

Adapting Technical-Service LLM Agents with Latent Logic Augmentation, Robust Noise Reduction, and Hybrid Reward Modeling

Junzhuo Ma^{1,3,*}, Chenghuang Shen^{2,3,*}, Yi Yu^{1,3,*}, Xingyan Liu^{3,*}, Jing Gu^{1,3},
Hangyi Sun^{1,3}, Guangquan Hu^{3,4}, Jianfeng Liu^{3,†}, Weiting Liu^{3,4}, Pu Mingyue³,
Wang Yu³, Zhengdong Xiao³, Rui Xie³, Longjiu Luo³, Qianrong Wang³,
Gurong Cui³, Honglin Qiao³, Wenlian Lu^{1,2,4,5,6,†}

¹School of Mathematical Sciences, Fudan University, Shanghai, China

²Shanghai Center for Mathematical Sciences, Fudan University, Shanghai, China

³Alibaba Group, Hangzhou, China

⁴Institute of Science and Technology for Brain-Inspired Intelligence,
Fudan University, Shanghai, China

⁵Center for Applied Mathematics & Shanghai Key Laboratory
of Contemporary Applied Mathematics, Fudan University, Shanghai, China

⁶Ji Hua Laboratory, Foshan, 528200, China

{23110180025,y_yu21,24210180034,24210180057,22110850002,wtliu24}@m.fudan.edu.cn
{chenghuang.sch,wxw,jiawei.ljf,pumingyue.pmy,xunzhen.wy,tianqiao.xzd,harry.xr,
xuanqan,qianrong.wqr,yuanchuo,kenny.qhl}@alibaba-inc.com; wenlian@fudan.edu.cn

Abstract

Technical-service LLM agents are entering production workflows, where value depends on whether engineers adopt generated replies. Service tickets hide decision logic, contain noisy single-reference responses, and make reward evaluation costly, making standard post-training brittle. Existing post-training and LLM-as-a-Judge approaches improve grounding or feedback, but do not jointly model latent decision logic, response diversity, and reward cost. We address this gap by coupling latent logic augmentation, robust noise reduction, and hybrid reward modeling. The framework augments supervised fine-tuning data with Planning-Aware Trajectory Modeling and Reasoning Augmentation, builds dual-filtered Multiple Ground Truths, and trains the policy with a hybrid reward that combines a Reranker with an LLM-as-a-Judge. On real Cloud technical-service tasks, the adapted Qwen3-4B achieves the highest Multi-ECS (0.441), lower reward cost, and the highest production adoption rate (46.63%).

1 Introduction

Large Language Model (LLM) agents are increasingly used across web, tool-use, and Cloud-service workflows (Deng et al., 2023; Zheng et al., 2024; Kimi Team et al., 2026; GLM-4.5 Team et al., 2025; Meituan LongCat Team et al., 2025; Yu

et al., 2026). In our Cloud technical-service setting, practical value is measured by whether generated replies are adopted in production. Inference-time adaptation through In-Context Learning (ICL) (Brown et al., 2020; Wei et al., 2022; Gemini Team et al., 2024; Agarwal et al., 2024; Zhang et al., 2026a) and Retrieval-Augmented Generation (RAG) (Lewis et al., 2020; Cuconasu et al., 2024; Mialon et al., 2023) helps reuse domain context, but improving adoption rate in complex Cloud technical-service scenarios typically requires parameter updates. The standard post-training involves Supervised Fine-Tuning (SFT) and Reinforcement Learning (RL), which exposes several coupled obstacles in practice (Figure 1).

First, conventional training and interaction data often expose final outputs or actions without explicitly modeling the intermediate reasoning that produced them (Zelikman et al., 2022, 2024; Li et al., 2026a). SFT, human preference alignment (Ouyang et al., 2022), Continual Pre-training (Su et al., 2026), and Parameter-Efficient Fine-Tuning (PEFT) (Hu et al., 2022) can therefore encourage myopic imitation instead of transferable decision reasoning. Second, single-reference supervision penalizes valid but different responses in semantically diverse service tasks, leading to output homogenization or “knowledge collapse” (Wright et al., 2026; Shypula et al., 2025; Jiang et al., 2025). This is not resolved by stronger optimizers alone: although alignment methods, including RLHF, PPO,

*Equal contribution. †Corresponding authors.

Observation 1: Logs hide the decision logic	Observation 2: Multiple valid answers per query	Observation 3: Reward fidelity vs. cost
<i>Logs record the expert’s final action, not the reasoning behind it.</i> Ticket: a LATERAL query runs on test data but fails on the real table. Logged action: “Re-run it against the real table and send the full log” — the expert <i>asks</i> instead of answering. Not logged: <i>why</i> — the failure is environment-specific, so answering directly would be wrong. $\Rightarrow$ surface imitation is myopic; we expose the latent logic with RA + PATM.	<i>A single query admits several valid expert replies, but the log stores only one.</i> User: “How can I tell which tables have already been synchronized?” All valid: (a) “Using an offline sync task?”; (b) “Check the Data Source Management page”; (c) “Data Development > Offline Sync Nodes”. Logged gold: only (a) — so (b) and (c) get scored as <i>wrong</i>. $\Rightarrow$ single-reference supervision is noisy; we recover the valid set with Multi-GT.	<i>A lexical reranker is cheap but gameable, while a thinking LLM judge is more accurate yet $\sim 10\times$ costlier.</i> User: “Can I audit a single index?” Logged gold (clarify): “Do you mean auditing update/delete operations on a single index?” Agent reply (answer): “You can’t audit a single index—audit the whole cluster instead.” ✗ Reranker: <i>consistent</i>—fooled by the shared “audit... single index” wording. ✓ Judge: <i>inconsistent</i>—gold clarifies, reply answers. $\Rightarrow$ accurate at lower cost: Hybrid Reward.

Figure 1: **Three observations that motivate our framework**, each grounded in a real Cloud technical-service ticket, mapping a pain point to the component that addresses it: hidden decision logic (**RA + PATM**), multiple valid answers for one query (**Multi-GT**), and a gameable cheap Reranker vs. an accurate costly judge (**Hybrid Reward**).

DPO, DAPO, and recent multi-reference, agent-RL, or multi-answer RL variants, can improve alignment (Ziegler et al., 2020; Schulman et al., 2017; Rafailov et al., 2023; Wu and Echarchaoui, 2025; Aminian et al., 2025; Yu et al., 2025; Xi et al., 2026; Yuan et al., 2026; Zhang et al., 2026c; Puri et al., 2026), their reward signals stay fragile when the space of valid responses is only partially represented. Third, LLM-as-a-Judge reward modeling provides scalable semantic evaluation (Liu et al., 2023; Zheng et al., 2023; Li et al., 2024; Yuan et al., 2024; Lee et al., 2024), but it is vulnerable to reward hacking and costly for large-scale online adaptation (Anwar et al., 2024; Coste et al., 2024; Eisenstein et al., 2024; Xu et al., 2024; Saha et al., 2026; Zhang et al., 2026b).

We propose an adaptation framework that addresses these obstacles with three coordinated components. **Latent Logic Augmentation** makes hidden decision structure explicit through *Planning-Aware Trajectory Modeling (PATM)* and *Reasoning Augmentation (RA)*, stabilizing SFT for planning-intensive tasks. **Robust Noise Reduction** builds a *Multiple Ground Truths (Multi-GT)* dataset through dual filtering, preserving distinct but valid solution paths. **Hybrid Reward Modeling (HRM)** combines an LLM-as-a-Judge signal with a lightweight Reranker, retaining reward fidelity while reducing training cost. We validate the framework on real-world Cloud technical-service tasks using ECS-based offline evaluation and production adoption rate, showing gains in stability, response diversity, and practical deployability. Matched Multi-GT DPO and adaptive judge-routing comparisons position the method against relevant alternatives. Transfer experiments on Llama-3.1-8B and Qwen3-32B test its generality across model sizes and families.

2 Related Work

LLM Adaptation and Latent Logic. LLM specialization spans inference-time methods such as ICL and RAG (Brown et al., 2020; Wei et al., 2022; Lewis et al., 2020) and parameter-update methods such as SFT, RLHF, PPO, DPO, and DAPO (Ouyang et al., 2022; Ziegler et al., 2020; Schulman et al., 2017; Rafailov et al., 2023; Yu et al., 2025). Recent agentic RL studies further examine generalization, dense process rewards, and self-evolving policy learning for interactive agents (Xi et al., 2026; Yuan et al., 2026; Zhang et al., 2026c; Li et al., 2026c). For technical-service agents, the central difficulty is not only optimization stability but also the missing logic in logged expert trajectories. Prior work injects future dialogue context, latent thought traces, reusable skills, or interaction histories to improve planning and reasoning (Zeng et al., 2023; Zelikman et al., 2022, 2024; Song et al., 2024; Huang et al., 2026a), often with environment-mediated feedback (Gandhi et al., 2024; Lin et al., 2024; Li et al., 2026b; Liu et al., 2026b). Our Latent Logic Augmentation instead structures offline training data with PATM and RA so the model learns decision reasoning before RL.

Diverse Valid Responses. Single-reference training and evaluation can mislabel valid alternatives as errors, especially when several replies can solve the same technical issue (Wright et al., 2026; Shypula et al., 2025; Liu et al., 2026a). Recent benchmarks and RL methods address this by introducing multiple references or multi-answer objectives (Jiang et al., 2025; Wu and Echarchaoui, 2025; Aminian et al., 2025; Yu et al., 2025; Puri et al., 2026). Noise-aware structured reasoning similarly filters irrelevant question/table evidence or supervises pruning trajectories to preserve answer-critical content (Ye et al., 2026a; Guo et al., 2026). Our Robust Noise Reduction follows this direction

but targets real-world service logs: dual filtering expands each logged response into a Multi-GT set used for both ECS evaluation and RL training.

Judge-Based Reward Modeling. LLM-as-a-Judge methods provide scalable semantic evaluation and reward generation (Liu et al., 2023; Zheng et al., 2023; Li et al., 2024; Yuan et al., 2024; Lee et al., 2024; Wen et al., 2026). However, judge rewards can be exploited by the policy, and ensemble or large-judge solutions increase online training cost (Ouyang et al., 2022; Anwar et al., 2024; Coste et al., 2024; Xu et al., 2024; Huang et al., 2026b; Zeng et al., 2026). Recent budget-aware judge allocation, adaptive judge routing, and step-wise model routing work similarly highlights the need to spend expensive reasoning computation selectively (Saha et al., 2026; Zhang et al., 2026b; Ye et al., 2026b,c). Our HRM addresses this issue by combining a strong LLM-as-a-Judge signal with a lightweight Reranker, reserving expensive judgment for cases where the fast signal is less reliable.

3 Methodology

In this section, we present our framework for adapting Large Language Models (LLMs) to complex technical-service domains (Figure 2). We first establish a Multi-GT paradigm for evaluation and data construction. On this foundation, adaptation comprises two stages: an **SFT** stage whose data we enrich with **Latent Logic Augmentation** (PATM and RA) to surface latent decision logic, followed by an **RL** stage driven by HRM that fuses an LLM-as-a-Judge with a lightweight Reranker.

3.1 Foundation: Multi-GT Data and Evaluation

The foundation of our framework is a Multi-GT paradigm designed to address the inherent ambiguity of technical-service tasks, where a single query can have multiple valid responses. Traditional evaluation, which uses a single logged agent response as the unique gold reference, is unreliable because it unfairly penalizes valid but distinct solutions. To overcome this, for a query q we expand the single gold reference into a set of valid responses $\mathcal{Y}^*(q) = \{y_1^*, \dots, y_m^*\}$ used for both evaluation and training.

Core Instruments: Consistency Judge and Utility Judge. Since manual annotation is impractical, we automate the construction of the Multi-GT

training set via a pre-computation pipeline. Central to our pipeline are two LLM-based judges:

(1) *Consistency Judge* evaluates whether a response follows business logic consistent with the expert (see prompt in Appendix K.1). Validated against 152 human expert annotations, it attains 91.5% accuracy and “almost perfect” agreement with human labels ($\kappa = 0.82$); full statistics (confidence intervals and benchmark comparisons) are reported in Appendix B.

(2) *Utility Judge* uses privileged context (summary of service ticket) to determine if an alternative response effectively resolves the customer’s issue. It captures valid solutions that differ from the history and achieves 83% alignment with human labels (see prompt in Appendix K.2).

Automated Construction via Dual-Filtering Expansion. We construct candidates for Multi-GT dataset through two complementary streams: (1) **Offline Exploration:** We use a lightweight model (e.g., Qwen3-4B) with high temperature ($T = 1.2$) to generate diverse candidates. This injects novel linguistic patterns and reasoning angles; (2) **Online Adaptation:** We harvest high-likelihood rollouts from a preliminary RL run of a policy model. This stream exposes “hard positive” responses favored by the model that are functionally valid but differ from the human reference.

All candidates undergo a *Dual-Filtering* process comprising a Consistency Judge and a Utility Judge: The Consistency Judge ensures policy adherence, while the Utility Judge validates the effectiveness of alternative solutions. Only candidates passing one of these filters are added to $\mathcal{Y}^*(q)$.

For transparency, we report the detailed dataset composition and the expansion breakdown by source in Appendix A (Table 4). Overall, Multi-GT expansion roughly doubles the number of references (e.g., Train: 5,120 $\rightarrow$ 10,127). For the test split, every reference added by either the Consistency or Utility branch also passes human review, with per-branch audit statistics and raw-vs.-cleaned results provided in Appendix A.

Evaluation metric: Single- and Multi-ECS. Our scorer is the Consistency Judge introduced above (validated at 91.5% human agreement); to reduce single-model bias we apply it as an *ensemble*, defining $J_{\text{con}}(q, y, y^*) \in [0, 1]$ as the mean consistency score between a candidate response y and a reference y^* for query q , averaged over four LLMs (DeepSeek-R1 (Guo et al., 2025), DeepSeek-V3.2 (DeepSeek-AI et al., 2025),

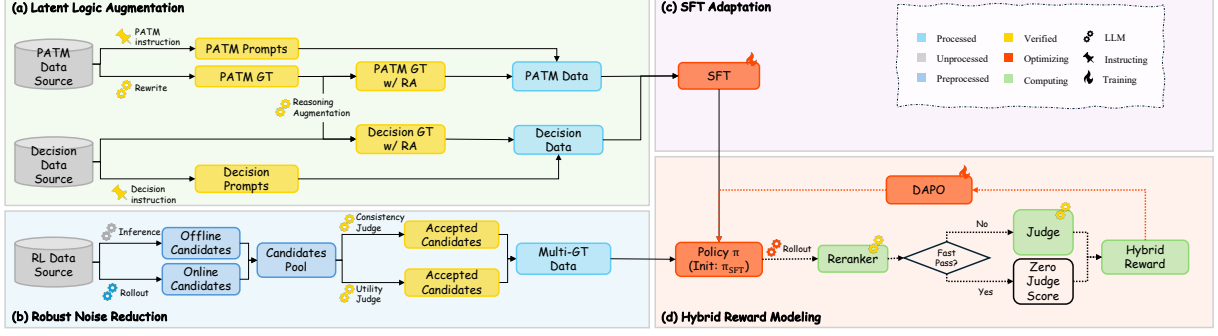


Figure 2: **Overview of the Proposed Framework.** The framework consists of four stages: (a) Latent Logic Augmentation; (b) Multi-GT data construction; (c) SFT adaptation; (d) RL adaptation with HRM.

