SAFE-QAQ: End-to-End Slow-Thinking Audio-Text Fraud Detection via Reinforcement Learning

Peidong Wang[1]* **Zhiming Ma**[1,2]* **Xin Dai**[1]* **Yongkang Liu**[3] **Shi Feng**[1]†
Xiaocui Yang[1] **Wenxing Hu**[4] **Zhihao Wang**[1] **Mingjun Pan**[2]
Li Yuan[5] **Daling Wang**[1]
[1]School of Computer Science and Engineering,
Northeastern University, Shenyang 110819, China
[2]China Mobile Internet Company Limited, Guangzhou, China
[3]School of Computer and Communication Engineering, Northeastern University, Qinhuangdao
[4]Shanghai University of Electric Power, China [5]Peking University, Shenzhen, China
pdongwang@163.com, fengshi@cse.neu.edu.cn

Abstract

Existing fraud detection methods predominantly rely on transcribed text, suffering from ASR errors and missing crucial acoustic cues like vocal tone and environmental context. This limits their effectiveness against complex deceptive strategies. To address these challenges, we first propose **SAFE-QAQ**, an end-to-end comprehensive framework for audio-based slow-thinking fraud detection. First, the SAFE-QAQ framework eliminates the impact of transcription errors on detection performance. Secondly, we propose rule-based slow-thinking reward mechanisms that systematically guide the system to identify fraud-indicative patterns by accurately capturing fine-grained audio details through hierarchical reasoning processes. Besides, our framework introduces a dynamic risk assessment framework during live calls, enabling early detection and prevention of fraud. Experiments on the TeleAntiFraud-Bench demonstrate that SAFE-QAQ achieves dramatic improvements over existing methods in multiple key dimensions, including accuracy, inference efficiency, and real-time processing capabilities. Currently deployed and analyzing over 70,000 calls daily, SAFE-QAQ effectively automates complex fraud detection, reducing human workload and financial losses. Code: https://github.com/Control-derek/SAFE-QAQ.

1 Introduction

With the rapid development of mobile communication technology, the problem of telecom fraud has become increasingly severe, emerging as a global social challenge. As illustrated in Figure 1(a), most existing methods (Chang et al., 2024; Korkanti, 2024; Yang et al., 2025; Shen et al., 2025b; Hu et al., 2024) for fraud detection have made significant advancements by utilizing transcribed texts and the powerful representation capabilities of large language models (LLMs) (Dubey et al., 2024; Achiam et al., 2023; Guo et al., 2025). However, these methods often result in misjudgments in real-time fraud detection systems due to their inability to update information and policies in a timely manner. To address this issue, researchers have proposed using Retrieval-Augmented Generation (RAG) to stay updated with the latest information and policies, eliminating the need for retraining (Singh et al., 2025). Despite these efforts, the application of fraud detection systems in

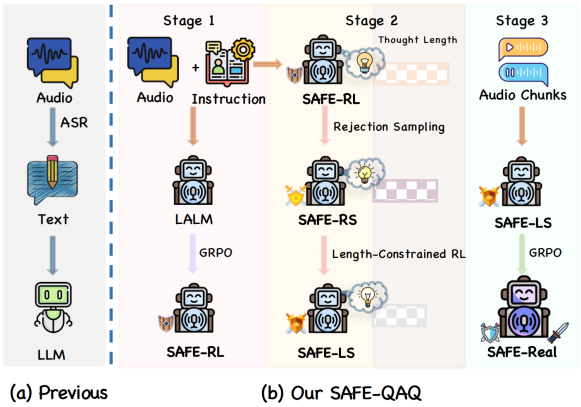

Figure 1: Comparison of (a) Previous Method and (b) Our Proposed Method: End-to-End Call Fraud Detection via Reinforcement Learning (RL). Our approach trains LALMs in three stages: (i) developing slow-thinking reasoning through RL, (ii) optimizing thought length using rejection sampling fine-tuning and length-constrained RL, and (iii) achieving real-time detection with audio chunks training.

* These authors contributed equally to this research.
† Corresponding author.

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 26138–26157
July 2-7, 2026

real-world scenarios remains challenging and far from fully successful, primarily due to the following three reasons. First, the transcription of speech into text by Automatic Speech Recognition (ASR) systems can introduce error noise, leading to a decrease in model performance (Ma et al., 2025; Hu et al., 2024; Chakraborty et al., 2024). Second, text-based fraud detection systems cannot capture the fine-grained information conveyed by speech, such as vocal tone, emotional stress, and environmental acoustics, all of which are crucial for effective fraud detection. Third, modern fraudsters employ elaborate layered deceptive strategies—combining manipulated speech patterns (e.g., voice spoofing or synthetic audio), fabricated background sounds (e.g., fake call center noise), and psychologically coercive scripts—that require iterative reasoning to unravel (Ma et al., 2025). However, current text-based pipeline methods lack mechanisms for deep reasoning to effectively address such complexities. These limitations highlight the inadequacy of text-only approaches in tackling modern telecommunications fraud and sophisticated deceptive strategies.

To address these challenges, we propose SAFE-QAQ (**S**low-thinking **A**udio-text **F**raud d**E**tection using **Q**wen **A**udio with **Q**uestion), an end-to-end comprehensive framework for audio-based slow-thinking fraud detection. Building upon recent advancements in Large Audio-Language Models (LALMs) (Chu et al., 2024; Huang et al., 2025; Hurst et al., 2024; Zeng et al., 2024), SAFE-QAQ establishes a complete end-to-end pipeline that directly processes raw audio signals to preserve crucial multimodal features while incorporating three key innovations (as shown in Figure 1(b)):

- SAFE-QAQ develops slow-thinking reasoning through rule-based reward, enabling the system to systematically analyze fine-grained details, which are frequently concealed by fraudsters using layered deceptive strategies.
- we further optimize reasoning efficiency by reducing reasoning chain lengths by 48.87% through rejection sampling fine-tuning (producing *SAFE-RS*) and length-constrained RL (resulting in *SAFE-LS*), ensuring concise yet accurate reasoning.
- SAFE-QAQ achieves real-time detection by dynamically assessing information sufficiency through structured prompting and phase recognition rewards, culminating in the final model (*SAFE-Real*) that enables timely interventions during live calls.

By eliminating reliance on error-prone ASR transcriptions and integrating slow-thinking reasoning with reinforcement learning-optimized multimodal processing, SAFE-QAQ establishes a fully end-to-end framework that achieves both high accuracy and practical efficiency. This viability is demonstrated by its successful deployment in a production pipeline processing over 70,000 calls daily, where it effectively alleviates manual audit burdens and prevents financial losses through timely, automated intervention.

2 Related work

2.1 LLM-Based Telecom Fraud Detection

Recent advances in LLMs have shown promise for telecom fraud detection, with methods like Retrieval-Augmented Generation (RAG) for real-time call analysis (Singh et al., 2025) and intent-based warning systems (Shen et al., 2025b). However, these approaches rely solely on transcribed text, discarding critical audio features (e.g., tone, emotion) that signal fraud (Chang et al., 2024). While multimodal benchmarks like TeleAntiFraud-28k (Ma et al., 2025) address this gap, their supervised fine-tuning (SFT) methods underutilize modern LLMs' reasoning capabilities. Current systems also suffer from inefficiency, requiring multi-stage pipelines for transcription and fraud detection (Yang et al., 2025). SAFE-QAQ overcomes these limitations by processing raw audio end-to-end via reinforcement learning (RL), preserving multimodal cues while eliminating intermediate steps. This approach enables faster, more accurate fraud detection tailored to real-world dynamics.

2.2 Large Audio Language Models (LALMs)

LALMs such as Qwen2-Audio, GLM-4-Voice, GPT-4o, and Step-Audio have shown strong performance in speech understanding, capturing tone, emotion, and intent in real time (Chu et al., 2024; Zeng et al., 2024; Hurst et al., 2024; Huang et al., 2025), but their application to fraud detection remains limited. General-purpose LALMs often fail to detect scripted deception, where vocal delivery subtly contradicts scripted calmness. **SAFE-QAQ** bridges this gap by aligning audio-language modeling with domain-specific RL optimization for telecom fraud detection, enabling context-aware, risk-sensitive reasoning.

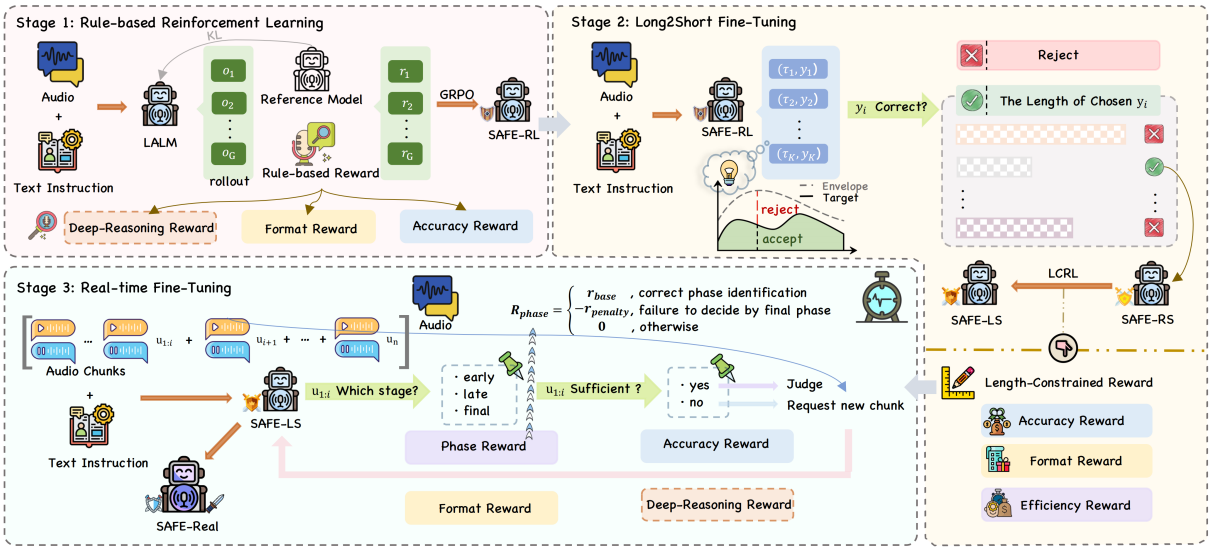

Figure 2: Overview of Our Method. Starting from an LALM, we: (i) apply rule-based RL to obtain SAFE-RL with slow-thinking capabilities; (ii) refine it using rejection sampling (SAFE-RS) and length-constrained RL (SAFE-LS) to improve reasoning efficiency; and (iii) perform real-time fine-tuning on audio chunks to derive SAFE-Real.

