can be interpreted as an indirect way of ending the conversation or call due to the inability to resolve the customer's issue further. The agent acknowledges that they cannot provide any other solution and are essentially closing the conversation by initiating the refund process.**Answer: yes**
- M_l synthesis: The conversation shows that the agent continues to attempt to communicate with the customer even after the customer stops responding. However, there is no clear statement from the agent indicating that they are ending the call due to the customer's lack of response. The agent's last communication is still focused on providing information and resolving the issue, rather than explicitly stating an intent to disconnect. **Answer: no**
- $M_s \xrightarrow{\mathcal{R}} M_l$ synthesis: - The conversation shows the customer being very vocal and

Model Configuration	AoI P (%)	AoI R (%)	AoI F1 (%)	Macro F1 (%)	Cost
$M_s \xrightarrow{\mathcal{R}}$ gpt-4o	62.3%	**73.8%**	**65.2%**	**74.1%**	$49,216
$M_s \xrightarrow{\mathcal{R}}$ gpt-4o-mini	58.3%	68.1%	60.9%	70.8%	$2,068
$M_s \xrightarrow{\mathcal{R}}$ claude-3.5-haiku	62.5%	73.1%	**65.2%**	73.8%	$14,789
$M_s \xrightarrow{\mathcal{R}}$ claude-4-sonnet	**63.0%**	71.1%	64.9%	73.9%	$67,759
$M_s \xrightarrow{\mathcal{R}}$ nova-lite	60.7%	67.3%	61.5%	71.7%	**48**
$M_s \xrightarrow{\mathcal{R}}$ nova-pro	60.6%	68.2%	62.0%	71.9%	$11,577
$M_s \xrightarrow{\mathcal{R}}$ nova-premier	62.2%	69.5%	63.3%	73.1%	$37,134
$M_s \xrightarrow{\mathcal{R}}$ o3-mini	60.0%	68.0%	60.8%	71.0%	$30,791

Table 5: Probabilistically Stacked-model inference ($M_s \xrightarrow{\mathcal{R}} M_l$) without M_s context. **Bold** highlights best values across all models.

expressing frustration. The agent continues to respond to the customer's concerns and provides explanations and apologies. There is no explicit disclosure from the agent about disconnecting the call due to the customer's lack of response. **Answer: no**

- $M_s \xrightarrow{\mathcal{R}} M_l + M_s reasoning$ synthesis: While the agent does not use the exact phrases mentioned in the sub-criteria, their statement about waiting for the refund process to complete can be interpreted as an indirect way of ending the conversation or call due to the inability to resolve the customer's issue further. The agent acknowledges that they cannot provide any other solution and are essentially closing the conversation by initiating the refund process. Additionally, the customer's last message is unclear and incomplete, which could indicate that the agent decided to end the call due to the lack of a coherent response.**Answer: yes**

This illustrates a *cognitive anchoring effect*: M_l becomes biased by the interpretive lens of M_s, treating its speculative reasoning as strong evidence. Rather than acting as an independent verifier, M_l starts echoing M_s's flawed logic.

Thus, this behavior supports the hypothesis that M_s's reasoning is not just a soft prior but an overpowering influence, especially harmful when M_s is wrong. This undermines the core goal of the routing framework, which is to leverage M_l's robustness to correct M_s's errors, not amplify them.

Model Configuration	AoI P (%)	AoI R (%)	AoI F1 (%)	Macro F1 (%)	Cost
$M_s \xrightarrow{\mathcal{R}}$ o3-mini	**84.3%**	49.8%	57.8%	72.1%	$34,539
$M_s \xrightarrow{\mathcal{R}}$ nova-pro	81.2%	50.3%	59.2%	73.0%	$12,252
$M_s \xrightarrow{\mathcal{R}}$ nova-premier	84.0%	52.2%	62.4%	75.2%	$40,588
$M_s \xrightarrow{\mathcal{R}}$ nova-lite	78.0%	45.8%	53.4%	69.4%	**$954**
$M_s \xrightarrow{\mathcal{R}}$ gpt-4o-mini	78.9%	46.4%	54.2%	69.8%	$2,294
$M_s \xrightarrow{\mathcal{R}}$ claude-3.5-haiku	76.6%	51.9%	60.4%	73.4%	$15,403
$M_s \xrightarrow{\mathcal{R}}$ claude-4-sonnet	**86.6%**	**53.4%**	**63.8%**	**75.9%**	$73,149
$M_s \xrightarrow{\mathcal{R}}$ gpt-4o	82.7%	52.9%	62.4%	75.4%	$49,803

Table 6: Deterministic Stacked-model inference ($M_s \xrightarrow{\mathcal{R}} M_l$) without M_s context. **Bold** highlights best values across all models.