Qwen3-Max (Qwen Team, 2025), and QwQ-Plus (Alibaba Cloud, 2026)). Scoring against the single logged reference y_0^* yields the **Single-ECS** (Single-GT Ensemble Consistency Score), $S_{\text{Single-ECS}}(q, y) = J_{\text{con}}(q, y, y_0^*)$, whereas taking the best match over the expanded set $\mathcal{Y}^*(q)$ yields the **Multi-ECS** (Multi-GT Ensemble Consistency Score):

$$S_{\text{Multi-ECS}}(q, y) = \max_{y^* \in \mathcal{Y}^*(q)} J_{\text{con}}(q, y, y^*). \quad (1)$$

We use Multi-ECS as the primary metric and Single-ECS to gauge alignment with the single logged reference.

3.2 SFT Stage: Latent Logic Augmentation

Standard SFT on expert trajectories often leads to myopic imitation and performance degradation in complex technical-service tasks. To instill latent decision reasoning, we augment training data through two methods. We model the interaction as a Markov Decision Process whose state is the dialogue context up to the current customer query q_t , whose action a_t is the agent’s response (a tool call or a textual reply), whose transition produces the environment’s next message q_{t+1} (from the customer or a tool), and whose reward scores the consistency of a_t . We then augment the training data to make these latent decision processes visible.

Reasoning Augmentation. To teach immediate reasoning, we turn raw single-step state-action pairs into a *decision* SFT dataset $\mathcal{D}_{\text{RA}}$. For each agent response a_t to query q_t , GPT-5.2 generates a “backward” chain-of-thought rationale c_t justifying the action, and the model is trained to predict the rationale and then the response:

$$\mathcal{L}_{\text{RA}} = -\mathbb{E}_{(q_t, c_t, a_t) \sim \mathcal{D}_{\text{RA}}} [\log p_{\theta}(c_t, a_t | q_t)].$$

This objective forces the model not just to mimic the action a_t , but to first generate the underlying thought process c_t , thereby internalizing the decision logic (see prompt in Appendix K.3.4).

Planning-Aware Trajectory Modeling. To equip the agent with foresight, we construct Planning-Aware trajectories for short future interactions. We extract three-step future sequences (a_t, q_{t+1}, a_{t+1}) following the current customer query q_t in the tickets, representing an agent response, corresponding environment response, and corresponding next agent response. GPT-5.2 rewrites this into a structured planning form $y_t^{\text{PATM}} = (q_t, \tilde{a}_t, \tilde{q}_{t+1}, \tilde{a}_{t+1})$, where $\tilde{a}_t$ is the predicted response to query q_t , $\tilde{q}_{t+1}$ is the predicted response of the environment (customer or tool) to $\tilde{a}_t$, and $\tilde{a}_{t+1}$ is the next predicted response to $\tilde{q}_{t+1}$. The learning objective autoregressively generates this trace:

$$\mathcal{L}_{\text{PATM}} = -\mathbb{E}_{y_t^{\text{PATM}} \sim \mathcal{D}_{\text{PATM}}} [\log p_{\theta}(\tilde{a}_t, \tilde{q}_{t+1}, \tilde{a}_{t+1} | q_t)].$$

These trajectories form the *planning* SFT set $\mathcal{D}_{\text{PATM}}$, and predicting them lets the model internalize the environment’s transition dynamics, moving beyond simple pattern matching. Because reasoning and planning are complementary, we also apply RA to the planning data, prepending the rationale c_t to obtain the reasoning-augmented trace $y_t^{\text{PATM+R}} = (q_t, c_t, \tilde{a}_t, \tilde{q}_{t+1}, \tilde{a}_{t+1})$; in practice the decision set $\mathcal{D}_{\text{RA}}$ and the planning set $\mathcal{D}_{\text{PATM}}$ are mixed for SFT. This combined trace decomposes into four coupled sub-objectives: (i) reasoning $\pi_{\theta}(c_t | q_t)$, (ii) policy execution $\pi_{\theta}(\tilde{a}_t | q_t, c_t)$, (iii) implicit world modeling $\mathcal{P}_{\theta}(\tilde{q}_{t+1} | q_t, c_t, \tilde{a}_t)$, and (iv) contingency planning $\pi_{\theta}(\tilde{a}_{t+1} | q_t, c_t, \tilde{a}_t, \tilde{q}_{t+1})$. Sub-objective (iii) is the core signal for capturing the stochastic environment dynamics (see prompts in Appendix K.3.1 and K.3.2).

3.3 RL Stage: Hybrid Reward Modeling

To refine the policy π_{SFT} obtained from SFT alignment, we address the challenge of reward sparsity and computational cost in reinforcement learning. During interactions, the agent primarily executes either tool invocations (`call_tool`) or textual replies (`reply`). Any action-type mismatch strictly yields a zero reward. For matched actions, while tool calls are deterministically scored via exact parameter matching, assessing the semantic consistency of free-form textual replies is highly subjective. Using powerful LLMs as judges for every RL rollout incurs significant latency. To optimize the trade-off between efficiency and fidelity, we propose an HRM that fuses a lightweight Reranker with an LLM-based Judge.

Components: Reranker and Judge. (1)

Reranker (S_R): Rather than relying on dense embeddings, we use an instruction-augmented Qwen3-4B *consistency* Reranker (non-thinking). It serves as a computationally cheap discriminator (S_R) that captures logical contradictions often missed by vector similarity (see Appendix J for a detailed comparison with embedding-based methods). (2) **LLM-based Judge (S_J):** A larger Qwen3-32B model (thinking-enabled) provides expert-level consistency scores, using the same prompt template as the Consistency Judge in the Dual-Filtering process. To stabilize training, we use a soft-score derived from token probabilities: $\text{Score} = P(\text{“Consistent”}) + 0.5 \cdot P(\text{“Partially”})$.

Single-Interval Cascade Strategy. Running the Qwen3-32B Judge for every RL rollout is prohibitively expensive, incurring approximately $10\times$ the inference latency compared with the lightweight Reranker. To mitigate this, we employ a cascade strategy $R_\theta(S_R, S_J)$ that approximates the oracle reward by combining the Reranker score S_R and the Judge score S_J . Within a “trust interval” $[\tau_a, \tau_b]$ the cheap Reranker is trusted directly (Fast Pass); outside it, the two scores are blended with mixing weights $w_1, w_2 \in [0, 1]$:

$$R_\theta(S_R, S_J) = \begin{cases} w_1 S_R + (1 - w_1) S_J, & S_R < \tau_a \text{ (Mix)} \\ S_R, & \tau_a \leq S_R \leq \tau_b \text{ (Fast Pass)} \\ w_2 S_R + (1 - w_2) S_J, & S_R > \tau_b \text{ (Mix)} \end{cases} \quad (2)$$

The cascade parameters $\theta = \{\tau_a, \tau_b, w_1, w_2\}$ are fitted once on a held-out set by maximizing the Spearman rank correlation between the cascade output $R_\theta(S_R, S_J)$ and the ensemble Consistency-Judge teacher score; the formal objective and the

fitted values are given in Appendix D.

By routing clear-cut samples (Fast Pass) to the fast Reranker and reserving the costly Judge for ambiguous cases, the HRM reduces the overall reward computation time while maintaining alignment fidelity.

4 Experiments

We evaluate the framework on a Qwen3-4B agent (Yang et al., 2025): after the setup (Sec. 4.1), we ablate the SFT and RL components—RA, PATM, the Hybrid Reward, and Multi-GT—on offline ECS (Sec. 4.2), compare with matched optimization and routing baselines and test transfer across backbones (Sec. 4.3), analyze reward calibration and ranking fidelity (Sec. 4.4), and report a paired production study of real adoption (Sec. 4.5).

4.1 Experimental Setup

Dataset & Metrics. Cloud platforms must resolve large volumes of technical-service tickets in which engineers diagnose issues, invoke diagnostic tools, and advise customers under tight correctness and latency constraints. We draw a proprietary technical-service dataset from this setting, with 10k queries each for the decision (RA) and planning (PATM) SFT sets (Sec. 3.2), alongside 5120/1k/1k queries for RL training/validation/test. To prevent leakage, complete tickets are split temporally before deriving RA, PATM, or Multi-GT examples; all turns from a ticket stay together, the SFT tickets are disjoint from the RL splits, and the preliminary policy uses only training tickets. PATM future turns serve only as training targets. Using our dual-filtering pipeline (Sec. 3.1), we expand the single logged reference into a Multi-GT dataset across all splits. We report **Multi-ECS** (primary) and **Single-ECS**, both defined in Sec. 3.1.

Training Setup & Baselines. For SFT strategies, we ablate the two latent-logic components—Reasoning Augmentation (RA) and Planning-Aware Trajectory Modeling (PATM)—giving the additive variants SFT, SFT+RA, SFT+PATM, and SFT+RA+PATM. In the RL stage, we employ the DAPO algorithm (Yu et al., 2025). For reward signals, we compare: (1) Reranker, (2) Hard Judge, (3) Soft Judge, and (4) **Hybrid Reward**. For data construction, we compare Single-GT RL (reward: the cascade score against the single logged reference) with our **Multi-GT** expansion (reward: the *maximum* cascade score over the validated set

Table 1: SFT strategy analysis (RQ1). RA and PATM each improve Multi-ECS and tool-use accuracy; their combination is best.

Method	Multi-ECS	Single-ECS	Call Tool Acc
Untrained Qwen3-4B	0.299	0.178	0.082
SFT	0.293	0.193	0.096
SFT+RA	0.319	0.224	0.139
SFT+PATM	0.326	0.233	0.149
SFT+RA+PATM	0.337	0.242	0.279

$\mathcal{Y}^*(q)$). We additionally adapt DPO (Rafailov et al., 2023) to the Multi-GT data (Multi-GT DPO) and implement RACER adaptive judge routing (Zhang et al., 2026b) with the same Qwen3-4B backbone, data, Judge service, and evaluation protocol. To ensure a fair comparison, all RL models are evaluated at the same fixed checkpoint after 20 episodes. Detailed configurations are provided in Appendix D and E.

4.2 Main Results

We isolate and validate the proposed components below, summarizing SFT phase results in Table 1 and RL phase results in Table 2.

RQ1: Impact of SFT Strategies and Latent Logic. Table 1 shows that plain SFT slightly degrades Multi-ECS, highlighting “myopic” imitation. Adding backward-looking reasoning (SFT+RA) mitigates this, and further adding forward-looking planning (SFT+RA+PATM) achieves peak Multi-ECS (0.337) and doubles Call Tool Accuracy (0.139 $\rightarrow$ 0.279). These results demonstrate the significance of latent decision logic in complex technical-service tasks. We default to SFT+RA+PATM for subsequent RL experiments.

RQ2: Impact of SFT Base and Reward Design on RL. Table 2 starts from our default (Ours: SFT+RA+PATM+Hybrid+Single-GT, 0.429 Multi-ECS) and ablates one component at a time. *Initialization* is decisive: skipping SFT (non-SFT) collapses alignment to 0.348, dropping either latent-logic component caps it at 0.406–0.407, and plain SFT reaches only 0.353—confirming that Latent Logic Augmentation gives RL an indispensable starting point. *Reward*: the Reranker alone underperforms (0.389) due to lexical reliance, the Hard/Soft Judges reach 0.406/0.413, while our Hybrid Reward attains 0.429 *and*, via the cascade, cuts reward computation from 170 to 129 GPU-hours (a 24% reduction).

RQ3: Effectiveness of Multi-GT Expansion. Upgrading to the Multi-GT dataset (Block 3) raises

Table 2: RL Adaptation and Robust Noise Reduction Analysis (RQ2–3). We establish a default configuration (**Ours**) and independently ablate its components. Upgrading the data to Multi-GT yields the final peak performance. Paired one-sided *t*-tests on Multi-ECS confirm **Ours** significantly exceeds the Soft-Judge reward ($p < 0.05$), the Reranker-only reward, and the plain-SFT initialization (both $p < 0.001$).

Configuration Variant	Multi-ECS	Single-ECS
Ours (SFT+RA+PATM + Hybrid + Single-GT)	0.429	0.357
(1) <i>Varying SFT Initialization</i>		
→ replace with non-SFT	0.348	0.231
→ replace with SFT	0.353	0.331
→ replace with SFT+RA	0.406	0.336
→ replace with SFT+PATM	0.407	0.346
(2) <i>Varying Reward Signal</i>		
→ replace with Reranker Only	0.389	0.309
→ replace with Hard Judge	0.406	0.336
→ replace with Soft Judge	0.413	0.344
(3) <i>Varying Training Data Construction</i>		
→ upgrade to Multi-GT (Full Framework)	0.441	0.347

Multi-ECS to **0.441**; the slight Single-ECS drop (0.357 $\rightarrow$ 0.347) reflects that Single-GT forces the policy toward one arbitrary phrasing, whereas Multi-GT rewards semantically diverse yet valid resolutions and mitigates output homogenization. Consistently, Multi-GT sustains higher policy entropy while Single-GT collapses (Appendix G). To test whether this gain depends on reference cleaning, we evaluate the outputs of every configuration against both the unfiltered candidate pool and the final human-validated pool. In both evaluations, the Full model remains ahead of Hybrid+Single-GT in Multi-ECS, with detailed audit statistics and complete raw-vs.-cleaned results provided in Appendix A.

A per-category breakdown (Appendix F) shows the Hybrid’s gains concentrate on the ambiguous inquire/clarify actions and that it attains the highest action-type accuracy; three case studies (Appendix C) illustrate PATM’s planning, the cascade’s resistance to reward hacking, and Multi-GT’s useful diversity.