2.3 Reinforcement Learning for Slow-Thinking

Recent advancements in Reinforcement Learning (RL) have enabled LLMs to develop slow-thinking capabilities, mimicking human-like System 2 reasoning (Kahneman, 2011). Methods like OpenAI's o1/o3 (Jaech et al., 2024; OpenAI, 2025), DeepSeek R1 (Guo et al., 2025), and Satori (Shen et al., 2025a) demonstrate notable improvements in tasks requiring step-by-step analysis, such as mathematics (Zhu et al., 2022; Lu et al.), logic (Jin et al., 2024), and multimodal reasoning (Xu et al., 2024; Thawakar et al., 2025), with the underlying mechanisms of such reasoning models further surveyed by Wang et al. (2025). These models leverage techniques such as Monte Carlo Tree Search (MCTS) (Świechowski et al., 2023) and reward-guided fine-tuning (Trung et al., 2024) to generate extended reasoning chains, enhancing their ability to solve complex problems. However, current RL-based approaches primarily focus on text-based reasoning, leaving multimodal domains like audio-text integration underexplored. Challenges such as overthinking (Chen et al., 2024) and inefficiency in dynamic scenarios (Qi et al., 2024) highlight the need for more tailored solutions.

3 Method

Figure 2 illustrates the three-stage framework of our approach. In Stage 1, we use rule-based reinforcement learning to train a model capable of slow-thinking. In Stage 2, we refine it via Long2Short fine-tuning to shorten reasoning and mitigate overthinking. Finally, in Stage 3, we apply Real-Time fine-tuning to optimize the model for efficient, real-time fraud detection.

3.1 Problem Definition

The task involves three classification objectives based on audio analysis: **scenario classification**, **fraud detection**, and **fraud type classification**. Given an input pair (u, t) consisting of raw audio u and text instruction t, the model π generates output $o = (\tau, y)$, where τ is the step-by-step reasoning process and y contains both the classification rationale and final result. The objective is to develop π that accurately performs these tasks while providing interpretable reasoning.

3.2 Rule-based Reinforcement Learning

As illustrated in Figure 2, Stage 1 employs rule-based reinforcement learning for data-efficient self-evolution, yielding a slow-thinking model (SAFE-RL) that analyzes audio-text cues to detect subtle fraud patterns beyond text-only approaches.

Group Relative Policy Optimization. In contrast to traditional actor-critic algorithms such as Proximal Policy Optimization (PPO) (Schulman et al., 2017), we optimize our model using Group Relative Policy Optimization (GRPO), which eliminates the need for a critic model with parameter complexity comparable to that of the policy model π_θ. Instead, GRPO estimates the relative advan-

tage of each response based on intra-group scoring. Specifically, for each audio-text instruction pair $(u,t) \sim P(U,T)$, the policy model π_θ samples multiple reasoning processes and their corresponding responses. The output for the i-th sample is represented as $o_i = (\tau_i, y_i)$. Each response y_i is evaluated by a rule-based reward model to compute the reward value $r_i = R(y_i)$. The intra-group relative advantage $A_i = \frac{r_i - mean(\{r1,r2,...,r_G\})}{std(\{r1,r2,...,r_G\})}$, derived from these reward values, is then used to optimize the model via the objective function $J_{GRPO}(\theta)$:

$$J_{GRPO}(\theta) = \mathbb{E}_{(u,t)\sim P(U,T),\{o_i\}_{i=1}^{G}\sim\pi_{\theta_{old}}(O|u,t)} \frac{1}{G} \sum_{i=1}^{G}(min(\rho_i A_i, clip_\epsilon(\rho_i)A_i) - \beta\mathbb{D}_{KL}(\pi_\theta \parallel \pi_{ref})) \quad (1)$$

The importance sampling factor $\rho_i(\theta)$ is defined as the ratio between the current policy π_θ and the sampling policy $\pi_{\theta_{\text{old}}}$. The clipping function $clip_\epsilon(\rho_i)$ constrains ρ_i within the interval $[1-\epsilon, 1+\epsilon]$, ensuring conservative policy updates. The hyperparameter β controls the strength of the KL divergence $\mathbb{D}_{KL}$ unbiasedly estimated using:

$$\mathbb{D}_{KL}(\pi_\theta||\pi_{ref}) = \frac{\pi_{ref}(o_i|u,t)}{\pi_\theta(o_i|u,t)} - \log\frac{\pi_{ref}(o_i|u,t)}{\pi_\theta(o_i|u,t)} - 1 \quad (2)$$

Reward Modeling. Training an outcome-based or process-based neural reward model is complex and resource-intensive. Transferring a general-purpose neural reward model to a specific domain requires considerable amounts of data and computational resources. In contrast, a rule-based reward model can effectively model rewards by simply designing validation rules for the answers.

- **Accuracy Reward (R_{acc})**: Validates the final answers y_i extracted from <answer> tags:

$$R_{acc} = \mathbb{I}(y_i \text{ is correct}) \quad (3)$$

- **Format Reward (R_{fmt})**: Enforces structure with <think> and <answer> tags:

$$R_{fmt} = \mathbb{I}(\text{format is satisfied}) \quad (4)$$

- **Deep-Reasoning Reward (R_{depth})**: Uses length-sensitive rewards for deeper reasoning. Logarithmic normalization improves sensitivity to shorter chains:

$$R_{depth} = \min\left(\frac{\ln(|\tau|+1)}{\ln(L_{max}+1)}, 1\right)\cdot R^{max} \quad (5)$$

where $|\tau|$ is reasoning token count, L_{max} the token limit and R^{max} is the reward ceiling.

The total reward R_{total} is computed as:

$$R_{total} = \alpha R_{acc} + \beta R_{fmt} + \gamma R_{depth} \quad (6)$$

We set weights $\alpha = 5$ and $\beta = 1$ to prioritize accuracy and format, and define $\gamma = \mathbb{I}_{\text{non-SFT}}$ to encourage deep reasoning specifically for non-SFT models. This configuration balances reliable fraud detection with structured, thorough analysis.

3.3 Long2Short Fine-Tuning

As shown in Figure 2, Stage 2 optimizes efficiency via **Long2Short Fine-Tuning**. This stage combines Rejection Sampling (SAFE-RS) and Length-Constrained RL (SAFE-LS) to compress reasoning chains without sacrificing precision.

Rejection Sampling Fine-Tuning. Sampling K candidates $\{o_i = (\tau_i, y_i)\}_{i=1}^{K}$ from the proposal $\pi_\theta(\cdot|u,t)$, we define a target π^* to prioritize correctness and brevity:

$$\pi^*(o|u,t) \propto \mathbb{I}(y \text{ is correct}) \cdot (1+|\tau|)^{-1} \quad (7)$$

The optimal response o^* is selected by maximizing:

$$o^* = \arg\max_{i\in[K]} \pi^*(o_i|u,t) \quad (8)$$

This selects the shortest correct response. The resulting dataset $\{(u,t,o^*)\}$ is then used for SFT to train π_θ towards concise reasoning.

Length-Constrained Reinforcement Learning (LCRL). Following SAFE-RS, we optimize efficiency via a composite reward $R_{LC} = \alpha R_{acc} + \beta R_{fmt} + \lambda R_{eff}$, incorporating accuracy (Eq. 3) and format (Eq. 4) objectives. We set $\lambda = 1$. The efficiency reward R_{eff} penalizes token excess $E = \max(0, |\tau| - L_{threshold})$:

$$R_{eff} = -\min\left(\max\left(\frac{\ln(E+10)}{\ln(B)}, 0.1\right), 1\right)\cdot P^{max} \quad (9)$$

where $B = 1000$ controls curvature and P^{max} sets the magnitude. This logarithmic scaling curbs verbosity without hindering necessary reasoning, ensuring rapid and reliable fraud detection.