Model Configuration	AoI P (%)	AoI R (%)	AoI F1 (%)	Macro F1 (%)	Cost
$M_s \xrightarrow{\mathcal{R}}$ o3-mini	78.0%	59.8%	64.0%	75.4%	$35,542
$M_s \xrightarrow{\mathcal{R}}$ nova-pro	79.0%	46.8%	55.9%	70.8%	$16,185
$M_s \xrightarrow{\mathcal{R}}$ nova-premier	**82.8%**	**58.0%**	**65.5%**	**76.6%**	$54,751
$M_s \xrightarrow{\mathcal{R}}$ nova-lite	64.1%	47.5%	52.6%	67.7%	**$1,186**
$M_s \xrightarrow{\mathcal{R}}$ gpt-4o-mini	69.1%	55.0%	58.5%	71.3%	$2,719
$M_s \xrightarrow{\mathcal{R}}$ claude-3.5-haiku	71.4%	54.5%	58.9%	72.0%	$18,569
$M_s \xrightarrow{\mathcal{R}}$ claude-4-sonnet	82.5%	54.4%	63.2%	75.2%	$81,448
$M_s \xrightarrow{\mathcal{R}}$ gpt-4o	78.3%	59.4%	64.9%	76.0%	$62,849

Table 7: Deterministic Stacked-model inference ($M_s \xrightarrow{\mathcal{R}} M_l$) with M_s context. **Bold** highlights best values across all models.

Model Configuration	AoI P (%)	AoI R (%)	AoI F1 (%)	Macro F1 (%)	Cost
$M_s \xrightarrow{\mathcal{R}}$ o3-mini (M_l)	76.8%	71.4%	69.9%	78.8%	$39,246
$M_s \xrightarrow{\mathcal{R}}$ nova-pro (M_l)	76.2%	71.9%	70.7%	79.3%	$13,727
$M_s \xrightarrow{\mathcal{R}}$ nova-premier (M_l)	77.8%	70.7%	72.6%	80.9%	$45,470
$M_s \xrightarrow{\mathcal{R}}$ nova-lite (M_l)	73.0%	67.5%	66.2%	76.1%	**$1,065**
$M_s \xrightarrow{\mathcal{R}}$ gpt-4o-mini (M_l)	72.6%	66.4%	65.7%	75.8%	$2,583
$M_s \xrightarrow{\mathcal{R}}$ claude-3.5-haiku (M_l)	71.5%	73.0%	70.0%	78.3%	$17,019
$M_s \xrightarrow{\mathcal{R}}$ claude-4-sonnet (M_l)	**78.1%**	74.9%	73.9%	81.3%	$79,649
$M_s \xrightarrow{\mathcal{R}}$ gpt-4o (M_l)	76.7%	**77.0%**	**75.0%**	**82.0%**	$55,156

Table 8: Two-tier inference ($M_s \xrightarrow{\mathcal{R}} M_l$) with Router $\mathcal{R}$ (stacking, no M_s reasoning added). **Bold** highlights best values across all configurations.

Model Configuration	AoI P (%)	AoI R (%)	AoI F1 (%)	Macro F1 (%)	Cost
$M_s \xrightarrow{\mathcal{R}}$ o3-mini	73.6%	70.6%	69.7%	78.7%	$40,354
$M_s \xrightarrow{\mathcal{R}}$ nova-pro	70.2%	61.2%	63.7%	74.5%	$17,842
$M_s \xrightarrow{\mathcal{R}}$ nova-premier	74.3%	71.3%	71.3%	79.4%	$60,950
$M_s \xrightarrow{\mathcal{R}}$ nova-lite	61.3%	61.8%	60.0%	70.8%	**$1,437**
$M_s \xrightarrow{\mathcal{R}}$ gpt-4o-mini	65.5%	63.4%	62.5%	73.1%	$3,113
$M_s \xrightarrow{\mathcal{R}}$ claude-3.5-haiku	65.9%	61.2%	61.3%	72.4%	$20,476
$M_s \xrightarrow{\mathcal{R}}$ gpt-4o	73.5%	72.6%	70.1%	76.2%	$67,775
$M_s \xrightarrow{\mathcal{R}}$ claude-4-sonnet	**76.0%**	**74.2%**	**72.3%**	**80.0%**	$87,533

Table 9: Routing $\mathcal{R}$ Stacked-model inference ($M_s \xrightarrow{\mathcal{R}} M_l$) with M_s context. **Bold** highlights best values across all models.