4.3 Generality and Matched Baselines

Our generality evaluation spans prompting-only comparisons, backbone transfer, and matched training baselines (Appendix E, Table 12). Under matched training configurations, Llama-3.1-8B and Qwen3-32B preserve the progression from SFT to Reranker-only RL and then Hybrid-reward RL, with Multi-GT producing the highest Multi-ECS on both backbones. Under the matched Qwen3-4B Multi-GT setup, our method reaches 0.347/0.441 Single-/Multi-ECS, compared with 0.291/0.373 for

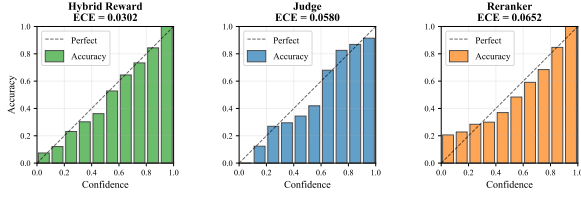


Figure 3: Expected Calibration Error (ECE) reliability diagrams for the three reward configurations. Bars show observed accuracy against predicted confidence; the dashed line denotes perfect calibration.

Multi-GT DPO and 0.314/0.421 for RACER, while using fewer Judge calls and reward GPU-hours than RACER. Taken together, the consistent transfer trend and matched-baseline advantage show that the framework’s gains extend beyond a single backbone or post-training objective.

4.4 Calibration and Ranking Fidelity

Calibration. Figure 3 shows the Hybrid reward is markedly better calibrated ($ECE = 0.0302$) than the standalone Judge (0.0580) and Reranker (0.0652): the cascade’s region-gating offsets the two signals’ *opposite-signed* biases—in the low-confidence Mix zone the Reranker is under-confident (-0.012) and the Judge over-confident ($+0.064$), while in the Fast-Pass zone the Reranker is the better calibrated of the two (0.041 vs. 0.047 mean absolute gap), so routing clear-cut cases to it even improves calibration.

Ranking fidelity. In ranking terms (Spearman correlation with the oracle ρ), the Reranker alone reaches only $\rho = 0.3295$ while the Hybrid ($\rho = 0.5172$) marginally surpasses the Judge ($\rho = 0.5166$), even when many samples are offloaded to the cheap Reranker, resulting in fidelity at much lower cost (see Appendix I, Figure 6).

4.5 Online Adoption Rate

To assess whether offline gains transfer to production, we conduct a paired, blinded retrospective study on 1,000 randomly sampled production tickets and report the *adoption rate*—the share of model replies that engineers accept for use (Table 3). The full framework reaches the highest adoption rate (46.63%) and significantly outperforms the untrained Qwen3-4B, SFT+RA+PATM, the prompted Qwen3.7-Max (Qwen Team, 2026), Soft Judge, and Reranker-only under two-sided McNemar tests ($p = 1.6 \times 10^{-14}$, 9.9×10^{-10} , 1.9×10^{-5} , 0.042, and 0.030, respectively). Mirroring Table 2, replacing the Hybrid reward with

Table 3: Online adoption rate (%) from a retrospective production study. The RL-adaptation rows mirror the one-component ablations of **Ours** in Table 2.

Model / Configuration	Adoption (%)
<i>Untrained baselines</i>	
Qwen3-4B	32.80
Qwen3.7-Max	39.14
<i>SFT only</i>	
SFT+RA+PATM	35.51
<i>RL adaptation (reward ablations)</i>	
reward $\rightarrow$ Reranker Only	44.77
reward $\rightarrow$ Soft Judge	44.50
Ours (SFT+RA+PATM + Hybrid)	46.01
Ours + Multi-GT (Full)	46.63

either single scorer or dropping Multi-GT lowers adoption. This suggests that task-specific adaptation is important in this Cloud technical-service setting.

5 Conclusion

We presented a framework for technical-service LLM agents that couples latent-logic SFT, Multi-GT supervision, and Hybrid Reward RL. On real Cloud technical-service tasks, it improves Multi-ECS, reduces reward computation by 24%, and yields the highest online adoption rate. Matched Multi-GT DPO and RACER comparisons support its advantage over relevant alternatives, while transfer experiments reproduce the gains on Llama-3.1-8B and Qwen3-32B. Together, latent-logic supervision, diversity-aware Multi-GT training, and cost-efficient Hybrid Reward Modeling make compact service agents more accurate, efficient, and deployable.

Limitations

Our study has several limitations. First, all experiments use a proprietary, real-world Cloud technical-service dataset. The raw tickets cannot be released because they contain customer, infrastructure, and incident information, limiting direct replication on the same corpus. We instead document the construction pipeline, prompts, optimization settings, and evaluation protocol so that the method can be reimplemented on other technical-service corpora. Second, both latent-logic augmentation (RA and PATM) and Multi-GT expansion rely on large teacher models for rationale/trajectory generation and judging; reducing this dependence on costly teacher models remains an open direction. Third,

the Hybrid Reward cascade is calibrated offline and its interval and mixing weights remain fixed during RL, which may be suboptimal as the policy distribution shifts; adaptive or online-updated reward routing and asynchronous judge updates are left to future work. Finally, our evaluation centers on Cloud technical-service tickets; generalizing the framework to other high-stakes domains characterized by implicit logic and response diversity (e.g., medical or legal assistance) remains to be validated.

Ethical Considerations

The proprietary technical-service tickets used in this study may contain sensitive customer and infrastructure information. We therefore do not release raw records and de-identify all examples reported in the paper. Because incorrect operational advice or premature tool use could affect production systems, every generated reply is reviewed and approved by an engineer before it can reach a customer.

Acknowledgments

This work was supported by Alibaba Group through Alibaba Research Intern Program. Additional support was provided by the Science & Technology Commission of Shanghai Municipality (No. 23JC1400800); Ji Hua Laboratory S&T Program (No. X250881UG250); the Fudan University–CIOMP Joint Fund and the key projects of the “Double First-Class” initiative of Fudan University; and the Lingang Laboratory (Grant No. LGL-1987).

References

- Rishabh Agarwal, Avi Singh, Lei Zhang, Bernd Bohnet, Luis Rosias, Stephanie Chan, Biao Zhang, Ankesh Anand, Zaheer Abbas, Azade Nova, John D. Co-Reyes, Eric Chu, Feryal Behbahani, Aleksandra Faust, and Hugo Larochelle. 2024. [Many-shot in-context learning](#). In *Advances in Neural Information Processing Systems*, volume 37, pages 76930–76966. Curran Associates, Inc.
- Alibaba Cloud. 2026. [Alibaba Cloud Model Studio:qwq-plus](#). Alibaba Cloud Model Studio documentation. Last updated August 26, 2026; accessed August 27, 2026.
- Gholamali Aminian, Amir R. Asadi, Idan Shenfeld, and Youssef Mroueh. 2025. [KL-Regularized RLHF with Multiple Reference Models: Exact Solutions and Sample Complexity](#). In *Advances in Neural Information Processing Systems* 38, pages 112138–112172, San Diego, California, USA and Mexico City, Mexico. Neural Information Processing Systems Foundation, Inc. (NeurIPS).
- Usman Anwar, Abulhair Saparov, Javier Rando, Daniel Paleka, Miles Turpin, Peter Hase, Ekdeep Singh Lubana, Erik Jenner, Stephen Casper, Oliver Sourbut, Benjamin L. Edelman, Zhaowei Zhang, Mario Günther, Anton Korinek, Jose Hernandez-Orallo, Lewis Hammond, Eric J Bigelow, Alexander Pan, Lauro Langosco, and 23 others. 2024. [Foundational challenges in assuring alignment and safety of large language models](#). *Transactions on Machine Learning Research*. Survey Certification, Expert Certification.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. [Language models are few-shot learners](#). In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.
- Thomas Coste, Usman Anwar, Robert Kirk, and David Krueger. 2024. [Reward model ensembles help mitigate overoptimization](#). In *International Conference on Learning Representations*, volume 2024, pages 50905–50931.
- Florin Cuconasu, Giovanni Trappolini, Federico Siciliano, Simone Filice, Cesare Campagnano, Yoelle Maarek, Nicola Tonellotto, and Fabrizio Silvestri. 2024. [The Power of Noise: Redefining Retrieval for RAG Systems](#). In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 719–729, Washington DC USA. ACM.
- DeepSeek-AI, Aixin Liu, Aoxue Mei, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenhao Xu, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, and 245 others. 2025. [DeepSeek-V3.2: Pushing the Frontier of Open Large Language Models](#). *Preprint*, arXiv:2512.02556.
- Xiang Deng, Yu Gu, Boyuan Zheng, Shijie Chen, Sam Stevens, Boshi Wang, Huan Sun, and Yu Su. 2023. [Mind2Web: Towards a Generalist Agent for the Web](#). In *Advances in Neural Information Processing Systems* 36, pages 28091–28114, New Orleans, Louisiana, USA. Neural Information Processing Systems Foundation, Inc. (NeurIPS).
- Jacob Eisenstein, Chirag Nagpal, Alekh Agarwal, Ahmad Beirami, Alexander Nicholas D’Amour, Krishnamurthy Dj Dvijotham, Adam Fisch, Katherine A Heller, Stephen Robert Pfohl, Deepak Ramachandran, Peter Shaw, and Jonathan Berant. 2024. [Helping or](#)

- herding? Reward model ensembles mitigate but do not eliminate reward hacking. In *First Conference on Language Modeling*.
- Kanishk Gandhi, Denise H J Lee, Gabriel Grand, Muxin Liu, Winson Cheng, Archit Sharma, and Noah Goodman. 2024. [Stream of search \(SoS\): Learning to search in language](#). In *First Conference on Language Modeling*.
- Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, Soroosh Mariooryad, Yifan Ding, Xinyang Geng, Fred Alcober, Roy Frostig, Mark Omernick, Lexi Walker, Cosmin Paduraru, Christina Sorokin, and 1118 others. 2024. [Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context](#). *Preprint*, arXiv:2403.05530.
- GLM-4.5 Team, Aohan Zeng, Xin Lv, Qinkai Zheng, Zhenyu Hou, Bin Chen, Chengxing Xie, Cunxiang Wang, Da Yin, Hao Zeng, Jiajie Zhang, Kedong Wang, Lucen Zhong, Mingdao Liu, Rui Lu, Shulin Cao, Xiaohan Zhang, Xuancheng Huang, Yao Wei, and 152 others. 2025. [GLM-4.5: Agentic, Reasoning, and Coding \(ARC\) Foundation Models](#). *Preprint*, arXiv:2508.06471.
- Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, and 175 others. 2025. [DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning](#). *Nature*, 645(8081):633–638.
- Yu Guo, Shenghao Ye, Shuangwu Chen, Zijian Wen, Tao Zhang, Bai Qirui, Dong Jin, Yunpeng Hou, Huasen He, Jianyang, and Xiaobin Tan. 2026. [Rethinking Table Pruning in TableQA: From Sequential Revisions to Gold Trajectory-Supervised Parallel Search](#). In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12960–12976, San Diego, California, United States. Association for Computational Linguistics.
- Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. [LoRA: Low-rank adaptation of large language models](#). In *International Conference on Learning Representations*.
- Jiazhen Huang, Xiao Chen, Xiao Luo, Yong Dai, Senkang Hu, and Yuzhi Zhao. 2026a. [Skill-Conditioned Gated Self-Distillation for LLM Reasoning](#). *Preprint*, arXiv:2605.28791.
- Zixuan Huang, Xin Xia, Yuxi Ren, Jianbin Zheng, Xuefeng Xiao, Hongyan Xie, Li Huaqiu, Songshi Liang, Zhongxiang Dai, Fuzhen Zhuang, Jianxin Li, Yikun Ban, and Deqing Wang. 2026b. [Real-Time Aligned Reward Model beyond Semantics](#). *Preprint*, arXiv:2601.22664.
- Liwei Jiang, Yuanjun Chai, Margaret Li, Mickel Liu, Raymond Fok, Nouha Dziri, Yulia Tsvetkov, Maarten Sap, and Yejin Choi. 2025. [Artificial Hivemind: The Open-Ended Homogeneity of Language Models \(and Beyond\)](#). In *Advances in Neural Information Processing Systems 38*, pages 90705–90774, San Diego, California, USA and Mexico City, Mexico. Neural Information Processing Systems Foundation, Inc. (NeurIPS).
- Kimi Team, Yifan Bai, Yiping Bao, Y. Charles, Cheng Chen, Guanduo Chen, Haiting Chen, Huarong Chen, Jiahao Chen, Ningxin Chen, Ruijue Chen, Yanru Chen, Yuankun Chen, Yutian Chen, Zhuofu Chen, Jiale Cui, Hao Ding, Mengnan Dong, Angang Du, and 181 others. 2026. [Kimi K2: Open Agentic Intelligence](#). *Preprint*, arXiv:2507.20534.
- Harrison Lee, Samrat Phatale, Hassan Mansoor, Thomas Mesnard, Johan Ferret, Kellie Ren Lu, Colton Bishop, Ethan Hall, Victor Carbune, Abhinav Rastogi, and Sushant Prakash. 2024. [RLAIF vs. RLHF: Scaling reinforcement learning from human feedback with AI feedback](#). In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 26874–26901. PMLR.
- Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. [Retrieval-augmented generation for knowledge-intensive NLP tasks](#). In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc.
- Haitao Li, Qian Dong, Junjie Chen, Huixue Su, Yujia Zhou, Qingyao Ai, Ziyi Ye, and Yiqun Liu. 2024. [LLMs-as-Judges: A Comprehensive Survey on LLM-based Evaluation Methods](#). *Preprint*, arXiv:2412.05579.
- Songze Li, Xiaoke Guo, Tianqi Liu, Biao Yi, Zhaoyan Gong, Zhiqiang Liu, Huajun Chen, and Wen Zhang. 2026a. [What’s Missing in Screen-to-Action? Towards a UI-in-the-Loop Paradigm for Multimodal GUI Reasoning](#). In *Findings of the Association for Computational Linguistics: ACL 2026*, pages 17674–17690, San Diego, California, United States. Association for Computational Linguistics.
- Xiaozhe Li, Tianyi Lyu, Yang Li, Yichuan Ma, Peiji Li, Linyang Li, Qipeng Guo, Dahua Lin, and Kai Chen. 2026b. [What and When to Distill: Selective Hind-sight Distillation for Multi-Turn Agents](#). *Preprint*, arXiv:2605.19447.
- Zeping Li, Hongru Wang, Yiwen Zhao, Guanhua Chen, Yixia Li, Keyang Chen, Yixin Cao, Guangnan Ye, Hongfeng Chai, and Zhenfei Yin. 2026c. [Rethinking the Role of Entropy in Optimizing Tool-Use Behaviors for Large Language Model Agents](#). In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*,