3.4 Real-Time Fine-Tuning

Stage 3 enables dynamic risk assessment on sequential audio $u_{1:i}$. At each turn i, the model identifies the conversation phase (early, late, final). Through prompt engineering, we guide the model to: (1) permit early judgments, (2) formulate conclusions

Type	Model	Classification				Quality Assessment				Fin.
		Sce.	Fra.	FT.	AVG	Log.	Pra.	Exp.	SUM	
ASR+LLM	GLM4-9B-Chat	75.10	46.91	82.22	68.08	1.61	1.43	2.20	5.24	51.14
	InternLM2.5-20B	78.34	36.67	**85.42**	66.81	1.99	1.93	2.43	6.35	50.21
	Qwen2.5-72B	78.31	**51.44**	81.24	**70.33**	2.21	2.16	2.70	7.07	**52.87**
	Doubao 1.5 Pro	71.14	36.11	82.25	63.17	1.94	1.75	2.60	6.29	47.48
	Deepseek V3	**88.53**	14.62	66.71	56.62	**2.32**	**2.34**	**2.85**	**7.51**	42.59
ASR+LRM	Deepseek R1	83.60	79.25	85.16	82.67	**3.18**	**3.26**	**3.50**	**9.94**	62.17
LALM	GLM4-9B-Voice	0.00	26.83	38.33	21.72	0.89	0.64	0.65	2.18	16.33
	Gemini-2-Flash	80.51	59.61	83.53	74.55	**2.25**	**2.29**	**2.72**	**7.26**	56.03
	GPT4-o	80.25	50.00	**86.26**	72.17	2.12	2.10	2.56	6.78	54.24
	Step-Audio-Chat	76.35	40.65	79.71	65.57	1.64	1.62	2.01	5.27	49.27
	Qwen2-Audio-7B-Instruct	70.22	58.51	20.48	49.74	1.51	1.42	1.96	4.89	37.38
	AntiFraud-Qwen2Audio	**81.31**	**84.78**	82.91	**83.00**	2.06	2.07	2.31	6.44	**62.36**
LALM+Ours	**SAFE-RL**	81.57	**90.20**	**87.25**	**86.34**	**2.5**	2.64	2.97	**8.11**	**64.89**
	SAFE-RS	81.60	**89.61**	86.39	85.87	2.45	2.6	**2.99**	8.04	64.53
	SAFE-LS	**84.64**	**89.61**	**88.23**	**87.49**	2.49	**2.65**	2.97	**8.11**	**65.76**

Table 1: Performance of models on TeleAntiFraud-Bench. **Red** values represent SOTA results, **blue** values indicate the second-best performance, and **bold** values denote the best performance within the respective model type.

in late phases, and (3) mandate decisions by the final phase. We train phase awareness via:

$$R_{phase} = r_b \cdot \mathbb{I}_{corr} - r_p \cdot \mathbb{I}_{fail} \tag{10}$$

where $\mathbb{I}_{corr}$ denotes correct phase identification, $\mathbb{I}_{fail}$ marks final phase indecision, and r_b, r_p are the corresponding reward and penalty magnitudes. The total reward is $R_{total} = \alpha R_{acc} + \beta R_{fmt} + \eta R_{depth} + \delta R_{phase}$. We set $\delta = 5$ and $\eta = \mathbb{I}_{\text{non-SFT}}$. This configuration balances accuracy (R_{acc}) with phase-appropriate decision timing (R_{phase}).

4 Experiments

4.1 Experimental Setup

Datasets. We utilize **TeleAntiFraud-28k** (Ma et al., 2025) (28,511 pairs) for three tasks: 7-class scenario, 2-class fraud detection, and 7-class fraud type identification. For RL, we use only raw audio and context without reasoning annotations. In Real-Time Fine-Tuning, turns $u_{1:n}$ are segmented into early ($i < n/2$), late ($n/2 \leq i < n$), and final ($i = n$) phases. Evaluation is conducted on the distribution-preserving **TeleAntiFraud-Bench** (Ma et al., 2025).

Baselines. We compare our approach with proprietary models (GPT-4o (Hurst et al., 2024), Gemini-2-Flash (DeepMind, 2024), Doubao-1.5 (TEAM, 2025)) and open-source baselines including Deepseek V3/R1 (Liu et al., 2024; Guo et al., 2025), GLM-4-Voice (Zeng et al., 2024), Step-Audio (Huang et al., 2025), and Qwen2-Audio variants (Chu et al., 2024; Ma et al., 2025). This selection covers reasoning specialists, ASR+LLM cascades, and end-to-end multimodal architectures across diverse scales. For text-based baselines, we use the human-corrected transcripts provided by TeleAntiFraud-28k rather than external ASR outputs, so the comparison isolates the modality gap from transcription noise.

Evaluation Metrics. We report Weighted F1 for Scenario (Sce.), Fraud (Fra.), and Type (FT.) tasks, averaged as **AVG**. Reasoning quality is scored (0-5) on Logical Rigor (Log.), Practical Value (Pra.), and Expression (Exp.), summing to **SUM** (0-15). The final metric is Fin. $= 0.75 \cdot \text{AVG} + 0.25 \cdot (\text{SUM}/15)$. Following the original TeleAntiFraud benchmark, we retain **Fin.** for direct comparability while always reporting **AVG** and **SUM** separately to expose the trade-off between task performance and reasoning quality. Additional validation of the LLM judge and further discussion of metric interpretability are provided in Appendix D.

Implementation Details. Our backbone is AntiFraud-Qwen2Audio (Ma et al., 2025), an SFT version of Qwen2-Audio-7B-Instruct (Chu et al., 2024). The training pipeline sequentially applies Rule-based RL (SAFE-RL), Rejection Sampling (SAFE-RS), and Length-Constrained RL (SAFE-LS). All stages use the same underlying training corpus, with later stages changing the optimization target rather than the data distribution, which reduces the risk of stage-wise forgetting. Detailed hyperparameters are provided in Appendix C.

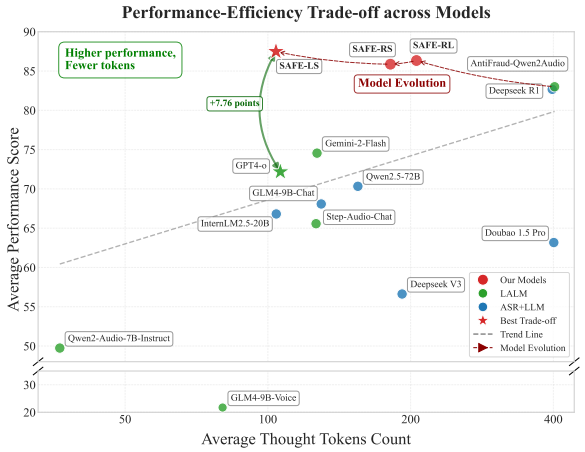

Figure 3: Performance-Efficiency Trade-off: Scatter Plot of Average Thinking Tokens vs. Average Classification Performance. Models closer to the top-left corner achieve a better balance of higher efficiency (fewer thinking tokens) and superior performance (higher classification scores). The points representing the best trade-offs for the baselines and our model are highlighted with star markers.

4.2 Effectiveness of SAFE-QAQ in Telecom Fraud Analysis

Performance Improvements Across Tasks. The experimental results in Table 1 reveal a consistent performance hierarchy across TeleAntiFraud-Bench tasks, with our SAFE-QAQ series models demonstrating progressive improvements over both general baselines and the specialized AntiFraud-Qwen2Audio foundation. In scenario classification ($Sce.$), while massive LLMs like Deepseek V3 (88.53) leverage their textual understanding capabilities to dominate this largely language-based task, our SAFE-LS (84.64) notably outperforms its precursor AntiFraud-Qwen2Audio (81.31) despite sharing the same architecture, confirming that our reinforcement learning framework enhances performance even for tasks where the base model already showed competence. This 3.33-point improvement is particularly notable given that the 8B-parameter AntiFraud-Qwen2Audio had already surpassed most ASR+LLM baselines through domain-specific fine-tuning.

The most striking advancements emerge in fraud detection ($Fra.$), where the evolutionary trajectory from base model to final system becomes apparent. The general-purpose Qwen2-Audio-7B-Instruct achieves only 58.51, while its SFT-enhanced version AntiFraud-Qwen2Audio reaches 84.78 through slow-thinking adaptation - already outperforming specialized text-based models like Deepseek R1 (79.25). Our SAFE-RL then extends this to 90.20 through rule-based reinforcement learning, representing a 5.42-point absolute improvement that demonstrates our method's exceptional capability in identifying subtle multimodal fraud patterns. This progression from general LALM to domain-adapted SFT model to RL-optimized system validates the complementary value of each training phase, with the final SAFE-LS achieving 88.23 in fraud-type classification ($FT.$) - surpassing even GPT4-o (86.26) and establishing new benchmarks for fine-grained multimodal analysis.

Model	Sce.	Fra.	FT.	AVG	Dur.	Turns
base	70.22	58.51	20.48	49.74	*48.31s*	*6.36*
w SFT	81.31	84.78	82.91	83.00	*48.31s*	*6.36*
SAFE-RL	81.57	90.20	87.25	86.34	*48.31s*	*6.36*
SAFE-RS	81.60	89.61	86.39	85.87	*48.31s*	*6.36*
SAFE-LS	84.64	89.61	88.23	87.49	*48.31s*	*6.36*
SAFE-Real	91.40	88.93	77.56	85.96	**8.98s**	**1.25**

Table 2: SAFE-Real vs. Baselines

Reasoning Advancement. Comprehensive metrics further validate this approach. While AntiFraud-Qwen2Audio (62.36 Fin.) already exceeds specialized text models like Deepseek R1 (62.17) through multimodal fine-tuning, our SAFE-LS (65.76) sets a new state-of-the-art through its full optimization pipeline. The 3.40-point final improvement reflects balanced advancements across all capabilities, with quality assessment scores (8.11 SUM) approaching those of dedicated reasoning models like Deepseek R1 (9.94 SUM), despite SAFE-LS using only 8B parameters compared to Deepseek R1's 671B. This efficiency gain is achieved by our reinforcement learning framework that systematically promotes slow-thinking processes, enhancing the model's logical reasoning, practical judgment, and expressive quality. These results collectively demonstrate that while traditional approaches excel in narrow competencies, our end-to-end multimodal framework delivers superior real-world performance where detection accuracy, classification precision, and reasoning quality must operate synergistically. An additional text-only ablation further confirms that these gains are not solely due to stronger reasoning supervision: removing raw audio causes clear degradation for both AntiFraud-Qwen2Audio and SAFE-LS across all three tasks, with the largest drop appearing in fraud detection. The detailed comparison is reported in

Model	Sce.	Fra.	FT.	AVG
SAFE-RL	81.57	90.20	87.25	**86.34**
w/o SFT	73.37	85.70	82.91	80.66
w/o SFT w R_{depth}	77.25	86.97	81.94	82.05
SAFE-RS	81.60	89.61	86.39	**85.87**
w/o SFT	79.81	87.14	83.82	83.59
SAFE-LS	84.64	89.61	88.23	**87.49**
w/o SFT	82.89	91.42	87.07	87.13
w/o RS	82.39	90.43	86.10	86.31

Table 3: Ablation Study: Performance of Our Models

Model	Log.	Pra.	Exp.	Sum	tokens
SAFE-RL	2.50	2.64	2.97	8.11	205.76
w/o SFT	1.93	1.92	2.39	6.24	31.46
w/o SFT w R_{depth}	2.10	2.07	2.59	6.76	212.72
SAFE-RS	2.45	2.60	2.99	8.04	181.33
w/o SFT	2.09	2.09	2.60	6.78	173.78
SAFE-LS	2.49	2.65	2.97	8.11	104.02
w/o SFT	2.17	2.25	2.64	7.06	74.05
w/o RS	2.23	2.31	2.78	7.32	106.20

Table 4: Ablation Study: Reasoning Quality and Token Efficiency of Our Models

Appendix G.

4.3 Performance-Efficiency Trade-off

In real-world fraud detection systems, computational efficiency directly impacts operational costs and response time - shorter reasoning chains enable faster fraud identification during live calls while reducing infrastructure expenses. Figure 3 examines the performance-efficiency trade-off across different detection systems, quantified through our proposed Thinking Efficiency Metric (TEM = $AVG/\log(|\tau|)$). This metric offers a hardware-agnostic view of efficiency based on reasoning token counts; a detailed analysis of system-level performance, including wall-clock latency and throughput, is provided in Appendix A. The scatter plot positions each model based on its average reasoning token length ($|\tau|$, log-scaled) versus average classification performance (AVG), revealing fundamental architectural differences and optimization trajectories. Systems positioned closer to the top-left region achieve superior balance between computational efficiency (shorter reasoning chains) and detection accuracy (higher F1 scores).

Advantages of Audio. LALM architectures (green circles) systematically outperform ASR+LLM baselines (blue circles) in TEM, with average TEM scores of 33.20 versus 29.72 respectively. This efficiency advantage stems from LALMs' native multimodal processing capabilities, which not only eliminate the error accumulation inherent in cascaded ASR+LLM pipelines but also capture audio-specific semantic information beyond just speech content, encompassing rich paralinguistic signals.

Scaling Laws in Fraud Reasoning. We observe a scaling law-like relationship between reasoning complexity and performance gains, where increasing the reasoning tokens yields logarithmic improvements in detection accuracy. Our analysis reveals remarkably consistent scaling patterns: when comparing GPT4-o to Gemini-2-Flash ($\Delta log(|\tau|) = 0.0773$) and GLM4-9B-Chat to Qwen2.5-72B ($\Delta log(|\tau|) = 0.0772$), we find nearly identical performance gains ($\Delta AVG = 2.38$ vs 2.25 respectively), with the ratio $\Delta log(|\tau|)/\Delta AVG$ remaining stable across model families (0.0325±0.0018). This scaling behavior suggests that fraud reasoning tasks exhibit fundamental dynamics similar to those observed in large language model pre-training, though our reinforcement learning framework ultimately breaks this pattern through targeted optimization.

Effectiveness of Multi-Stage Optimization. Our SAFE optimization pathway (dark red trajectory) demonstrates systematic efficiency gains while maintaining performance superiority. Our optimization starts from the slow-thinking AntiFraud-Qwen2Audio (TEM=31.87), rule-based reinforcement learning in SAFE-RL reduces average reasoning tokens by 48.87% while improving $F1_{avg}$ by 3.34 points (TEM=37.32). Subsequent rejection sampling fine-tuning (SAFE-RS) achieves an additional 11.87% token reduction before length-constrained RL finalizes the optimization in SAFE-LS (TEM=43.38). This three-stage refinement yields 36.12% higher TEM than the original base model, ultimately outperforming GPT4-o's TEM by 7.76 points through coordinated reasoning compression and performance enhancement.

Cost-Effective SOTA Performance. The rightmost cluster contains specialized reasoning models like Deepseek R1 (TEM=16.03) that use exhaustive token generation (>397 tokens on average) to achieve competitive accuracy. While these systems approach the performance ceiling, their operational costs hinder real-world deployment. Our SAFE-LS achieves state-of-the-art accuracy (88.23 AVG) using only 25.85% of the tokens required by comparable models. This efficiency enables our deployed

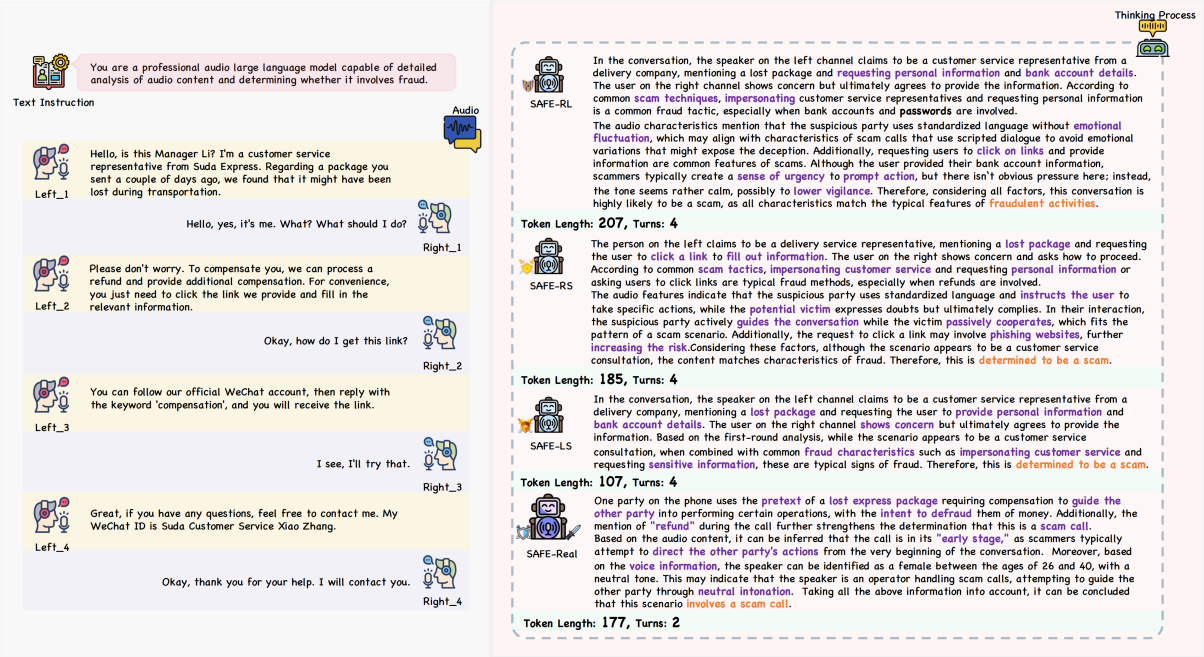

Figure 4: Model Output Case Study: Input with Text Instruction and Audio (ASR Results for Clarity, Left), Reasoning Process of SAFE-QAQ Series (Right). Key reasoning points are highlighted in purple, and inference results are marked in orange.

system to process over 70,000 screened calls daily, significantly reducing manual audit burdens and preventing financial losses through automated intervention. We additionally provide preliminary evidence on an emerging out-of-distribution fraud pattern in Appendix H. This sets a new practical benchmark for fraud detection.

4.4 Real-Time Detection Performance

Table 2 demonstrates that SAFE-Real achieves real-time detection with an average duration of just 8.98 seconds (81.4% faster than non-real-time models), while maintaining robust fraud detection performance (88.93 F1). Although its fraud-type classification accuracy decreases by 12.2% (77.56 F1) compared to SAFE-LS, this trade-off is operationally justified: in real-world fraud prevention, early detection takes precedence over precise typing, as promptly stopping active scams can prevent financial losses. The model's superior scenario classification (91.40 F1) and ultra-low average of 1.25 conversational turns enable highly effective live call interception. To complement these aggregate results, we further report deployment-oriented operating-point statistics, including precision, recall, false positive rate, and false negative rate, in Appendix I.

4.5 Ablation Studies

The ablation studies in Tables 3 and 4 demonstrate the critical role of SFT pretraining and the progressive improvements of our SAFE framework. Removing SFT leads to notable degradation across all metrics (e.g., SAFE-RL's performance drops 5.68 points to 80.66 and reasoning quality decreases 1.87 points to 6.24), though incorporating Deep-Reasoning Reward (R_{depth}) without SFT partially mitigates these losses (improving to 82.05 and 6.76 respectively). Our token-efficient variants (SAFE-RS/SAFE-LS) maintain strong performance (85.87/87.49) while using substantially fewer tokens (181.33/104.02 vs 205.76), while preserving high reasoning quality (8.04/8.11 respectively). Yet they still benefit from SFT's foundational capabilities: Removing SFT causes SAFE-RS/SAFE-LS to drop to 83.59/87.13 with lower reasoning quality (6.78/7.06). This confirms SFT's essential role in establishing baseline abilities that subsequent RL stages enhance rather than replace. Notably, when we remove rejection sampling in the Long2Short stage (SAFE-LS w/o RS), we observe performance degradation (87.49 to 86.31) along with decreased reasoning quality, demonstrating the necessity of Rejection Sampling Fine-Tuning for maintaining performance while improving efficiency in subsequent stages. We additionally ex-

amine the decision to disable the depth reward for SFT-initialized models, and the corresponding ablation confirms that enabling this reward yields nearly identical classification quality while slightly increasing output length; the detailed results are presented in Appendix E.

5 Case Study

Figure 4 shows our models' reasoning processes in a typical "lost package refund" scam scenario. From SAFE-RL to SAFE-RS and SAFE-LS, the length of the models' reasoning processes progressively shortens (from 207 to 107 tokens) as a result of the Long2Short optimization. Concurrently, the density of key reasoning points increases, demonstrating that Long2Short enables more efficient reasoning and enhances model efficiency. Due to the analysis to assess the current stage of the call, SAFE-Real employs a moderate-length reasoning process. Notably, SAFE-Real achieves interpretable fraud detection using only two rounds of dialogue audio, underscoring its high efficiency. Critically, our models extract essential paralinguistic cues (accents, emotions, vocal tones) directly from raw audio, which ASR transcriptions would lose, thereby exposing the subtle mismatch between calm tone and urgent intent.