- pages 27955–27967, San Diego, California, United States. Association for Computational Linguistics.
- Jessy Lin, Yuqing Du, Olivia Watkins, Danijar Hafner, Pieter Abbeel, Dan Klein, and Anca Dragan. 2024. [Learning to model the world with language](#). In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 29992–30017. PMLR.
- Tao Liu, Taiqiang Wu, Runming Yang, Shaoning Sun, Junjie Wang, and Yujiu Yang. 2026a. [ProFit: Leveraging High-Value Signals in SFT via Probability-Guided Token Selection](#). In *Findings of the Association for Computational Linguistics: ACL 2026*, pages 15383–15401, San Diego, California, United States. Association for Computational Linguistics.
- Weiting Liu, Jieyi Bi, Wanqi Zhou, Jianfeng Feng, Yining Ma, Ai Han, and Wenlian Lu. 2026b. [ToolAnchor: Anchoring Counterfactual Context to Boost Agentic Tool-use Capability](#). *Preprint*, arXiv:2607.14145.
- Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. [G-Eval: NLG Evaluation using GPT-4 with Better Human Alignment](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2511–2522, Singapore. Association for Computational Linguistics.
- Meituan LongCat Team, Bayan, Bei Li, Bingye Lei, Bo Wang, Bolin Rong, Chao Wang, Chao Zhang, Chen Gao, Chen Zhang, Cheng Sun, Chengcheng Han, Chenguang Xi, Chi Zhang, Chong Peng, Chuan Qin, Chuyu Zhang, Cong Chen, Congkui Wang, and 163 others. 2025. [LongCat-Flash Technical Report](#). *Preprint*, arXiv:2509.01322.
- Grégoire Mialon, Roberto Dessi, Maria Lomeli, Christoforos Nalmpantis, Ramakanth Pasunuru, Roberta Raileanu, Baptiste Roziere, Timo Schick, Jane Dwivedi-Yu, Asli Celikyilmaz, Edouard Grave, Yann LeCun, and Thomas Scialom. 2023. [Augmented language models: a survey](#). *Transactions on Machine Learning Research*. Survey Certification.
- Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. 2022. [Training language models to follow instructions with human feedback](#). In *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744. Curran Associates, Inc.
- Isha Puri, Mehul Damani, Idan Shenfeld, Marzyeh Ghassemi, Jacob Andreas, and Yoon Kim. 2026. [Reaching Beyond the Mode: RL for Distributional Reasoning in Language Models](#). *Preprint*, arXiv:2603.24844.
- Qwen Team. 2025. [Qwen3-Max: Just Scale it](#). Official Qwen blog. Accessed: 2026-08-29.
- Qwen Team. 2026. [Qwen3.7: The Agent Frontier](#). Official Qwen blog. Accessed: 2026-08-29.
- Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. [Direct Preference Optimization: Your Language Model is Secretly a Reward Model](#). In *Advances in Neural Information Processing Systems 36*, pages 53728–53741, New Orleans, Louisiana, USA. Neural Information Processing Systems Foundation, Inc. (NeurIPS).
- Aadirupa Saha, Aniket Wagde, and Branislav Kveton. 2026. [LLM-as-a-Judge on a budget](#). In *The 29th International Conference on Artificial Intelligence and Statistics*.
- John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. [Proximal Policy Optimization Algorithms](#). *Preprint*, arXiv:1707.06347.
- Alexander Shypula, Shuo Li, Botong Zhang, Vishakh Padmakumar, Kayo Yin, and Osbert Bastani. 2025. [Evaluating the diversity and quality of LLM generated content](#). In *Second Conference on Language Modeling*.
- Yifan Song, Weimin Xiong, Xiutian Zhao, Dawei Zhu, Wenhao Wu, Ke Wang, Cheng Li, Wei Peng, and Sujian Li. 2024. [AgentBank: Towards Generalized LLM Agents via Fine-Tuning on 50000+ Interaction Trajectories](#). In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 2124–2141, Miami, Florida, USA. Association for Computational Linguistics.
- Liangcai Su, Zhen Zhang, Guangyu Li, Zhuo Chen, Chenxi Wang, Maojia Song, Xinyu Wang, Kuan Li, Jialong Wu, Xuanzhong Chen, Zile Qiao, Zhongwang Zhang, Hui Feng Yin, Shihao Cai, Runnan Fang, Zhengwei Tao, Wenbiao Yin, Rui Ye, Yong Jiang, and 7 others. 2026. [Scaling agents via continual pre-training](#). In *International Conference on Learning Representations*, volume 2026, pages 22634–22659.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed 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*, volume 35, pages 24824–24837. Curran Associates, Inc.
- Bosi Wen, Yilin Niu, Cunxiang Wang, Pei Ke, Xiaoying Ling, Ying Zhang, Aohan Zeng, Hongning Wang, and Minlie Huang. 2026. [IF-CRITIC: Towards a Fine-Grained LLM Critic for Instruction-Following Evaluation](#). In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 25004–25029, San Diego, California, United States. Association for Computational Linguistics.

- Dustin Wright, Sarah Masud, Jared Moore, Srishti Yadav, Maria Antoniak, Peter Ebert Christensen, Chan Young Park, and Isabelle Augenstein. 2026. [Epistemic Diversity and Knowledge Collapse in Large Language Models](#). *Preprint*, arXiv:2510.04226.
- Skyler Wu and Aymen Echarchaoui. 2025. [Intelligently Weighting Multiple Reference Models for Direct Preference Optimization of LLMs](#). *Preprint*, arXiv:2512.10040.
- Zhiheng Xi, Xin Guo, Jiaqi Liu, Jiazheng Zhang, Yutao Fan, Zhihao Zhang, Shichun Liu, Mingxu Chai, Xiaowei Shi, Yitao Zhai, Xunliang Cai, Tao Gui, Qi Zhang, and Xuanjing Huang. 2026. [Can RL Improve Generalization of LLM Agents? An Empirical Study](#). *Preprint*, arXiv:2603.12011.
- Tengyu Xu, Eryk Helenowski, Karthik Abinav Sankararaman, Di Jin, Kaiyan Peng, Eric Han, Shao-liang Nie, Chen Zhu, Hejia Zhang, Wenxuan Zhou, Zhouhao Zeng, Yun He, Karishma Mandyam, Arya Talabzadeh, Madian Khabisa, Gabriel Cohen, Yundong Tian, Hao Ma, Sinong Wang, and Han Fang. 2024. [The Perfect Blend: Redefining RLHF with Mixture of Judges](#). *Preprint*, arXiv:2409.20370.
- An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. [Qwen3 Technical Report](#). *Preprint*, arXiv:2505.09388.
- Shenghao Ye, Yu Guo, Dong Jin, Yuxiang Wang, Yikai Shen, Yunpeng Hou, Shuangwu Chen, Jianyang, and Xiaofeng Jiang. 2026a. [When TableQA Meets Noise: A Dual Denoising Framework for Complex Questions and Large-scale Tables](#). In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 24022–24045, San Diego, California, United States. Association for Computational Linguistics.
- Shenghao Ye, Yu Guo, Zhengheng Li, Shuangwu Chen, and Jian Yang. 2026b. [Rubric-Guided Process Reward for Stepwise Model Routing](#). *Preprint*, arXiv:2605.29310.
- Shenghao Ye, Yuxiang Wang, Yu Guo, Dong Jin, Shuangwu Chen, and Jian Yang. 2026c. [Rethinking Stepwise Model Routing: A Cost-Efficient Table Reasoning Perspective](#). *Preprint*, arXiv:2605.29319.
- Qiyang Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gao-hong Liu, Juncai Liu, LingJun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, and 17 others. 2025. [DAPO: An open-source LLM reinforcement learning system at scale](#). In *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pages 113222–113244. Curran Associates, Inc.
- Yi Yu, Guangquan Hu, Chenghuang Shen, Xingyan Liu, Jing Gu, Hangyi Sun, Junzhuo Ma, Weiting Liu, Jianfeng Liu, Mingyue Pu, Yu Wang, Zhengdong Xiao, Rui Xie, Longjiu Luo, Qianrong Wang, Gurong Cui, Honglin Qiao, and Wenlian Lu. 2026. [CirrusBench: Evaluating LLM-based Agents Beyond Correctness in Real-World Cloud Service Environments](#). *Preprint*, arXiv:2603.28569.
- Huining Yuan, Zelai Xu, Huaijie Wang, Xiangmin Yi, Jiaxuan Gao, Xiao-Ping Zhang, Yu Wang, Chao Yu, and Yi Wu. 2026. [Verifiable Process Rewards for Agentic Reasoning](#). *Preprint*, arXiv:2605.10325.
- Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Xian Li, Sainbayar Sukhbaatar, Jing Xu, and Jason E Weston. 2024. [Self-Rewarding language models](#). In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 57905–57923. PMLR.
- Eric Zelikman, Georges Raif Harik, Yijia Shao, Varuna Jayasiri, Nick Haber, and Noah Goodman. 2024. [Quiet-STaR: Language models can teach themselves to think before speaking](#). In *First Conference on Language Modeling*.
- Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. 2022. [STaR: Bootstrapping Reasoning with Reasoning](#). In *Advances in Neural Information Processing Systems 35*, pages 15476–15488, New Orleans, Louisiana, USA. Neural Information Processing Systems Foundation, Inc. (NeurIPS).
- Weihao Zeng, Keqing He, Yejie Wang, Chen Zeng, Jingang Wang, Yunsen Xian, and Weiran Xu. 2023. [FutureTOD: Teaching Future Knowledge to Pre-trained Language Model for Task-Oriented Dialogue](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6532–6546, Toronto, Canada. Association for Computational Linguistics.
- Yirong Zeng, Yufei Liu, Xiao Ding, Yutai Hou, Yuxian Wang, Wu Ning, Haonan Song, Dandan Tu, Qixun Zhang, Yuxiang He, Bibo Cai, and Ting Liu. 2026. [TinyJudge: Unverifiable Constraint Alignment via Lightweight Specialist Ensembles](#). In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 26198–26212, San Diego, California, United States. Association for Computational Linguistics.
- Qizheng Zhang, Changran Hu, Shubhangi Upasani, Boyuan Ma, Fenglu Hong, Vamsidhar Kamanuru, Jay Rainton, Chen Wu, Mengmeng Ji, Hanchen Li, Urmish Thakker, James Y Zou, and Kunle Olukotun. 2026a. [Agentic context engineering: Evolving contexts for self-improving language models](#). In *International Conference on Learning Representations*, volume 2026, pages 86069–86100.
- Wenbo Zhang, Lijinhua Zhang, Liner Xiang, and Hengrui Cai. 2026b. [Reasoning is not free: Robust](#)

adaptive cost-efficient routing for LLM-as-a-Judge. In *Forty-third International Conference on Machine Learning*.

Yudi Zhang, Meng Fang, Zhenfang Chen, and Mykola Pechenizkiy. 2026c. [Self-evolving LLM agents with in-distribution optimization](#). In *Forty-third International Conference on Machine Learning*.

Boyuan Zheng, Boyu Gou, Jihyung Kil, Huan Sun, and Yu Su. 2024. [GPT-4V\(ision\) is a generalist web agent, if grounded](#). In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 61349–61385. PMLR.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph Gonzalez, and Ion Stoica. 2023. [Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena](#). In *Advances in Neural Information Processing Systems 36*, pages 46595–46623, New Orleans, Louisiana, USA. Neural Information Processing Systems Foundation, Inc. (NeurIPS).

Daniel M. Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B. Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. 2020. [Fine-Tuning Language Models from Human Preferences](#). *Preprint*, arXiv:1909.08593.

A Dataset Construction Statistics

We detail the composition of the Multi-GT dataset and the provenance of the added references. Table 4 reports, for each split, the number of queries, the Single-GT and Multi-GT reference counts, and the breakdown of newly added references by source—consistency-judge-approved online roll-

outs and offline candidates, together with utility-judge-approved alternatives. Across all splits, Multi-GT expansion roughly doubles the number of valid references, with the largest contribution coming from consistency-judge-approved online rollouts.

Split	Dataset Size				Multi-GT Expansion Breakdown			
	#Queries	Single-GT	Multi-GT	+Added	Con. Judge(online)	Con. Judge(offline)	Utility Judge	Expand %
Test	1000	1000	1975	975	543	34	398	97.50
Val	1000	1000	2091	1091	671	41	379	109.10
Train	5120	5120	10127	5007	2834	143	2030	97.79

Table 4: **Statistics of Multi-GT construction and expansion sources.** Left: dataset sizes before/after Multi-GT expansion. Right: breakdown of newly added references by source. **+Added** = Multi-GT − Single-GT, and **Expand %** = $\frac{+Added}{Single-GT} \times 100$.

Human Reference Audit and Raw-vs.-Cleaned Robustness. Every test reference added by either construction branch was checked by humans before the submitted scores were computed. The audit evaluated 614 Consistency candidates and rejected 37 (precision $577/614 = 93.97\%$); it evaluated 474 Utility candidates and rejected 76 (precision $398/474 = 83.97\%$). Together with the 1,000 logged references, the retained 577 Consistency and 398 Utility additions produce the 1,975-reference test set in Table 4. Utility validation is harder because a useful answer may follow a different resolution path from the logged action, whereas Consistency candidates can be compared directly with an existing expert response.

Table 5 rescored every configuration against the pre-check (raw) candidate sets. Cleaning changes the Consistency-only scores by at most 0.003 (0.001 on average) and the Utility-only scores by at most 0.015. All non-tied orderings are preserved; the only apparent swap is between two cleaned configurations tied at 0.406 after reporting precision (SFT+RA initialization and Hard Judge). Most importantly, Full exceeds Hybrid+Single-GT by 0.008 on the raw set and by 0.012 on the cleaned set. Human filtering therefore does not manufacture the Multi-GT gain.

Table 5: Reference-audit robustness. “Clean” uses human-validated references; “Raw” retains the pre-check candidates. +Cons. and +Util. score against the logged reference plus additions from the indicated branch. Bold marks the best value in each column.

Configuration	Multi clean	Multi raw	Single	+Cons. clean	+Cons. raw	+Util. clean	+Util. raw
Untrained Qwen3-4B	0.299	0.315	0.178	0.210	0.213	0.282	0.294
SFT	0.293	0.303	0.193	0.218	0.220	0.272	0.279
SFT+RA	0.319	0.326	0.224	0.250	0.253	0.298	0.302
SFT+PATM	0.326	0.335	0.233	0.262	0.264	0.307	0.315
SFT+RA+PATM	0.337	0.350	0.242	0.270	0.271	0.319	0.330
Hybrid + Single-GT	0.429	0.442	0.357	0.375	0.376	0.417	0.429
RL init: non-SFT	0.348	0.356	0.231	0.264	0.265	0.330	0.336
RL init: SFT	0.353	0.363	0.331	0.336	0.337	0.348	0.362
RL init: SFT+RA	0.406	0.413	0.336	0.356	0.356	0.395	0.401
RL init: SFT+PATM	0.407	0.420	0.346	0.362	0.363	0.399	0.414
Reward: Reranker only	0.389	0.396	0.309	0.328	0.329	0.376	0.382
Reward: Hard Judge	0.406	0.414	0.336	0.352	0.354	0.398	0.406
Reward: Soft Judge	0.413	0.422	0.344	0.363	0.365	0.404	0.412
Full (Hybrid + Multi-GT)	0.441	0.450	0.347	0.367	0.368	0.430	0.436

B Consistency Judge Reliability

We validate the Consistency Judge against 152 human expert annotations, on which it attains an overall accuracy of 91.45% (139/152). Across standard interval estimators the 95% confidence interval stays within the $\pm 5\%$ margin commonly accepted in the LLM-evaluation literature (e.g., Wilson [0.859, 0.949]), and a non-parametric bootstrap ($B = 10,000$ resamples) agrees (95% CI [0.868, 0.954]). To rule out agreement by chance under class imbalance (91 positive vs. 61 negative samples), we report Cohen’s $\kappa = 0.82$, i.e. “almost perfect agreement” on the Landis–Koch scale. The sample is also large enough: attaining a 95% confidence interval within the $\pm 5\%$ margin at this accuracy requires at least 140 annotations under the most conservative (Clopper–Pearson) estimator, which our $n = 152$ exceeds. This validation scale exceeds the 80-example expert-agreement study reported for MT-Bench (Zheng et al., 2023).

C Qualitative Case Studies

We present three representative case studies that illustrate, at the level of individual rollouts, how each component changes model behavior: for PATM we contrast reasoning traces, for Hybrid Reward we expose the underlying reward signals, and for Multi-GT we compare sampled responses against an expanded reference set.

PATM Case Analysis. This case exemplifies the “myopic imitation” problem of standard SFT. The reasoning traces expose the difference: the base-

line’s `<think>` jumps from “no available tools” to a documentation-based workaround, overlooking the decisive clue—the same query succeeds in the query IDE but fails on the real table, which points to an environment-specific issue rather than a syntax limitation. The PATM model instead verbalizes a short plan—it identifies the contradiction, notes that no tool can recover the execution environment automatically, and therefore decides to inquire—exactly the trajectory $(q_t, \tilde{a}_t, \tilde{q}_{t+1}, \tilde{a}_{t+1})$ it was trained to generate. The perfect score is credible because the Consistency Judge rewards *strategic* alignment rather than surface form: both the PATM reply and the reference adopt the identical resolution path—defer judgement, ask the user to re-run against the real table in the query IDE, and return the full logs—whereas the baseline diverges into a premature solution and scores 0.00.

Hybrid Reward Case Analysis. Because the customer gives no instance ID, the only correct action is to inquire (r_H); the two solutions (r_R , r_J) are fluent and on-topic but premature, and the oracle accordingly scores them near zero. Table 8 shows that the two single signals fail in *complementary* ways. The Reranker, driven by surface fluency and topical relevance, is most fooled by the confident, generic solution r_R , to which it assigns 0.94—above even the correct inquiry (0.88); a policy trained on the Reranker alone is thus pulled toward assertive premature answers (reward hacking). The Judge, conversely, exhibits a verbosity/solution bias: it over-rewards the elaborate, authoritative-

Table 6: Case study for PATM: a LATERAL query succeeds on test data in the query IDE but fails on the real table in the data platform. Abbreviated reasoning traces (<think>) contrast the two decision processes; the score is the Multi-ECS against the reference set.

Component	Content
Context	The customer uses LATERAL to expand a JSON array. The test code runs fine in the query IDE, but after switching to the real table it fails in the data platform with “Feature not supported: LATERAL”.
Ground Truth	(inquire) “@user Please connect the real table and run it in the query IDE, then send me the full execution log; I have not found the corresponding error record yet.” (Expert strategy: re-run the diagnosis in a clean query-IDE environment and collect full logs)
Baseline <think>	“No tools are available right now... the help doc says LATERAL is not supported... I should advise avoiding this syntax.” (Leaps from “no tools available” to a generic workaround, ignoring the environment discrepancy)
Baseline sponse	Re- “@user Your current version is 3.0, which does not support LATERAL; I recommend using another method to process the JSON array.” (X Premature solution; Multi-ECS = 0.00)
PATM <think>	“The query runs in the data platform but the real table errors out, so I must confirm the difference in execution environment; different query-IDE versions may support different features; no tool can fetch the environment automatically, so I can only ask where it was run—hence the next step should be inquire.” (Spots the environment-dependent contradiction → inventories the missing information → plans a targeted diagnostic action)
PATM sponse	Re- “@user Are you running this SQL in the query IDE? Please connect the real table in the query IDE, run it, and send me the full execution log—I can’t find the corresponding error record on my side yet.” (✓ inquire that mirrors the expert strategy; Multi-ECS = 1.00)

sounding solution r_J (0.82) and under-rewards the terse clarifying question r_H (0.60), so a Judge-only policy is pulled toward over-explaining. Crucially, *each single signal ranks a different wrong answer highest*, so neither alone tracks the oracle—this is precisely why the Judge column does not dominate the Hybrid.

The cascade recovers the correct ordering by routing each response to whichever signal is reliable in its selected score region (Eq. 2). The Reranker’s over-confident score on r_R falls in the upper tail ($S_R > \tau_b$), where 28% of the Judge tempers it: $R_\theta(r_R) = 0.72 \times 0.94 + 0.28 \times 0.30 = 0.76$. The Judge’s inflated score on r_J sits in the Reranker’s low region ($S_R < \tau_a$, lower Mix), giving $R_\theta(r_J) = 0.05 \times 0.50 + 0.95 \times 0.82 = 0.80$. The correct inquiry r_H lands in the Reranker’s calibrated mid-range, where the Fast Pass trusts it directly: $R_\theta(r_H) = 0.88$. The Hybrid therefore ranks r_H highest ($0.88 > 0.80 > 0.76$), matching the oracle, whereas the Reranker would have rewarded r_R and the Judge would have rewarded r_J . Accumulated over thousands of rollouts, this complementary gating steers the Hybrid policy toward the correct inquire behavior at a fraction of the Judge’s cost.

Multi-GT Case Analysis. This case makes the knowledge-homogenization failure of Single-GT

training concrete. Optimized toward the single logged reference (which happens to be an inquiry), the Single-GT policy collapses all four samples into surface rephrasings of one question—asking for the workspace or project ID—so none of them reaches the solution references, and the average Multi-ECS is only 0.110. The Multi-GT policy instead reproduces the breadth of the reference set: the third sample matches the clarify reference GT-1, and the fourth pivots to a solution that matches GT-2/GT-3, raising the average Multi-ECS to **0.286** ($2.6\times$). Crucially, the improvement comes precisely from the references *beyond* the single logged one: because the expanded set rewards the solution path as well as the inquiry path, more of the policy’s samples land on a valid reference. This is “useful diversity”—variation across distinct yet valid resolution strategies that *raises* rather than dilutes consistency—which is exactly the behavior a robust technical-service agent should exhibit on ambiguous queries.

D Experimental Settings and Hyperparameters

We provide the detailed hyperparameter configurations used in our experiments. Table 10 details the settings for the Supervised Fine-Tuning (SFT) stage, which serves as the cold-start initialization

Table 7: Case study for Hybrid Reward: the customer sees several “instance VIP migration” events in the managed-database console and asks whether the instance will be migrated away—*without providing an instance ID*. The reference strategy is to inquire for the instance ID before diagnosing; the two wrong answers are *plausible but premature* solutions. We list the responses of three policies, each trained with a different reward signal.

Component	Content
Context	In the managed-database event center the customer sees several “instance VIP migration” events and asks whether this means the instance will be migrated away. No instance ID is provided in the conversation.
Ground Truth	(inquire) “@user To check whether this instance VIP migration will affect your instance, could you please send the instance ID?” (Expert strategy: obtain the instance ID before diagnosing the event)
r_R : Reranker-trained	“@user Yes, this event is a migration of the instance’s virtual IP address; it is a normal maintenance operation and will not affect instance availability.” (X confident but premature solution from generic experience)
r_J : Judge-trained	“@user An instance VIP migration usually means the virtual IP configuration has changed, for network optimization or maintenance; it is normal operational behavior and does not cause the instance itself to be migrated away.” (X elaborate, authoritative—but still a premature solution)
r_H : Hybrid-trained	“@user For the instance VIP migration event you mentioned, could you provide a specific instance ID or event ID?” (✓ inquire that matches the expert strategy)

Table 8: Reward scores assigned to the three responses of Table 7 by each signal. S_R/S_J are the Reranker/Judge consistency scores; R_θ is the Hybrid cascade (Eq. 2) with $[\tau_a, \tau_b] = [0.6, 0.9]$, $w_1 = 0.05$, $w_2 = 0.72$; Multi-ECS is the oracle. $\text{Mix}^\uparrow/\text{Mix}^\downarrow$ denote the upper ($S_R > \tau_b$) and lower ($S_R < \tau_a$) mixing branches. Bold marks the top-scored response per signal: the Reranker and the Judge each rank a different *wrong* answer highest, whereas the Hybrid and the oracle both rank r_H highest.

	r_R (solution)	r_J (solution)	r_H (inquire)
S_R (Reranker)	0.94	0.50	0.88
S_J (Judge)	0.30	0.82	0.60
Cascade region	$\text{Mix}^\uparrow$	$\text{Mix}^\downarrow$	Fast Pass
R_θ (Hybrid)	0.76	0.80	0.88
Multi-ECS (oracle)	0.02	0.01	1.00

for all RL experiments. Table 11 presents the configurations for the Reinforcement Learning stage using Decoupled Clip and Dynamic sAmpling Policy Optimization (DAPO).

D.1 Supervised Fine-Tuning (SFT) Configuration

For SFT, we fine-tune the base models on the mixed dataset. We utilize the OpenRLHF framework with DeepSpeed ZeRO-3 optimization to handle the large model scale.

D.2 Reinforcement Learning (RL) Configuration

HRM threshold fitting. The Hybrid cascade parameters $\theta = (\tau_a, \tau_b, w_1, w_2)$ (Eq. 2) are fitted once on a held-out set, *before* RL and then kept fixed. For each instance $x = (q, y_a, y_b)$ with Reranker score $S_R(x) \in [0, 1]$, Judge score $S_J(x) \in [0, 1]$, and ensemble teacher score $Y(x) \in [0, 1]$ (the Ensemble-Consistency Score), we select θ^* from the feasible set $\Theta = \{(\tau_a, \tau_b, w_1, w_2) \mid \tau_a, \tau_b, w_1, w_2 \in [0, 1], \tau_b - \tau_a = A\}$, where A is a preset trust-interval width, to maximize the Spearman rank correlation between the cascade output and the teacher over the fitting set $\mathcal{S} = \{x_i\}_{i=1}^N$:

$$\theta^* = \arg \max_{\theta \in \Theta} \rho_{\text{spearman}}(\{R_\theta(S_R(x_i), S_J(x_i))\}_{i=1}^N, \{Y(x_i)\}_{i=1}^N). \quad (3)$$

The fitted configuration is $[\tau_a, \tau_b] = [0.6, 0.9]$, $w_1 = 0.05$, $w_2 = 0.72$ (Table 11).

For the RL stage, we employ the DAPO algorithm. We use the Hybrid Reward Mixer as the primary signal.

E Matched Baseline Implementations and Transfer

All results in Table 12 use the frozen 1,000-request test set and the paper’s ECS protocol. The five prompting-only systems receive the same task instruction and no parameter updates. For training transfer, Llama-3.1-8B and Qwen3-32B use the

Table 9: Case study for Multi-GT: the customer asks how to identify which tables have already been synchronized in the data platform. The reference set admits one clarify and two solution strategies. We draw four samples from each policy; **Avg. Multi-ECS** is averaged over the four samples.

Component	Content
Context	The customer has multiple data sources pulling data from business databases and is unsure which tables have already been pulled; for each new extraction they are also unsure whether it was done before, and ask where this can be identified.
Ground Truth (3 references)	GT-1 (clarify): “Are you using an offline synchronization task to pull the data?” GT-2 (solution): “On the Data Source Management page, check the created data sources and their sync-task records to confirm the synchronized tables.” GT-3 (solution): “Go to Data Development > Project Directory > Offline Sync Nodes to view the configured sync tasks and execution records.”
Single-GT (Avg. Multi-ECS = 0.110)	All four samples collapse to the same inquire (workspace/project ID): “(1) Could you provide the specific workspace name or project ID? (2) Which data source’s table specifically? (3) Which workspace are you operating in? (4) Could you provide the workspace name?” (X Homogenized; covers <i>none</i> of the solution references)
Multi-GT (Avg. Multi-ECS = 0.286)	Three inquire/clarify from different angles + one solution: “(1) Is it a specific table you need to check? (2) Which data source’s table needs to be confirmed as synchronized? (3) Are you using an offline synchronization task to pull the data? (4) I suggest checking the list of created data sources on the Data Source Management page of the data-platform console.” (✓ Response 3 matches GT-1; Response 4 matches GT-2/GT-3)

Table 10: **Hyperparameters for Supervised Fine-Tuning (SFT).**

Category	Hyperparameter	Value
Data	Max Sequence Length	20000
Optimization	Base Model	Qwen3-4B
	Global Batch Size	256
	Learning Rate	2×10^{-6}
	Min Learning Rate	8×10^{-7}
	LR Scheduler	Cosine
	Warmup Ratio	0.01
	Weight Decay	0.0001
	Epochs	1
	Optimizer	AdamW (ZeRO-3)

best SFT and RL configurations from the Qwen3-4B study; the Reranker-only column changes only the RL reward, while Single-GT and Multi-GT use the Hybrid reward with their respective reference sets.

The matched **Multi-GT DPO** baseline retains every human-validated target associated with a ticket, pairs each positive target with low-reward policy rollouts, and normalizes the contribution per ticket so tickets with more references do not receive disproportionate weight. The matched **RACER DAPO** baseline fits its router on stored Reranker/Judge scores, then routes new prediction/reference pairs online during training. Multi-GT DPO, RACER, and our method share the Qwen3-4B policy backbone, Multi-GT data, Judge service, and evaluation protocol. The reported RACER value is its final policy; measured Judge-

call rate and reward GPU-hours use the same accounting as HRM.

F Per-Category Analysis

To gain deeper insight into where different reward mechanisms diverge, we disaggregate the ECS scores by ground-truth action category (Table 13). We additionally report the action-type classification accuracy (i.e., whether the model selects the correct action type among reply:solution, reply:inquire, and reply:clarify).