6 Conclusion

We present SAFE-QAQ, an end-to-end slow-thinking audio-text fraud detection framework trained via reinforcement learning. By integrating GRPO with rule-based rewards, Long2Short optimization, and real-time learning, our model achieves state-of-the-art performance on TeleAntiFraud-Bench (88.23 F1) with significantly improved efficiency (48.87% shorter chains and 81.4% faster speed). Beyond academic metrics, SAFE-QAQ has been successfully deployed in a production pipeline processing over 70,000 calls daily, where it effectively reduces manual audit burdens and prevents financial losses. This work validates that multimodal slow-thinking architectures can be both robust and practically efficient, offering a scalable solution for real-world security challenges.

Limitations

While SAFE-QAQ demonstrates superior performance, our experimental scope is inevitably constrained by the fact that TeleAntiFraud-28k is currently the only open-source dataset suitable for training Large Audio-Language Models for fraud detection. This data scarcity restricts a more extensive evaluation of generalization capabilities across highly diverse fraud scenarios or unexpected acoustic conditions. Consequently, although our model shows strong noise resilience and we report a preliminary out-of-distribution result on an emerging fraud pattern in Appendix H, further systematic validation is still required to ensure robustness in extreme real-world environments with severe interference, signal degradation, and newly evolving scam tactics.

Ethical Statement

This research upholds strict data privacy standards. For experimental validation, we used the anonymized TeleAntiFraud-28k dataset, ensuring no exposure of personally identifiable information (PII). Regarding the real-world deployment, our system is implemented in collaboration with telecom operators via privacy-preserving intermediate number services. This approach masks actual phone numbers, with all data processing authorized by enterprises strictly for anti-fraud quality inspection. We emphasize that SAFE-QAQ is designed exclusively as a defensive tool, and we mandate continuous monitoring to prevent misuse and mitigate potential algorithmic bias in practical applications.

Acknowledgments

The work is supported by the National Natural Science Foundation of China (No. 62272092, No. 62172086), National Science Foundation for Young Scientists of China (No. 62502081), and the Fundamental Research Funds for the Central Universities under Grants (N2523011, N25XQD004). Special thanks go to Qingyun Pan, Tianhao Wu and Jintao Huang for their insightful guidance and helpful contributions.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Joymallya Chakraborty, Wei Xia, Anirban Majumder, Dan Ma, Walid Chaabene, and Naveed Janvekar. 2024. Detoxbench: Benchmarking large language

models for multitask fraud & abuse detection. *arXiv preprint arXiv:2409.06072*.

Chen-Wei Chang, Shailik Sarkar, Shutonu Mitra, Qi Zhang, Hossein Salemi, Hemant Purohit, Fengxiu Zhang, Michin Hong, Jin-Hee Cho, and Chang-Tien Lu. 2024. Exposing llm vulnerabilities: Adversarial scam detection and performance. In *2024 IEEE International Conference on Big Data (BigData)*, pages 3568–3571. IEEE.

Xingyu Chen, Jiahao Xu, Tian Liang, Zhiwei He, Jianhui Pang, Dian Yu, Linfeng Song, Qiuzhi Liu, Mengfei Zhou, Zhuosheng Zhang, and 1 others. 2024. Do not think that much for 2+ 3=? on the overthinking of o1-like llms. *arXiv preprint arXiv:2412.21187*.

Yunfei Chu, Jin Xu, Qian Yang, Haojie Wei, Xipin Wei, Zhifang Guo, Yichong Leng, Yuanjun Lv, Jinzheng He, Junyang Lin, and 1 others. 2024. Qwen2-audio technical report. *arXiv preprint arXiv:2407.10759*.

Google DeepMind. 2024. Introducing gemini 2.0: our new ai model for the agentic era.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Sihao Hu, Tiansheng Huang, Ka-Ho Chow, Wenqi Wei, Yanzhao Wu, and Ling Liu. 2024. Zipzap: Efficient training of language models for large-scale fraud detection on blockchain. In *Proceedings of the ACM Web Conference 2024*, pages 2807–2816.

Ailin Huang, Boyong Wu, Bruce Wang, Chao Yan, Chen Hu, Chengli Feng, Fei Tian, Feiyu Shen, Jingbei Li, Mingrui Chen, and 1 others. 2025. Step-audio: Unified understanding and generation in intelligent speech interaction. *arXiv preprint arXiv:2502.11946*.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, and 1 others. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, and 1 others. 2024. Openai o1 system card. *arXiv preprint arXiv:2412.16720*.

Mingyu Jin, Qinkai Yu, Dong Shu, Haiyan Zhao, Wenyue Hua, Yanda Meng, Yongfeng Zhang, and Mengnan Du. 2024. The impact of reasoning step length on large language models. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 1830–1842.

Daniel Kahneman. 2011. *Thinking, fast and slow*. macmillan.

Sukanth Korkanti. 2024. Enhancing financial fraud detection using llms and advanced data analytics. In *2024 2nd International Conference on Self Sustainable Artificial Intelligence Systems (ICSSAS)*, pages 1328–1334. IEEE.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, and 1 others. 2024. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.

Pan Lu, Liang Qiu, Kai-Wei Chang, Ying Nian Wu, Song-Chun Zhu, Tanmay Rajpurohit, Peter Clark, and Ashwin Kalyan. Dynamic prompt learning via policy gradient for semi-structured mathematical reasoning. In *The Eleventh International Conference on Learning Representations*.

Zhiming Ma, Peidong Wang, Minhua Huang, Jinpeng Wang, Kai Wu, Xiangzhao Lv, Yachun Pang, Yin Yang, Wenjie Tang, and Yuchen Kang. 2025. TeleAntiFraud-28k: An audio-text slow-thinking dataset for telecom fraud detection. In *Proceedings of the 33rd ACM International Conference on Multimedia (MM '25)*, pages 1–9, New York, NY, USA. ACM.

OpenAI. 2025. Openai o3-mini.

Biqing Qi, Xinquan Chen, Junqi Gao, Dong Li, Jianxing Liu, Ligang Wu, and Bowen Zhou. 2024. Interactive continual learning: Fast and slow thinking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12882–12892.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Maohao Shen, Guangtao Zeng, Zhenting Qi, Zhang-Wei Hong, Zhenfang Chen, Wei Lu, Gregory Wornell, Subhro Das, David Cox, and Chuang Gan. 2025a. Satori: Reinforcement learning with chain-of-action-thought enhances llm reasoning via autoregressive search. *arXiv preprint arXiv:2502.02508*.

Zitong Shen, Sineng Yan, Youqian Zhang, Xiapu Luo, Grace Ngai, and Eugene Yujun Fu. 2025b. " it warned me just at the right moment": Exploring llm-based real-time detection of phone scams. *arXiv preprint arXiv:2502.03964*.

Gurjot Singh, Prabhjot Singh, and Maninder Singh. 2025. Advanced real-time fraud detection using rag-based llms. *arXiv preprint arXiv:2501.15290*.

Maciej Świechowski, Konrad Godlewski, Bartosz Sawicki, and Jacek Mańdziuk. 2023. Monte carlo tree search: A review of recent modifications and applications. *Artificial Intelligence Review*, 56(3):2497–2562.

DOUBAO TEAM. 2025. Doubao-1.5-pro.

Omkar Thawakar, Dinura Dissanayake, Ketan More, Ritesh Thawkar, Ahmed Heakl, Noor Ahsan, Yuhao Li, Mohammed Zumri, Jean Lahoud, Rao Muhammad Anwer, and 1 others. 2025. Llamav-o1: Rethinking step-by-step visual reasoning in llms. *arXiv preprint arXiv:2501.06186*.

Luong Trung, Xinbo Zhang, Zhanming Jie, Peng Sun, Xiaoran Jin, and Hang Li. 2024. Reft: Reasoning with reinforced fine-tuning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7601–7614.

Zihan Wang, Xingle Xu, Hao Wang, Yiwen Ye, Yuchen Li, Linhao Wang, Hongze Tan, Peidong Wang, Shi Feng, Guoqing Chen, and 1 others. 2025. A survey on entropy mechanism in large reasoning models. *Authorea Preprints*.

Guowei Xu, Peng Jin, Li Hao, Yibing Song, Lichao Sun, and Li Yuan. 2024. Llava-o1: Let vision language models reason step-by-step. *arXiv preprint arXiv:2411.10440*.

Shu Yang, Shenzhe Zhu, Zeyu Wu, Keyu Wang, Junchi Yao, Junchao Wu, Lijie Hu, Mengdi Li, Derek F Wong, and Di Wang. 2025. Fraud-r1: A multi-round benchmark for assessing the robustness of llm against augmented fraud and phishing inducements. *arXiv preprint arXiv:2502.12904*.

Aohan Zeng, Zhengxiao Du, Mingdao Liu, Kedong Wang, Shengmin Jiang, Lei Zhao, Yuxiao Dong, and Jie Tang. 2024. Glm-4-voice: Towards intelligent and human-like end-to-end spoken chatbot. *arXiv preprint arXiv:2412.02612*.

Xinyu Zhu, Junjie Wang, Lin Zhang, Yuxiang Zhang, Ruyi Gan, Jiaxing Zhang, and Yujiu Yang. 2022. Solving math word problems via cooperative reasoning induced language models. *arXiv preprint arXiv:2210.16257*.

A Computational Efficiency Analysis

To validate that our theoretical reductions in reasoning token counts translate into practical wall-clock time savings, we conducted a detailed profiling of inference latency and throughput.

Experimental Setup. All efficiency evaluations were performed on NVIDIA A100 GPUs. We measured the performance of our three iterative models: SAFE-RL, SAFE-RS, and SAFE-LS. To ensure consistent and reproducible measurements, we utilized greedy decoding (temperature $= 0$). We report the median latency ($p50$), the 95th percentile latency ($p95$), and the overall system throughput (samples per second).