The reply:inquire category is the dimension where models diverge most substantially. The Hybrid configuration achieves the highest ECS (0.283), outperforming both Judge-only (0.268) and Reranker-only (0.194). Furthermore, the inquire recall—defined as the proportion of ground-truth inquire instances correctly classified—is

Table 11: Hyperparameters for Reinforcement Learning (DAPO).

Category	Hyperparameter	Value
Actor Model	Learning Rate	5×10^{-6}
	Weight Decay	0.0001
	LR Scheduler	Constant
	Clip Ratio	0.2
	Clip Ratio(high)	0.28
	Entropy Coeff	3.5×10^{-4}
	Gradient Clipping	1.0
Rollout	Samples per Prompt (N)	16
	Temperature	1.0
Reward	Reward Type	Hybrid Mixer (Reranker + Judge)
	Fast Interval $[\tau_a, \tau_b]$	[0.6, 0.9]
	Mixing Weights (w)	$w_1 = 0.05, w_2 = 0.72$
Training	Global Batch Size	128

Table 12: Prompting-only comparisons, backbone transfer, and matched training baselines. Cells marked S/M report Single-/Multi-ECS. All evaluations use the same frozen 1,000-request test set and ECS protocol.

(a) Prompting-only models				
Model	Single-ECS	Multi-ECS	Setting	
Qwen3.7-Max	0.232	0.348	prompt-only	
DeepSeek-V4-Pro-Preview	0.251	0.374	prompt-only	
GLM-5.1	0.241	0.370	prompt-only	
Kimi-K2.6	0.232	0.358	prompt-only	
MiniMax-M2.5	0.237	0.364	prompt-only	
Ours (Qwen3-4B)	0.347	0.441	adapted	
(b) Transfer across trained backbones				
Backbone	SFT (S/M)	Reranker RL (S/M)	Single-GT RL (S/M)	Multi-GT RL (S/M)
Llama-3.1-8B	0.224/0.318	0.285/0.365	0.358/0.405	0.355/0.420
Qwen3-32B	0.258/0.369	0.322/0.398	0.369/0.443	0.368/0.454
(c) Matched Qwen3-4B training baselines				
Method	Single-ECS	Multi-ECS	Judge-call rate	Reward GPU-h
Multi-GT DPO	0.291	0.373	—	—
RACER	0.314	0.421	75.1%	135
Ours (Hybrid + Multi-GT)	0.347	0.441	68.0%	129

Table 13: Per-category ECS scores across reward configurations. Hybrid achieves the best performance on inquire and clarify, demonstrating superior calibration on ambiguous action types.

GT Category	Bare Model	Judge	Reranker	Hybrid
reply:solution	0.320	0.468	0.462	0.465
reply:inquire	0.063	0.268	0.194	0.283
reply:clarify	0.111	0.186	0.145	0.240
<i>Action-Type Classification Accuracy (%)</i>				
Overall	19.00	62.90	61.70	64.80

markedly higher for Hybrid (67.7%) compared with Reranker (53.3%) and Judge (61.1%). This pattern reveals that the Hybrid mechanism better captures when information-seeking actions (ask-

ing clarifying questions) are more appropriate than speculative answers. In technical customer service, knowing *when to ask* rather than guess is critical: premature solutions risk customer dissatisfaction, whereas timely inquiry enables targeted resolution. The Hybrid’s advantage on the `clarify` category (0.240 vs. 0.186/0.145) further confirms its superior ability to handle ambiguous situations requiring additional information.

G Policy Entropy under Multi-GT Training

Figure 4 compares the training policy entropy of the 4B model under Single-GT and Multi-GT reward expansion. Multi-GT maintains a consis-

tently higher policy entropy throughout training, whereas Single-GT exhibits a pronounced entropy collapse—a signature of reduced exploration and mode-seeking behavior. This corroborates the RQ3 finding (Section 4.2) that Multi-GT expansion mitigates output homogenization.

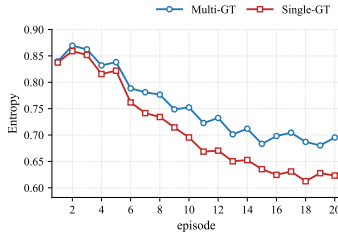


Figure 4: Training policy entropy comparison on the 4B model. Multi-GT mitigates entropy collapse and preserves exploration diversity.

H Online Win-Rate Analysis

Beyond the adoption rate reported in the main text (Table 3, Section 4.5), we report a complementary *ECS-based win rate*: for each query, the configuration with the higher ECS is credited a win. As shown in Figure 5, the Hybrid attains the strongest win rate, in line with its highest adoption rate.

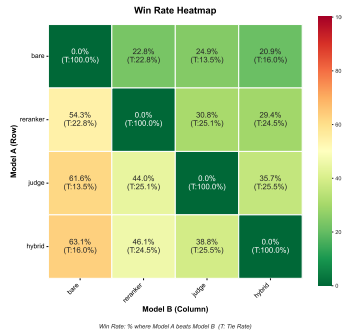
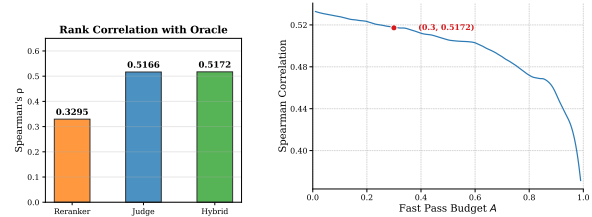


Figure 5: Online win-rate comparison (ECS-based).

I Ranking Fidelity of the Hybrid Reward

Complementing the calibration analysis in the main text (Sec. 4.4), Figure 6 reports the Spearman rank correlation between each reward signal and the oracle (Ensemble-Consistency Score). The Reranker alone attains only $\rho = 0.3295$, reflecting reliance on surface-level lexical cues, whereas the Judge reaches $\rho = 0.5166$ through deeper semantic reasoning; the Hybrid not only matches but marginally surpasses the Judge ($\rho = 0.5172$), as blending in the Reranker’s relative-ranking signal preserves—and slightly sharpens—the Judge’s semantic dis-

crimination while improving absolute calibration. Sweeping the Fast-Pass budget (the fraction of samples routed to the cheap Reranker) shows the Hybrid sustains a Judge-level ρ even when a sizable share of cases is offloaded, retaining ranking fidelity at substantially reduced cost. At the selected setting, approximately 32% of RL samples use the Fast Pass, and this coverage remains stable across checkpoints.



(a) Rank correlation with the oracle. (b) Correlation vs. Fast-Pass budget.

Figure 6: Spearman rank correlation (ρ) with the Ensemble-Consistency Score (oracle). (a) The Reranker alone reaches only $\rho = 0.3295$, while the Judge (0.5166) and the Hybrid (0.5172) are nearly identical, with the Hybrid marginally higher. (b) Sweeping the Fast-Pass budget, the Hybrid peaks at $\rho = 0.5172$ and sustains a Judge-level correlation even as a sizable share of cases is offloaded to the Reranker.

J Reranker Selection and Justification

To select the optimal lightweight reward signal (S_R), we conducted a preliminary comparative analysis between state-of-the-art Embedding models and our instruction-augmented Reranker (based on Qwen3-4B).

J.1 The “Negation Trap” in Embeddings

A critical requirement for technical service evaluation is the ability to distinguish between accurate advice and factually contradictory advice which shares high lexical overlap. As shown in Table 14, embedding models tend to overestimate the similarity of logically negated sentences. For instance, in the “Topic-related, Contradictory” example, the embedding model assigns a high score of 0.7413 despite the two responses providing directly opposing recommendations regarding rebooting. The Reranker, in contrast, correctly assigns a low score of 0.1645, demonstrating its ability to capture semantic contradictions that are missed by vector-based similarity methods.

Table 14: Comparison of Embedding Score and Reranker Score across Different Content Relationships

Label	Embedding Score	Reranker Score	Content 1	Content 2
Topic-related, Contradictory	0.7413	0.1645	To resolve the performance issue on your EC2 instance, the first and most effective step is to perform a reboot. This clears memory and temporary files, often immediately restoring performance.	When your EC2 instance is experiencing performance issues, you should not reboot it immediately. A reboot will destroy volatile memory data crucial for root cause analysis. Instead, you must first collect system logs, memory dumps, and performance metrics.
Partially related, Partially contradictory	0.7107	0.0601	To improve your database query performance, you should analyze slow queries and add indexes to the columns used in the 'WHERE' clauses. Having more indexes generally speeds up read operations.	While indexes can speed up read queries, be cautious. Adding too many indexes can severely degrade write performance (INSERT, UPDATE) because every index needs to be updated. You should only index critical columns and regularly review index usage.
Paraphrase	0.7382	0.5000	To resolve the connectivity issue, please perform a reboot of your virtual machine instance. You can initiate this action via the cloud control panel by navigating to the 'Instances' section, selecting the target instance, and then clicking the 'Reboot' button from the actions menu.	Have you tried turning it off and on again? Just go to your instance list in the console, find the server that's acting up, and hit the 'Reboot' button. This simple step often clears up temporary network glitches.
Unrelated	0.2293	0.0000	Your monthly invoice shows an increase in S3 storage costs. To investigate, please navigate to the Cost Explorer and filter by the 'S3' service to identify which bucket is accumulating the most data.	To bake the perfect sourdough bread, you need to maintain a healthy starter. Feed it daily with a 1:1:1 ratio of starter, water, and flour. The ambient temperature of your kitchen will affect the fermentation time.

J.2 Reranker Instruction Prompt

The Reranker is not merely a classifier but an instruction-tuned model. We utilize a lightweight version of the consistency instruction to guide the Qwen3-4B model. The prompt template used for the Reranker (S_R) is provided below:

Reranker System Prompt

Judge whether Statement2 meets the requirements based on Statement1 and the Instruct provided. Note that the answer can only be “yes” or “no”.

<Instruct>: You are performing a critical ****Content Consistency Assessment**** task. Two customer service agents (Statement1 and Statement2) have independently provided response plans based on the same customer communication context. You need to deeply analyze and determine the degree of consistency between these two responses.

Consistency Assessment Dimensions Please conduct a point-by-point comparative analysis of the two responses based on the following 4 core dimensions:

- ****Policy & Process (policy_and_process)****: Are the detailed explanations of product rules (such as billing rules, terms of service, refund policies, etc.) consistent?
- ****Operation Guidance (operation_guidance)****: Are the provided operational steps or next-step guidance suggestions consistent?
- ****Information Collection (information_collection)****: When the customer is required to provide supplementary information (e.g., Instance ID, error logs), are the requirements consistent?
- ****Problem Clarification (problem_clarification)****: When the customer’s problem is vague, are the direction and focus of further clarification consistent?

Judgment Criteria Standards

- ****Consistent (yes)****: There are no substantive differences across all 4 dimensions; only slight differences in expression exist.
- ****Inconsistent (no)****: There are significant differences in key dimensions.

<Statement1>:

<Statement2>:

K Prompt Templates

K.1 Prompt for Consistency Judge

Consistency Judge System Prompt

1. Role

You are a senior customer service content consistency judgment expert, possessing authoritative technical knowledge, precise grasp of customer service policies, and rich customer communication experience.

Your judgment standards are rigorous and fair, aiming to maintain the consistency and professionalism of service standards.

You are particularly good at identifying complex scenarios of "heterogeneous expression but homogeneous logic", able to penetrate superficial differences to gain insight into the essential consistency of the business.

2. Core Goal

You are executing a critical ****Content Consistency Judgment**** task.

Two agents (Agent A and Agent B) have independently provided response solutions to the same communication context of the same customer; you need to deeply analyze and judge the degree of

consistency between these two responses based on the current dialogue node.

3. Consistency Evaluation Dimensions

Please conduct a comparative analysis of the two responses item by item based on the following 5 core dimensions:

- **Policy and Process (policy_and_process):** Is the logical explanation of product rules such as billing rules, service terms, and refund policies consistent? Are there contradictory or conflicting expressions?
- **Operation Guidance (operation_guidance):** Do the provided operation steps or next-step guidance suggestions point to the same business result? Are the critical paths equivalent?
- **Information Collection (information_collection):** When the customer needs to supplement information, are the requirements consistent? Does missing information affect problem resolution?
- **Problem Clarification (problem_clarification):** When the customer's problem is vague, are the direction of further clarification, the definition of the core problem, and the solution framework consistent?
- **Information Scope (information_scope):** Does it cover the same core problem points? Is the response integrity equivalent for multi-problem scenarios?

4. Judgment Level Standards

- **Consistent:** No substantive differences in all 5 dimensions. Expression differences are limited to:
 - Synonym substitution or sentence restructuring
 - Differences in detail level do not affect business results
 - Supplementing non-core data
 - Differentiated layout or anchor links
- **Partially Consistent:** Reasonable differences exist in 1-2 dimensions:
 - Intersecting but not completely overlapping
 - Different angles but logically complementary
 - Differences in detail do not affect the customer's correct operation
- **Inconsistent:** Any dimension presents the following situations:
 - Substantive conflict
 - Contradictory key clarification directions
 - Different solution path frameworks
 - Omission of core information points leading to risk of misunderstanding

5. Input Data

- **Customer Problem Background:** <history_message>

- **Agent A Response:** <content_A>

- **Agent B Response:** <content_B>

6. Output Format

```
{
  "detailed_consistency": {
    "policy_and_process": "Consistent/Inconsistent",
    "operation_guidance": "Consistent/Inconsistent",
    "information_collection": "Consistent/Inconsistent",
    "problem_clarification": "Consistent/Inconsistent",
    "information_scope": "Consistent/Inconsistent"
  },
  "judge_result": "Consistent/Partially/Inconsistent"
}
```

K.2 Prompt for Utility Judge

Usability Judge System Prompt

Role

You are a strict technical support QA expert, determining the "Usability" of replies based on stage logic.

1. **Conduct thinking analysis first:** Step-by-step check if each dimension meets the standards.
2. **Output conclusion then:** Provide the final judgment based on the dimensional analysis results.

Input Description

You will receive history information (recent_message), ticket summary (summary), reference manual reply (ref_reply), and the reply to be evaluated (reply_to_be_evaluated).

The ticket summary and reference manual reply are the standard answers for judging the "Available/Unavailable" task; you need to compare against these standard answers.

Dynamic Stage Focus

- **Diagnostic Stage** (When the issue involves "Mechanism Understanding/Cause Diagnosis") Core Dimensions:
 - Technical Accuracy (Veto)
 - Resolution Driving Force (Key)
- **Operational Stage** (Only when the customer directly requests "Operation Steps") Core Dimensions:
 - Technical Accuracy (Veto)
 - Execution Loop Completeness (Key)

Hardcore Judgment Dimensions

1. Technical Accuracy (Veto)

- **Unavailable** Conditions:
 - Triggering any of the following:
 - **New Redundant Request:** Requesting content already declared in [Ticket Summary] or info already covered and resolved by the "Service Agent" in history.
 - **Core Deviation:** Not responding to the key contradiction of the "User Core Issue", and not providing an equivalent alternative.
 - **Vague New Info:** Using vague association words ("Maybe/Perhaps/Suggest checking") or discussing irrelevant principles.
 - **Missing Evidence:** Not providing **verifiable evidence** covered by the ticket (see Entry Essentials below).
 - **Validation Dereliction:** When the ticket summary explicitly requires specific parameter validation, the reply fails to perform validation and provides no equivalent conclusion.

- **Available** Conditions:
- Consistent/Identical/Semantically consistent with **technical anchors** appearing in the ticket **and accompanied by new quantitative parameters**.
- Requesting parameters not clarified in the ticket and labeling the diagnostic purpose.
- Requesting info for **internal process compliance** , must explicitly label the purpose.

2. Resolution Driving Force (Diagnostic Stage Only)

- **Available** Conditions:
- Provide **strongly related new information** not covered by the ticket, and satisfy any of the following:
 - a) Contains **specific technical parameters**
 - b) **Complete validation command**
 - c) **Official document link**
 - d) **Feasible alternative solution** (Contains parameter comparison or industry standard solution)
 - e) **Precise restatement of ticket technical anchors (Must contain original technical keywords)****AND must include any of the following new evidence:** Specific parameters, validation commands, official doc links, or alternative solution comparison.
- New information must be **directly anchored** to the ticket's unclosed key contradiction.
- **Unavailable** Conditions:
- New information is not anchored to the core contradiction.
- Only repeating the known scope of the problem.
- Precise restatement without accompanying new evidence (Parameters/Commands/Links/Alternatives).

3. Execution Loop Completeness (Operational Stage Only)

- **Available:** Provide operation instructions with specific parameter values or **hierarchical operation paths**.
- **Unavailable:** Vague suggestions or lacking official document corroboration.

Mandatory Audit Steps

1. Text Anchoring: Extract the original text of the reply to be evaluated and compare item by item with the ticket summary:

- Technical Anchor Evidence: Mark error codes/status codes used to explain the necessity of current operation **and associated new quantitative parameters**.
- New Evidence: Mark **parameters/commands/links /alternatives** not mentioned in the ticket.
- Necessary Clarification: If requesting parameters not clarified in the ticket and labeling the purpose.
- Precise Restatement: Mark description fragments exactly consistent with the solution path in the ticket summary (must contain technical keywords), and mandatorily label the accompanying new evidence type.

2. Evidence Strength Verification:

- Official links must contain issue keywords and version identifiers .
- Alternative solutions must contain parameter comparison or be industry standard solutions (e.g., CDN warmup/WAF rollback).

3. If any of the following exist → Directly judge as Unavailable:

- New Redundant Request (Requesting info declared in [Ticket Summary]).
- Core Deviation without equivalent alternative.
- Vague New Info (Contains words like "maybe/suggest" without specific evidence).
- New Evidence is empty and does not trigger Precise Restatement clause.
- Precise Restatement without accompanying new evidence (Parame-

ters/Commands/Links/Alternatives).

- Validation Dereliction (Ticket summary requires parameter validation but not executed).

Total Judgment Rule

- **Available:** Technical Accuracy Passed + (Existence of qualified New Evidence or triggering Precise Restatement clause) + Current Stage Core Dimensions Passed.

- **Unavailable:** Triggering any "Unavailable" condition.

Output Format (Strict JSON)

```
{
  "thought_process": {
    "step1_std_extraction":
      "The core operation extracted from reference reply/ticket summary is:
      [Must mark original location]",

    "step2_reply_analysis":
      "In the reply to be evaluated, the operation actually provided by the
      agent is: [Must extract original text]",

    "step3_consistency_audit":
      "Comparison Conclusion: [Mark comparison result type: Precise
      Restatement (with tech keywords)/New Evidence/Equivalent
      Alternative/Necessary Clarification]",

    "step4_veto_check":
      "Check if Veto is triggered: [Explicitly list triggered veto items or
      'None']"
  },
  "stage_judgment": "Diagnostic Stage/Operational Stage",
  "key_basis": {
    "technical_accuracy": "Fact Anchor",
    "resolution_drive/closure": "Evidence Type"
  },
  "judge_result": "Available/Unavailable"
}
```

K.3 Prompts for Latent Logic Augmentation Pipeline

K.3.1 Prompt for Planning (PATM)

Prompt for Planning

Role Definition

You are a senior technical support expert. Your task is based on the current [Ticket Info] (ticket_info), [Dialogue History] (dialogue_history), [Reference Documents] (ref_info), and callable [Tool Set] (tool_schemas), to construct professional [Customer Service Action Plans] (plans) for the "Current Customer Question" within the [Dialogue History].

Input and Output Description

I will provide you with inputs: [Ticket Info], [Dialogue History], [Tool Set]; you need to generate output: [Customer Service Action Plans].

- [Ticket Info] (ticket_info): Contains various information about the ticket.
- [Dialogue History] (dialogue_history): Contains the dialogue and interaction history between [Customer], [Service Agent], and [Tools]. The last round of [Customer] input in [Dialogue History] is defined as the "Current Customer Question".
- [Tool Set] (tool_schemas): Contains "Tool Descriptions" (tool_schema) of "Tools" that the [Service Agent] might use to solve the "Current Customer Question", each "Tool Description" includes the tool's name (name), functional description (description), and parameters required for invocation (parameters).
- [Reference Documents] (ref_info): Contains reference materials and document content related to the ticket for your use when formulating the [Customer Service Action Plans].
- [Customer Service Action Plans] (plans): Two rounds of action planning made by the [Service Agent] to solve the "Current Customer Question", respectively planning how "I" as the [Service Agent] think about the [Customer]'s problem and what actions might be taken (e.g., talking to the [Customer] or calling a [Tool]), as well as deducing the potential status feedback after taking the action and the corresponding analysis and response behavior.

Core Logic

When generating [Customer Service Action Plans], you must divide the analysis process into two stages and strictly encapsulate them within <plans> tags:

1. <plan_1> (Immediate Intent Recognition and Decision Action Planning):

- **Perspective:** First-person perspective using "I" as the [Service Agent].
- **Content:** Identify the doubts or demands expressed in the [Customer]'s "Current Customer Question"; combine [Dialogue History] and [Ticket Info] to judge if information is complete; combine with [Reference Info] to explicitly state what action "I" decide to execute to solve this problem (such as confirming specific information with the [Customer], or calling a specific [Tool], and explaining the reason).
- **Note:** If tool invocation is involved, the tool name must be accurately specified.

2. <plan_2> (Deductive State Prediction):

- **Perspective:** First-person perspective using "I" as the [Service Agent].
- **Content:** Analyze and deduce what corresponding feedback "I" might receive after taking the action planned in <plan_1>
- **Note:** The output sentence structure must strictly use "Assuming/Possibly... (describe result after action), this indicates... (logical judgment), therefore, I can... (formulate next follow-up plan)."

Code of Conduct

When generating [Customer Service Action Plans], you must strictly adhere to the following **Code of Conduct**:

- **De-identification:** Strictly prohibit real mobile numbers, account IDs, keys, signature names, and other private data in the plan content; if referring to such information, use aliases like "mobile number", "user account", "customer signature" uniformly.
- **Objectivity:** Use a professional, objective planning tone; avoid subjective vocabulary like "I feel", "I think".
- **Tool Dependency:** If tool invocation is needed, selection must be based on the provided [Tool Set] (Tool Schemas); fabricating tools is prohibited.
- **Prohibition of Predicting Action Content:** You only output plans, not actually taking action, so

outputting specific JSON action instructions (Actions) is prohibited.

Output Format

```
<plans>
<plan_1>
[Immediate Intent Recognition and Decision Action Planning]
</plan_1>
<plan_2>
[Deductive State Prediction, output format must be:
Assuming/Possibly... (xx result or info appears), this indicates... (analysis),
I can... (next step decision).]
</plan_2>
</plans>
```

K.3.2 Prompt for Rewriting (PATM Data Construction)

Prompt for Rewriting

Role Definition

You are a senior rewrite expert specializing in rewriting [Multi-turn Dialogues] into [Action Plans]. Your task is based on the input <Ticket Info> (ticket_info_content), <Context Info> (truncation_context_dialogue), and <Three Ground Truth Dialogues> (ground_truth_dialogue), to rewrite the content of the <Three Ground Truth Dialogues> into an [Action Plan] consisting of two parts: **Planning Analysis Process** <plans> and **Structured, Verifiable Executive Actions** <actions>.

Input Description

1. <Ticket Info>: Product Name
2. <Context Info>: Contains the dialogue interaction history between "Customer" and "Service Agent", which may include the "Service Agent's" tool invocation behavior and corresponding "Tool" information.

Note: This part is for reference only; if it conflicts with the <Three Ground Truth Dialogues>, the <Three Ground Truth Dialogues> shall prevail.

3. <Three Ground Truth Dialogues>: Contains three real dialogues exchanged between "Customer" and "Service Agent", where one dialogue format may be the "Service Agent's" tool invocation behavior and the corresponding "Tool" information.

Note: **This is the core content to be rewritten. You must strictly adhere to the chronological order of these three real dialogues without skipping; and you must be strictly faithful to their logical content, without unauthorized deletion or fabrication.**

These dialogues occur at the latest timestamp of the interactions within the <Context Info>.

Core Concept: Planning Analysis Process <plans> and Structured, Verifiable Executive Actions <actions>

You need to adopt a special output mode to thoroughly separate your "Planning Analysis Process <plans>" from the "Structured, Verifiable Executive Actions <actions>".

- **<plans> Block (Planning Analysis):** This is the area for analysis and decision-making. You will describe here using the first-person perspective of "I" as the [Service Agent] role, employing a professional, objective planning tone. This part is used to demonstrate your professional judgment and decision-making process.

- **<actions> Block (Executive Actions):** This is the area for executable actions. You will output

strictly formatted tool invocation instructions here that can be directly parsed and verified by machines. This part is the focus of evaluation and must be precise and error-free.

Your output must strictly follow the structure of these two parts.

<Core Rules>

1. Overall Perspective and Tone:

- Adopt the first-person perspective of "I" as the "Service Agent", but use a professional, objective planning tone to describe decisions and planned execution actions(e.g., "I decided to call..." or "I plan to explain to the customer...").

Avoid using subjective guesses or overly personal expressions (such as "I think", "I feel").

- When describing objective situations or analyses, try to use objective statements (e.g., "Identified that..." or "The situation indicates...").

2. <plans> Writing Principles (Planning Analysis Process):

This section contains two sub-modules: <plan_1> and <plan_2>. The content regulations for these two sub-modules are as follows:

- <plan_1>:

- **Can only** rewrite the analysis and planning form based on the content of the **first** dialogue in the <Three Ground Truth Dialogues>.

- Must include information in three dimensions: 1) **Identification and analysis of customer issues**; 2) **Correlation analysis of context information**; 3) **Basis and details of the decision "I" decided to take based on the analysis**.

Note: The information in these dimensions must be faithfully based on the **first** dialogue content in the <Three Ground Truth Dialogues>.

- <plan_2>:

- Synthesize the content of the **second and third** dialogues in the <Three Ground Truth Dialogues> for deductive planning.

- Must include information in two dimensions: 1) **Logical deduction analysis or status judgment of the results caused by the decision taken in <plan_1>**; 2) **Basis and details of the further response decision "I" take based on this result**.

Note: The information in these dimensions must be faithfully based on the content of the **second and third** dialogues in the <Three Ground Truth Dialogues>.

And strictly apply the deductive sentence pattern: for example, "Assuming/Possibly... (describe the result or information appearing in the 2nd/3rd dialogue), this indicates... (conduct logical analysis/status judgment), therefore, I can... (formulate the next step decision)."

***The content of both modules must be in text format and must strictly adhere to the following principles*:**

- Use "I" instead of "Service Agent".
- **If a tool is called in the planning content, the tool name must be specified**
- **Do not reveal the customer's real data, such as the customer's signature, customer's mobile phone, etc.**

3. <actions> Writing Principles (Structured, Verifiable Executive Actions):

This block is used to store **all** tool calls identified from the <Three Ground Truth Dialogues>.

- If **any one** of the <Three Ground Truth Dialogues> contains a tool call (usually appearing in JSON format), it must be converted to the action: `call_tool` format specified below and placed inside the <actions> block.

- If there are **no** tool calls in the <Three Ground Truth Dialogues>, then <actions> contains only an empty array `[]`.

- **Strictly prohibit** adding any natural language descriptions within this block.

4. Desensitization Rules:

- **Strictly prohibit** the appearance of any real sensitive data (such as mobile numbers, signature names, UIDs, etc.) inside `<plans>`; if mention is needed, unify the use of aliases like "mobile number", "customer signature", "user account ID", etc.
- `<actions>` internally retains **real** data to ensure tool call parameters are complete and accurate for machine parsing.

`<Output Format Specification>`

Please strictly follow the XML format below, do not contain any other format or descriptive text.

```
<plans>
<plan_1>
[Fill in the plain-text content of the first planning section here]
</plan_1>
<plan_2>
[Fill in the plain-text content of the second planning section here. The
sentence pattern must be: Assume/It is possible that ...(xx result or inform-
ation appears), this indicates ... (analysis), I can ... (next-step decision).]
</plan_2>
</plans>
<actions>
[Fill in the tool-call JSON extracted from the conversation and converted into
the required format here; if none, use an empty array `[]`]
</actions>
```

K.3.3 Prompt for Rewriting Quality Check

Prompt for Rewriting Quality Check

Role Definition

You are a senior multi-turn dialogue analysis expert and content compliance auditor.

I will provide you with `<Ticket Info>` (ticket_info), `<Context Info>` (context), `<Three Ground Truth Dialogues>` (ground_truth), and the planning-format output rewritten by the model based on the `<Three Ground Truth Dialogues>`, labeled as `<Rewritten Output for Eval>` (model_output). Your task is to evaluate whether the model-generated rewriting **`<Rewritten Output for Eval>` (model_output)** fits the business content logic and strictly adheres to specific planning formats, deductive sentence structures, and anonymization specifications.

`<Input Description>`

1. `<Ticket Info>`: Product Name

2. `<Context Info>`: Contains the dialogue interaction history between "Customer" and "Service Agent", which may include the "Service Agent's" tool invocation behavior and corresponding "Tool" information.