Results. As presented in Table 5, the optimization stages demonstrate a clear trajectory of efficiency improvement. SAFE-RL, which produces the longest reasoning chains, exhibits the highest latency. The Rejection Sampling stage (SAFE-RS) provides a moderate improvement by filtering out inherently long-winded responses. Most notably, the final Length-Constrained RL stage (SAFE-LS) achieves a $p50$ latency of 916.2ms and a throughput of 1.10 samples/s.

This corresponds to a $\approx 26.34\%$ reduction in median latency compared to the SAFE-RL baseline. These results confirm that the "Thinking Efficiency Metric" (TEM) discussed in Section 4.3 correlates strongly with real-world deployment metrics. The substantial drop in $p95$ latency (from 1895.7ms to 1207.8ms) also indicates that SAFE-LS is significantly more stable and robust against "infinite loops" or excessive overthinking, making it more suitable for time-sensitive fraud detection scenarios.

B Prompts

The prompts used in this study are designed to guide the model through various stages of reasoning and decision-making. Below is a detailed description of their roles:

- Figure 6 illustrates the prompts utilized in the first and second stages for training across three tasks: scenario classification, fraud detection, and fraud type classification. These prompts are structured to facilitate slow-thinking reasoning, enabling the model to capture subtle discrepancies in audio details, such as vocal tone fluctuations, emotional stress, and environmental cues.

Table 5: Inference Latency and Throughput Profiling on NVIDIA A100. Comparison across the three training stages shows that our Long2Short optimization (SAFE-LS) significantly reduces latency and improves throughput.

Model	p50	p95	Throughput
SAFE-RL	1243.9	1895.7	0.76
SAFE-RS	1204.0	1263.4	0.85
SAFE-LS	**916.2**	**1207.8**	**1.10**

- Figure 7, Figure 8, and Figure 9 present the prompts employed during the real-time detection phase. These prompts dynamically adjust based on the conversation phase (early, late, or final) to ensure timely and accurate fraud detection while considering the sufficiency of available information.
- Figure 10 showcases the prompts designed for evaluating the quality of the model's reasoning process. These prompts focus on assessing logical rigor, practical value, and expression quality, providing a comprehensive evaluation framework for the model's performance.

C Hyperparameter Settings and Sensitivity Analysis

To ensure reproducibility and facilitate further research, we provide a comprehensive, detailed description of our hyperparameter configurations, including the rationale behind specific choices and sensitivity analyses conducted during the development of SAFE-QAQ.

C.1 Implementation Platform

All experiments were conducted on a high-performance computing cluster equipped with 4 NVIDIA A100 (80GB) GPUs. We utilized a global batch size of 12 (implemented as a per-device batch size of $bs = 3$ with gradient accumulation). The training framework is built upon `ms-swift`[1], optimized for efficient large audio-language model fine-tuning.

C.2 Configuration Rationale

Our hyperparameter selection strategy balances training stability, inference efficiency, and task performance. The specific configurations are categorized as follows:

[1] https://github.com/modelscope/ms-swift

Reward Weight Configuration. The composite reward functions involve multiple components $(\alpha, \beta, \delta, \lambda)$. We determined their values based on the priority of objectives:

- **Accuracy Priority ($\alpha = 5, \beta = 1$):** We set the ratio $\alpha : \beta = 5 : 1$. This heavy weighting on α ensures that the model prioritizes the correctness of the final classification (Accuracy Reward) over mere structural compliance (Format Reward), while $\beta = 1$ remains sufficient to guide the parser.
- **Phase Awareness ($\delta = 5$):** For Real-Time Fine-Tuning, accurate phase recognition is critical for timely intervention. We set $\delta = 5$, equal to the accuracy weight α, to emphasize that identifying the correct conversation phase is as substantial as the fraud detection itself in live scenarios.
- **Efficiency Balance ($\lambda = 1$):** We set $\lambda = 1$ to introduce a regularization term for reasoning length. This value was chosen to curb verbosity without overpowering the accuracy reward, preventing the model from sacrificing necessary reasoning depth for brevity.

Reinforcement Learning (GRPO) Parameters. We adopt the Group Relative Policy Optimization (GRPO) algorithm.

- **Stability Factors ($\epsilon = 0.2, \beta_{KL} = 0.04$):** We adhere to the default settings recommended by the `ms-swift` framework. Specifically, the clipping coefficient $\epsilon = 0.2$ and the KL divergence coefficient $\beta_{KL} = 0.04$ are crucial for preventing policy collapse. In our preliminary experiments, we explored removing the KL penalty (i.e., $\beta_{KL} = 0$), which resulted in severe training instability and mode collapse. Thus, we retained the robust default values.
- **Group Size ($G = 9$):** We set the group size to 9. This value represents a trade-off between computational overhead and gradient variance reduction, ensuring stable convergence within limited GPU memory.

Generation and Length Constraints.

- **Sampling Strategy:** To balance generation diversity and quality during exploration, we utilize Nucleus Sampling with $top\_p = 0.9$,

$top\_k = 50$, and a temperature of 0.9. For Rejection Sampling, we set the number of candidates $K = 16$ to ensure sufficient coverage of the solution space.

- **Length Thresholds:** The maximum length threshold $L_{max} = 200$ and threshold $L_{threshold} = 200$ for R_{depth} and R_{eff} were determined based on the statistical distribution of reasoning chains in the TeleAntiFraud-28k dataset. $P^{max} = 5$ is set to cap the penalty magnitude, preventing excessive gradients that could destabilize the policy.

C.3 Hyperparameter Sensitivity Analysis

To validate our choice of learning rate, which is a critical factor in RL convergence, we conducted a grid search using the SAFE-RL (w/o SFT) model on a synthetic subset of TeleAntiFraud-Bench. The results are summarized in Table 6.

Table 6: Sensitivity analysis of Learning Rate (LR) on model performance. The selected setting ($3e^{-5}$) achieves the best balance across all metrics.

Learning Rate	Sce.	Fra.	FT.	AVG
$1e^{-5}$	85.34	72.86	75.36	77.85
$3e^{-5}$ (Ours)	84.31	**88.70**	75.74	**82.92**
$5e^{-5}$	85.12	78.10	**76.56**	79.93

As observed, a learning rate of $3e^{-5}$ yields the highest average F1 score (AVG: 82.92). Lower rates ($1e^{-5}$) resulted in underfitting, particularly in the Fraud Detection (Fra.) task, while higher rates ($5e^{-5}$) degraded performance, likely due to optimization overshooting. Consequently, $lr = 3e^{-5}$ was selected for all main experiments.

D LLM Judge Validation

D.1 Judge Configuration

For quality assessment of reasoning chains, we employ **DeepSeek-R1** as the LLM judge with the following parameters: Temperature = 0.6, Top-p = 0.95, Top-k = 50. To reduce sampling variance, the final score for each sample is computed as the average of 3 independent evaluations.

D.2 Correlation Analysis

To validate that our LLM-based quality assessment (SUM scores) reflects genuine reasoning quality rather than superficial stylistic artifacts, we conducted a correlation analysis between the judge scores and objective metrics.

D.3 Human Alignment and Robustness

To further verify that the judge is aligned with expert assessment rather than superficial fluency, we evaluated a diverse subset of 100 instances with both the LLM judge and 5 independent anti-fraud domain experts. The inter-rater reliability among the human experts reaches 0.78 under ICC, indicating substantial agreement. Table 7 summarizes the rank correlation between human ratings and the LLM judge across all reasoning dimensions. The consistently strong correlations show that the judge tracks expert preferences well, while the weak score-length correlation reported below indicates that these gains do not come from rewarding longer responses.

Dimension	Spearman Correlation
Logical	0.82
Practical	0.79
Expression	0.75
Overall SUM	0.81

Table 7: Alignment between the LLM judge and 5 anti-fraud domain experts on a 100-instance subset. All reported correlations are statistically significant.

Score–Accuracy Correlation. We observe a strong positive correlation between the LLM judge's SUM scores and objective classification correctness (Spearman's $\rho = 0.77$, Pearson's $r = 0.84$, $p < 0.001$). This confirms that higher-quality reasoning chains as evaluated by the judge correspond to more accurate fraud detection outcomes, validating SUM as a meaningful proxy for reasoning capability rather than a stylistic artifact.

Score–Length Correlation. The correlation between LLM judge scores and reasoning chain token length is weak ($r = 0.18$), indicating that the judge evaluates reasoning quality rather than rewarding verbosity. This confirms that the SUM score improvements observed across our SAFE models reflect genuine advances in logical rigor and practical value, not mere increases in output length.

We also analyze whether judge-based reasoning scores align with objective task performance at the model level. Table 8 reports the correlation between AVG and each judged dimension across all evaluated models. The consistently strong positive correlations further support that improvements in SUM reflect reasoning quality associated with better task execution rather than stylistic variation alone.

Figure 5 presents scatter plots illustrating these

Metric Pair	Spearman	Pearson
AVG vs. Logical	0.7679	0.8105
AVG vs. Practical	0.7714	0.8339
AVG vs. Expression	0.6917	0.8412
AVG vs. SUM	0.7703	0.8393

Table 8: Correlation between objective task performance and LLM-judged reasoning quality across the evaluated models. All reported correlations are statistically significant.

two correlations.

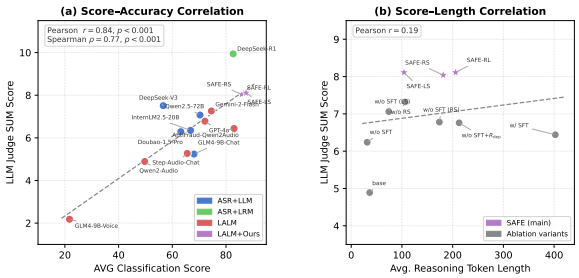

Figure 5: Scatter plots showing (a) the strong positive correlation between LLM judge SUM scores and classification correctness (Spearman's $\rho = 0.77$, Pearson's $r = 0.84$, $p < 0.001$), and (b) the weak correlation with reasoning chain length ($r = 0.18$). These results confirm that SUM score improvements reflect genuine reasoning quality rather than stylistic artifacts.

E Depth Reward Gating for SFT Models

Our main training setup disables the depth reward for SFT-initialized models because their reasoning traces are already substantially longer than the reward's effective range, making the reward nearly constant during optimization. To verify that this design choice does not hide an unrealized gain, we perform an explicit ablation in Table 9. The results show that enabling the depth reward on top of the SFT initialization preserves essentially the same precision, recall, and F1 across all three tasks while marginally increasing the average reasoning length. This supports the practical choice used in the main experiments: for SFT-initialized models, the depth reward adds extra verbosity without producing a meaningful performance benefit.

F Extended Baseline: Latest Proprietary Models

To ensure our evaluation remains current, we benchmarked the latest generation of Gemini models on TeleAntiFraud-Bench following the same evaluation protocol as the main experiments. As shown in Table 10, SAFE-LS consistently outperforms both Gemini-3.0-Pro and Gemini-3.0-Flash across all classification tasks, achieving a 3.55-point absolute improvement in AVG over the stronger Gemini-3.0-Pro (87.49 vs. 83.94). This further validates the effectiveness of our end-to-end RL-based framework against state-of-the-art proprietary systems.

G Audio Ablation

To isolate the contribution of raw audio, we construct text-only variants by removing the audio input and retaining only the textual content under the same task setup. Table 11 shows that both the SFT baseline and our final SAFE-LS model degrade substantially without audio, confirming that the end-to-end audio pathway contributes complementary evidence beyond transcript semantics alone. The effect is especially pronounced for fraud detection, where vocal cues and background acoustics provide strong signals that are not preserved in text-only inputs.

H Preliminary Generalization to an Emerging Fraud Pattern

Because full leave-one-type-out retraining is computationally expensive for the complete RL pipeline, we provide a preliminary out-of-distribution evaluation on a genuinely emerging fraud pattern observed after the benchmark was constructed: the *screen-sharing scam*. This tactic was not part of the seven benchmark fraud categories used in the main experiments. On 2,297 real-world instances collected from production traffic, SAFE-LS achieves strong performance, as summarized in Table 12. We present this result as preliminary evidence that the model learns transferable fraud indicators beyond memorizing fixed scripts, while leaving broader systematic OOD evaluation for future work.

I False Positive / False Negative Analysis

Real-world fraud detection requires careful consideration of the trade-off between False Negatives (FN, missed fraud) and False Positives (FP, unnecessary alerts). We analyze this on a manually verified subset of 10,000 records drawn from the high-risk call pool processed by our deployed system.

Overall Performance. SAFE-LS achieves a **Recall of 98.00%** and **Precision of 88.24%** on this

Depth Reward	Scenario			Fraud			Fraud Type			Avg. Tokens
	Pre.	Rec.	F1	Pre.	Rec.	F1	Pre.	Rec.	F1	
Off	82.93	80.97	81.57	85.70	95.19	90.20	87.78	88.07	87.25	205.76
On	82.96	81.00	81.60	85.74	95.23	90.24	87.60	87.89	87.07	207.72

Table 9: Ablation on enabling the depth reward for the SFT-initialized SAFE-RL model. The reward has negligible impact on classification metrics and slightly increases reasoning length.

Model	Sce.	Fra.	FT.	AVG
Gemini-3.0-Pro	83.64	81.93	86.25	83.94
Gemini-3.0-Flash	81.73	72.50	87.05	80.43
SAFE-LS	**84.64**	**89.61**	**88.23**	**87.49**

Table 10: Comparison with the latest Gemini models on TeleAntiFraud-Bench. SAFE-LS outperforms both proprietary models across all tasks.

Model	Scene	Fraud	Type	Avg F1
AntiFraud-Qwen2Audio	81.31	84.78	82.91	83.00
w/o Audio (Text-only)	71.55 (-9.76)	71.25 (-13.53)	77.93 (-4.98)	73.58 (-9.42)
SAFE-LS	84.64	89.61	88.23	87.49
w/o Audio (Text-only)	76.00 (-8.64)	77.50 (-12.11)	84.00 (-4.23)	79.17 (-8.32)

Table 11: Audio ablation on TeleAntiFraud-Bench. Removing raw audio consistently reduces performance for both the domain-adapted backbone and the final SAFE-LS model.

production subset, reflecting the system's deliberate optimization toward minimizing missed fraud.

False Negative Rate (FNR). The FNR is **2.00%**: approximately 3 out of 143 actual fraudulent calls are missed. Given that undetected fraud directly causes financial loss, we prioritize minimizing FNR via our reward design (high accuracy weight $\alpha = 5$), accepting a moderate FP rate in exchange.

False Positive Rate (FPR). The FPR is $\approx$**0.19%**: approximately 19 out of 9,857 legitimate calls are incorrectly flagged. In our cascaded deployment pipeline ($\sim$200M calls $\rightarrow$ rule-based filtering $\rightarrow$ $\sim$70K high-risk $\rightarrow$ SAFE-LS $\rightarrow$ manual verification), false positives are subsequently reviewed by human auditors, limiting their real-world impact to a modest increase in manual workload rather than direct harm to users.

This asymmetric design—prioritizing recall over precision—aligns with the operational requirements of telecom fraud prevention, where the cost of a missed fraud far exceeds that of an unnecessary review.

Metric	Value
Precision	87.96
Recall	97.30
F1	92.39

Table 12: Preliminary evaluation of SAFE-LS on an emerging out-of-distribution fraud pattern, *screen-sharing scam*, using 2,297 real-world instances. Values are reported as percentages.

Metric	Value
Precision	88.24
Recall	98.00
False Positive Rate	0.19
False Negative Rate	2.00

Table 13: Operating-point statistics of SAFE-LS on a manually verified 10,000-call subset from the production high-risk pool. Values are reported as percentages.

Prompt for Non-Real-Time

Task_1 Description:
Please analyze in detail which of the following types the call scenario belongs to based on the audio content you heard:
- Food ordering service
- Customer service inquiry
- Appointment service
- Transportation inquiry
- Daily shopping
- Ride-hailing service
- Food delivery service

Input:

Call record

Output Format:

Please provide your judgment result in the following format:
<think>
...(Detailed analysis process)
</think>
<answer>
```
{
"scene": "<scene_type>",
"reason": "<reason_for_judgment>",
"confidence": <confidence_level>
}
```
</answer>
- `scene`: A string indicating the identified call scenario type, must be chosen from the following closed set: `["Food ordering service", "Customer service inquiry", "Appointment service", "Transportation inquiry", "Daily shopping", "Ride-hailing service", "Food delivery service"]`.
- `reason`: A string providing a brief explanation of the reason for your judgment.
- `confidence`: A float indicating your confidence level in the judgment, ranging from 0 to 1, with 1 indicating complete confidence.

Task_2 Description:
You are a professional audio large language model capable of analyzing audio content in detail and determining whether it involves fraud. Based on the following information, provide your judgment:
1. The call scenario analyzed in the first round.
2. The audio content.

Output Format:

Please output your judgment result in the following format:
<think>
...(Detailed analysis process)
</think>
<answer>
```
{
  "reason": "<reason_for_judgment>",
  "confidence": <confidence_level>,
  "is_fraud": <true/false>
}
```
</answer>
- `reason`: A string briefly explaining the reason for your judgment.
- `confidence`: A float indicating your confidence level in the judgment, ranging from 0 to 1, with 1 indicating complete confidence.
- `is_fraud`: A boolean value indicating whether the audio involves fraud. `true` indicates fraud, and `false` indicates no fraud.

Task_3 Description:
You are a professional audio large language model capable of analyzing audio content in detail and determining the type of fraud involved. Based on the following information, provide your judgment:
1. The call scenario analyzed in the first round.
2. The second-round analysis of whether the call involves fraud.
3. The audio content.

Output Format:

Please provide your judgment result in the following format:
<think>
...(Detailed analysis process)
</think>
<answer>
```
{
  "fraud_type": "<fraud_type>",
  "reason": "<reason_for_judgment>",
  "confidence": <confidence_level>
}
```
</answer>
- `fraud_type`: A string indicating the identified fraud type, which must be chosen from the following closed set: `["Investment Fraud", "Phishing Fraud", "Identity Theft", "Lottery Fraud", "Bank Fraud", "Kidnapping Fraud", "Customer Service Fraud", "Email Fraud"]`.
- `reason`: A string briefly explaining the reason for your judgment.
- `confidence`: A float indicating your confidence level in the judgment, ranging from 0 to 1, with 1 indicating complete confidence.

Figure 6: Prompt for Non-Real-Time.

Prompt for Real-Time Scenario Classification

Task Description:
Please analyze the content of the audio you heard in detail and determine which of the following types the call scenario belongs to, as well as the stage of the call.

Scene Types:

- Food Ordering Service
- Customer Service Inquiry
- Appointment Service
- Transportation Inquiry
- Daily Shopping
- Ride-hailing Service
- Food Delivery Service

Input:

Audio clip of the call

Output Format:

Please strictly follow the format below to output your judgment result:

```
<think>
...(Detailed analysis process, including whether there is enough information to support the conclusion)
</think>
<answer>
...(Specific judgment content, format shown below)
</answer>
```

1. **`<think>` Section:**
   - Detailed analysis of the audio content, including:
     - Understanding of the call scenario
     - Extraction of key information
     - Evaluation of whether there is sufficient information to support the conclusion
     - Assessment of the current stage of the call
   - Must be wrapped with `<think>` and `</think>` tags.

2. **`<answer>` Section:**
   - Output the specific judgment result based on the analysis.
   - Must be wrapped with `<answer>` and `</answer>` tags.
   - Output a JSON object in the following format:
     ```json
     {
       "conversation_stage": "<stage>",
       "scene": "<scene_type>|null",
       "reason": "<reason_for_judgment>",
       "confidence": <confidence_level>
     }
     ```
     - `conversation_stage`: A string indicating the stage of the call, must be chosen from the following closed set:
       - `"early_stage"` (The first half of the call, where the intent of the conversation has not been fully revealed)
       - `"late_stage"` (The latter half of the call, nearing the end of the conversation)
       - `"complete"` (Full call record)
     - `scene`: A string or null, indicating the identified call scenario type, must be chosen from the following closed set or be null: `["Food Ordering Service", "Customer Service Inquiry", "Appointment Service", "Transportation Inquiry", "Daily Shopping", "Ride-hailing Service", "Food Delivery Service", null]`.
       - When `conversation_stage` is `"complete"`, this field cannot be null.
       - When `conversation_stage` is `"late_stage"`, a judgment should be provided if possible.
       - When `conversation_stage` is `"early_stage"`, this field can be null.
     - `reason`: A string briefly explaining the reason for your judgment, including an analysis of the call stage.
     - `confidence`: A float indicating your confidence level in the judgment, ranging from 0 to 1, with 1 indicating complete confidence.

3. **Notes:**
   - The overall output format must be strictly followed:
     ```
     <think>
     ...(Detailed analysis process)
     </think>
     <answer>
     ...(Specific judgment content)
     </answer>
     ```
   - Both the `<think>` and `<answer>` sections must exist, and their order cannot be reversed.
   - The JSON object must contain all specified fields.
   - The judgment of the call stage should be explained in detail in the `<think>` section.

Figure 7: Prompt for Real-Time Scenario Classification.

Prompt for Real-Time Fraud Detection

````
**Task Description:**
Based on the audio content you hear, analyze it in detail and determine whether it involves fraud and the stage of the call. Please provide your judgment based on the following information:

1. The call scenario analyzed in the first round.
2. The audio content.

**Output Format:**

Please strictly follow the format below to output your judgment results:

```
<think>
...(Detailed analysis process, including whether there is sufficient information to make a judgment)
</think>
<answer>
...(Specific judgment content, see format below)
</answer>
```

1. **`<think>` Section:**
   - Detailed analysis of the audio content, including:
     - Understanding of the call scenario
     - Extraction of key information
     - Determination of whether there is enough information to support the conclusion
     - Evaluation of the current stage of the call
   - Must be wrapped with `<think>` and `</think>` tags.

2. **`<answer>` Section:**
   - Based on the analysis results, output the specific judgment content.
   - Must be wrapped with `<answer>` and `</answer>` tags.
   - Output a JSON object in the following format:
     ```json
     {
       "conversation_stage": "<stage>",
       "reason": "<reason_for_judgment>",
       "confidence": <confidence_level>,
       "is_fraud": <true/false>|null
     }
     ```
     - `conversation_stage`: A string indicating the stage of the call, which must be chosen from the following closed set:
       - `"early_stage"` (The first half of the call, where the intent of the conversation has not been fully revealed)
       - `"late_stage"` (The second half of the call, nearing its conclusion)
       - `"complete"` (A full call record)
     - `reason`: A string briefly explaining the reason for your judgment.
     - `confidence`: A float indicating your confidence level in the judgment, ranging from 0 to 1, with 1 indicating complete confidence.
     - `is_fraud`: A boolean value or null, indicating whether the audio involves fraud. `true` indicates fraud, `false` indicates no fraud. If there is insufficient information, it should be null.
       - When `conversation_stage` is `"complete"`, this field cannot be null.
       - When `conversation_stage` is `"late_stage"`, a judgment should be provided if possible.
       - When `conversation_stage` is `"early_stage"`, it can be null.

3. **Notes:**
   - You must strictly adhere to the following overall output format:
     ```
     <think>
     ...(Detailed analysis process)
     </think>
     <answer>
     ...(Specific judgment content)
     </answer>
     ```
   - Both the `<think>` and `<answer>` sections must exist, and their order cannot be reversed.
   - The JSON object must contain all specified fields.
   - The judgment of the call stage should be explained in detail in the `<think>` section.
````

Figure 8: Prompt for Real-Time Fraud Detection.

Prompt for Real-Time Fraud Type Classification

Task Description:

Based on the audio content you hear, analyze it in detail and determine the type of fraud involved as well as the stage of the call. Please provide your judgment based on the following information:

1. The call scenario analyzed in the first round.
2. The analysis of whether the call involves fraud in the second round.
3. The audio content.

Output Format:

Please strictly follow the format below to output your judgment:

```
<think>
...(Detailed analysis process, including whether there is enough information to support the conclusion)
</think>
<answer>
...(Specific judgment content, see format below)
</answer>
```

1. **`<think>` Section:**
   - Provide a detailed analysis of the audio content, including:
     - Understanding of the call scenario.
     - Extraction of key information.
     - Assessment of whether there is sufficient information to support the conclusion.
     - Evaluation of the current stage of the call.
   - Must be enclosed in `<think>` and `</think>` tags.

2. **`<answer>` Section:**
   - Based on the analysis results, output the specific judgment content.
   - Must be enclosed in `<answer>` and `</answer>` tags.
   - Output a JSON object in the following format:
     ```json
     {
       "conversation_stage": "<stage>",
       "fraud_type": "<fraud_type>|null",
       "reason": "<reason_for_judgment>",
       "confidence": <confidence_level>
     }
     ```
     - `conversation_stage`: A string indicating the stage of the call, must be chosen from the following closed set:
       - `"early_stage"` (The first half of the call, where the intent of the conversation has not been fully revealed).
       - `"late_stage"` (The latter half of the call, where the conversation is nearing its end).
       - `"complete"` (A full record of the call).
     - `fraud_type`: A string or null, indicating the identified fraud type. Must be chosen from the following closed set or null: `["Investment Fraud", "Phishing Fraud", "Identity Theft", "Lottery Fraud", "Bank Fraud", "Kidnapping Fraud", "Customer Service Fraud", "Email Fraud", null]`.
       - When `conversation_stage` is `"complete"`, this field cannot be null.
       - When `conversation_stage` is `"late_stage"`, a judgment should be provided if possible.
       - When `conversation_stage` is `"early_stage"`, this field can be null.
     - `reason`: A string briefly explaining the reason for your judgment, including an analysis of the call stage.
     - `confidence`: A float indicating your confidence level in the judgment, ranging from 0 to 1, with 1 indicating complete confidence.

3. **Notes:**
   - The overall output format must be strictly followed:
     ```
     <think>
     ...(Detailed analysis process)
     </think>
     <answer>
     ...(Specific judgment content)
     </answer>
     ```
   - Both the `<think>` and `<answer>` sections must exist, and their order cannot be reversed.
   - The JSON object must contain all specified fields.
   - The judgment of the call stage should be explained in detail in the `<think>` section.

Figure 9: Prompt for Real-Time Fraud Type Classification.

Prompt for Evaluating the Thought Process

```
Please evaluate the model's reasoning process probabilistically based on three dimensions: **logical consistency, practicality, and clarity**. For each scoring criterion, calculate the probability of achieving the corresponding score, and compute the final score as the expected value.

---

#### **Input**
1. Model's reasoning process: {reasoning_process}
2. Model's final answer: {model_answer}
3. Reference answer: {reference_answer}
4. Reference reasoning process: {reference_reasoning}

---

### Scoring Rules
**1. Logical Rigor (5 points)**
  ☐ Complete reasoning chain without gaps (0-1 point)
    - Probability of earning 1 point: __%
  ☐ Reasonable and explicit key assumptions (0-1 point)
    - Probability of earning 1 point: __%
  ☐ Strictness of conclusion derivation (0-2 points)
    - Probability of earning 2 points: __%
    - Probability of earning 1 point: __%
  ☐ Advantage probability compared to reference reasoning (0-1 point)
    - Probability of earning 1 point: __%

**2. Practical Value (5 points)**
  ☐ Accuracy in identifying the essence of the problem (0-1 point)
    - Probability of earning 1 point: __%
  ☐ Effectiveness of the solution (0-2 points)
    - Probability of earning 2 points: __%
    - Probability of earning 1 point: __%
  ☐ Completeness of addressing requirements (0-1 point)
    - Probability of earning 1 point: __%
  ☐ Optimization probability compared to reference reasoning (0-1 point)
    - Probability of earning 1 point: __%

**3. Expression Quality (5 points)**
  ☐ Completeness of presenting key nodes (0-1 point)
    - Probability of earning 1 point: __%
  ☐ Clarity of expression (0-2 points)
    - Probability of earning 2 points: __%
    - Probability of earning 1 point: __%
  ☐ Brevity of information (0-1 point)
    - Probability of earning 1 point: __%
  ☐ Expression advantage compared to reference reasoning (0-1 point)
    - Probability of earning 1 point: __%

---

### Probability Constraints
1. If the conclusion is incorrect:
  - Logical rigor: Probability of conclusion alignment drops to zero; upper limit for other items is 40%.
  - Practical value: Probabilities for problem identification and solution effectiveness drop to zero.
2. Logical leaps: Each instance reduces probabilities by 15-30%.
3. Missing evidence: Total probability for the corresponding dimension decreases by 20%.

---

### Output Template
**Logical Rigor**
- Complete reasoning chain without gaps (1-point probability): <__%>
- Reasonable and explicit key assumptions (1-point probability): <__%>
- Strictness of conclusion derivation (2 points → __% | 1 point → __%)
- Advantage probability compared to reference reasoning (1-point probability): <__%>
Expected score: <Calculation formula>

**Practical Value**
- Accuracy in identifying the essence of the problem (1-point probability): <__%>
- Effectiveness of the solution (2 points → __% | 1 point → __%)
- Completeness of addressing requirements (1-point probability): <__%>
- Optimization probability compared to reference reasoning (1-point probability): <__%>
Expected score: <Calculation formula>

**Expression Quality**
- Completeness of presenting key nodes (1-point probability): <__%>
- Clarity of expression (2 points → __% | 1 point → __%)
- Brevity of information (1-point probability): <__%>
- Expression advantage compared to reference reasoning (1-point probability): <__%>
Expected score: <Calculation formula>
```

Figure 10: Prompt for Evaluating the Thought Process.