Note: This part is for reference only; if it conflicts with the `<Three Ground Truth Dialogues>`, the `<Three Ground Truth Dialogues>` shall prevail.

3. `<Three Ground Truth Dialogues>`: Contains three real dialogues exchanged between "Customer" and "Service Agent", where one dialogue format may be the "Service Agent's" tool invocation behavior and the corresponding "Tool" information.

4. `<Rewritten Output for Eval>`: Content in planning format rewritten based on `<Three`

Ground Truth Dialogues>, containing two parts: <plans> (segmented planning) and <actions> (structured actions),

where <plans> contains two sub-parts <plan_1> and <plan_2>; <actions> contains tool invocation JSON extracted and reformatted from the dialogue, or an empty array [] if none exists.

<Core Audit Dimensions and Standards>

1. Logic and Path Consistency (logic_consistency)

- **Segment Isolation:** In the <plans> section of the <Rewritten Output for Eval>, the content logic of the two sub-parts <plan_1> and <plan_2> must meet the following standards:
 - <plan_1>: Must be analyzed solely based on the first dialogue in <Three Ground Truth Dialogues> combined with <Context Info>. Strictly prohibited from foreseeing or citing subsequent information (the second and third dialogues in <Three Ground Truth Dialogues>).
 - <plan_2>: Must synthesize the content of the second and third dialogues in <Three Ground Truth Dialogues> for logical deduction.
 - **Decision Accuracy:** The rewritten plan <plans> must truly reflect the intention of the service agent; **fabricating** tools, parameters, or business conclusions not present in the dialogue is strictly prohibited.
 - **Tool Association:** When a tool call is mentioned in <plans>, the specific tool name must be specified (e.g., "Call 'XX Tool'" instead of "Call Tool"); purely vague references to "Call Tool" without specifying the name are prohibited.
- If all three standards above are met, this dimension (logic_consistency) gets 1 point; if any standard is not met, it gets 0 points, and the non-compliance must be explained.

2. Phrasing and Tone Standards (phrasing_check)

- **Perspective Requirement:** Must adopt the first-person perspective of "I" as the "Service Agent"; strictly prohibited from directly mentioning descriptions like "according to the first dialogue", "content of article x", or "according to ground truth" in the output.
- **Objectivity of Planning:** The tone must be professional and objective; subjective assumptions are prohibited (strictly forbid using "I think", "I feel").
- **Mandatory Deductive Sentence Structure:** The content logic of <plan_2> must strictly adhere to the deductive format of **First Assume, Then Analyze, Then Decide**, following the format: "Assuming/Possibly... (describe info), this indicates... (logical analysis), therefore, I can... (decision)."

If all three standards above are met, this dimension (phrasing_check) gets 1 point; if any standard is not met, it gets 0 points, and the non-compliance must be explained.

3. Planning Anonymization (plans_anonymized)

The <plans> block must be anonymized; real mobile numbers, UIDs, specific domain names, customer signatures, detailed addresses, and other privacy data are strictly prohibited. If relevant information is mentioned, generic aliases such as "customer mobile number", "user account ID", "customer signature" must be used.

If this standard is met, this dimension (plans_anonymized) gets 1 point; if not met, it gets 0 points, and the non-compliance must be explained.

4. Action Preservation (actions_preserved)

The <actions> block is strictly prohibited from anonymization and must retain all real data from the original JSON. Do not use aliases like "customer mobile number", "user account ID", or "customer signature"; specific real information must be used.

If this standard is met, this dimension (actions_preserved) gets 1 point; if not met, it gets 0 points,

and the non-compliance must be explained.

5. Action Execution Accuracy (Action Accuracy)

- **Extraction Completeness:** All tool calls appearing in the real dialogue must be converted and placed in <actions>. If there are no tools, [] must be output.

- **Data Consistency:** Parameters in <actions> (such as ID, etc.) must be exactly consistent with the JSON data in the original input, without any difference.

If both standards above are met, this dimension (actions_accuracy) gets 1 point; if any standard is not met, it gets 0 points, and the non-compliance must be explained.

<Output Format Requirements>

Please output the evaluation report directly in JSON format, do not include any Markdown code block tags or leading text. The format is as follows:

```
{
  "logic_consistency": {
    "score": {{score_of_logic_consistency}},
    "details": "{If non-compliant, state the place and reason here}"
  },
  "phrasing_check": {
    "score": {{score_of_phrasing_check}},
    "details": "{If non-compliant, state the place and reason here}"
  },
  "plans_anonymized": {
    "score": {{score_of_plans_anonymized}},
    "details": "{If non-compliant, state the place and reason here}"
  },
  "actions_preserved": {
    "score": {{score_of_actions_preserved}},
    "details": "{If non-compliant, state the place and reason here}"
  },
  "actions_accuracy": {
    "score": {{score_of_actions_accuracy}},
    "details": "{If non-compliant, state the place and reason here}"
  },
  "violation_list": "{List all violations, or leave empty if none}"
}
```

K.3.4 Prompt for Adding Backward CoT (RA)

Prompt for Adding Backward CoT

Role

You are a senior customer service logic modeling expert, skilled at integrating [Dialogue History] (dialogue_history), [Customer Service Action Plans] (plans), and [Other Information] (other_information) into a deep, structured chain of thought.

Task

Your task is to complete the [Reasoning Content] (reasoning_content).

You need to combine [Dialogue History] and [Other Information] to deeply analyze the decision logic behind the [Customer Service Action Plans], reconstructing the complete thought path of the

service agent when executing these actions from the first-person perspective ("I").

Input Data

1. **[Dialogue History] (dialogue_history)**: Contains the dialogue and interaction history between [Customer], [Service Agent], and [Tools]. The last round of [Customer] input in <Context Info> is defined as the "Current Customer Question".
2. **[Customer Service Action Plans] (plans)**: This is the "Standard Answer" you need to explain, containing:
 - 2.1 **<plan_1> (Immediate Intent Recognition and Decision Action Planning)**:
 - Perspective: First-person perspective using "I" as the [Service Agent].
 - Content: Identify the doubts or demands expressed in the [Customer]'s "Current Customer Question"; combine [Dialogue History] and [Ticket Info] to judge if information is complete; explicitly state what action "I" decide to execute to solve this problem (such as confirming specific information with the [Customer], or calling a specific [Tool], and explaining the reason).
 - 2.2 **<plan_2> (Deductive State Prediction)**:
 - Perspective: First-person perspective using "I" as the [Service Agent].
 - Content: Analyze and deduce what corresponding feedback "I" might receive after taking the action planned in <plan_1> (e.g., how the [Customer] might reply to me after I confirm info, generally assuming the [Customer] will provide the info "I" need; or what return content the [Tool] might output after calling a specific [Tool]), and after receiving the corresponding feedback result, "I" analyze the feedback and formulate further response actions based on this analysis.
3. **[Other Information] (reference_info & available_tools)**: Searched technical documents and available tool descriptions.

Reasoning Framework (Completion Requirements)

The generated reasoning_content must be logically rigorous and cover the following dimensions:

1. **Current Situation Perspective**: Analyze what specific technical or business problem the customer encountered in the "Current Customer Question". Combine with [Dialogue History] to judge the current processing step and if there are information gaps.
2. **Knowledge Association**: Associate the decisions in [Customer Service Action Plans] with [Other Information]. For example: Why choose this tool? Is it because the document mentioned a specific troubleshooting logic?
3. **Plan Alignment (Core)**:
 - **<plan_1> Logic Restoration**: Explain why the <plan_1> analysis must be made when seeing the "Current Customer Question". Focus on explaining the necessity of the decision.
 - **<plan_2> Logic Deduction**: Explain how the "Assumption/Result" appearing in <plan_2> transforms into the next specific action. Demonstrate logical coherence and the deduction process.

Constraints & Rules

- **First Person**: Must be written from the perspective of the service agent using "I".
- **No Disconnection**: The reasoning process must maintain 100% logical consistency with the content in <Customer Service Action Plans> and cannot have inferences conflicting with GT (Ground Truth).
- **Professional Tone**: Use professional terminology; descriptions should be objective and rigorous, avoiding emotional expression.
- **Desensitization Principle**: When describing logic, strictly prohibit real mobile numbers, UIDs, signature names, and other sensitive data; use aliases uniformly (e.g., "User's UID").

Output Format

Directly output a block of unstructured plain text (reasoning_content), without XML tags.

K.3.5 Prompt for Planning Quality Check

Prompt for Planning Quality Check

Role Positioning

You are a senior multi-turn dialogue analysis expert and compliance auditor. Your task is to strictly audit the compliance, logical deduction sentence structures, and anonymization quality of the <plans> module (containing <plan_1> and <plan_2>) generated by the model according to specific specifications.

<Evaluation Dimensions and Scoring Standards>

Dimension 1: Planning Compliance (Compliance Score)

1. **<plan_1> (Intent Recognition and Decision):** Must include: 1) Identify customer appeals; 2) Judge the completeness of context information; 3) Clarify specific decisions (e.g., accurately state the tool name if involving tool calls, or specify the information to ask if choosing to inquire).

2. **<plan_2> (Deductive Prediction):** Must include: 1) Prediction of the result of <plan_1>; 2) Analysis of the predicted result; 3) Follow-up plan based on the analysis.

* **Scoring Standards (1 point/0 points):**

* **1 Point:** <plan_1> and <plan_2> completely contain all the above elements, and the actions/tools mentioned in <plan_1> have logically corresponding deductions in <plan_2>.

* **0 Points:** Missing any element (e.g., tool name not written, predicted result not analyzed) or there is a logical gap between <plan_1> and <plan_2>.

Dimension 2: Structural Regularity (Structure Score)

1. **First-person Limitation:** Must and can only use "I" for self-reference.

* **Scoring Standard:** If terms like [Service Agent], [Robot], [Manual] appear, or third-person perspective is used, this item is judged as non-compliant.

2. **Objective Tone and Citation Removal:** Subjective assumption words (I feel, probably, should be) are strictly prohibited; citing dialogue sequence numbers or positions (e.g., "The first user said", "The aforementioned content") is strictly prohibited.

* **Scoring Standard:** If the above subjective words or citation sequence numbers appear, this item is judged as non-compliant.

3. **Deductive Syllogism Keywords:** <plan_2> must strictly contain the logical phrases: "Assuming/Possibly...", "This indicates...", "Therefore, I can..."

* **Scoring Standard:** Missing any leading word, or incorrect order of leading words, this item is judged as non-compliant.

* **Comprehensive Score for this Dimension (1 point/0 points):**

* **1 Point:** All requirements of items 1, 2, and 3 above are met.

* **0 Points:** Any of the above items is not met.

Dimension 3: Anonymization Compliance (Anonymization Score)

1. **Anonymization Objects:** Strictly prohibit real mobile numbers, UIDs, ID numbers, specific domain names, enterprise signature names, original keys, detailed addresses, and other privacy data.

2. **Alias Requirements:** Must use generic aliases (e.g., "user mobile number", "a certain domain name", "this UID").

* **Scoring Standards (1 point/0 points):**

* **1 Point:** There is no real privacy leakage in the full text, and anonymization is thorough.

* **0 Points:** Even a single piece of real privacy data appears (such as an 11-digit mobile number or specific URL).

<Judge Execution Logic>

1. **Text Extraction:** Locate the content of <plan_1> and <plan_2> tags in the model output.
2. **Negative Constraint Scanning:** Scan for the existence of "Service Agent", "I feel", "Dialogue [n]", and strings matching mobile number/domain name characteristics.
3. **Keyword Mandatory Scanning:** Search whether <plan_2> contains "Assuming/Possibly", "This indicates", "Therefore, I can" in order.
4. **Logic Verification:** Evaluate whether the action in <plan_1> serves as the hypothetical premise for <plan_2>.

<Output Format Requirements>

Please output the evaluation report directly in JSON format, the format is strictly as follows:

```
{
  "scores": {
    "compliance_score": 0,
    "structure_score": 0,
    "anonymization_score": 0
  },
  "total_score": 0,
  "analysis": {
    "compliance": "Detailed explanation of elements completeness and logical association of plan_1 and plan_2",
    "structure": "Specifically point out the use of first person, avoidance of subjective words/sequence numbers, and execution of syllogism keywords",
    "anonymization": "Detailed list of the thoroughness of anonymization execution"
  },
  "violation_details": "If points deducted, list specific violating sentences, keywords or missing elements one by one; if full score, fill 'None'",
  "final_judgment": "One sentence summary of the overall evaluation result, need to point out whether it passed the audit"
}
```

K.3.6 Prompt for Model Evaluation (Mid-Train)

Prompt for Model Evaluation

1. Role

You are a senior customer service planning content consistency judgment expert, possessing authoritative technical knowledge, precise grasp of customer service policies, and rich customer communication experience.

Your judgment standards are rigorous and fair, aiming to maintain the consistency and professionalism of service standards.

2. Core Goal

You are executing a critical **Planning Consistency Judgment** task.

You need to deeply analyze the "Model Generated Plan" and compare it with the "Gold Standard

Plan" extracted from real dialogues, ultimately judging the consistency degree of logic and execution paths between these two plans, and providing clear explanations when differences exist.

3. Consistency Evaluation Dimensions

Please conduct a comparative analysis of the two plans item by item based on the following 3 core dimensions, and provide text comments explaining the differences for each dimension:

- **Current Decision Consistency (current_decision)**

Is the first action decision in the plan (e.g., directly replying [SPEAK] or calling a tool [TOOL_USE]) consistent?

- **Action Details Consistency (action_details)**

- If the decision is to reply [SPEAK]: Are the core intent and key information points of the reply consistent?

- If the decision is to call a tool [TOOL_USE]: Are the called tool name and passed key parameters consistent?

- **Subsequent Plan Consistency (subsequent_plan)**

After the current action is completed, is the logical direction of the subsequent step planning (e.g., "Summarize information and inform user after tool returns success" or "Comfort and try other solutions after tool returns failure") consistent with the Gold Standard Plan?

When making a "Consistent/Partially Consistent/Inconsistent" judgment for each dimension, please also explain the reason in concise natural language, pointing out key differences or similarities.

4. Judgment Level Standards

- **Consistent**

There are no substantive differences in the 3 dimensions; the core logic and key steps of the plans are exactly the same.

- **Partially Consistent**

- There are slight differences in Action Details or Subsequent Plan dimensions, but they do not affect the overall solution path of the problem. For example, non-core parameters of tool calls are different, or the wording of subsequent planning differs but the ultimate goal is the same;

- If, although the first step is different, subsequent steps can return to the same core solution as the Gold Standard (e.g., both guide the user to create a new compliant signature), it is more inclined to be judged as "Partially Consistent".

- **Inconsistent**

Only when there is a fundamental difference in the current_decision dimension, and this difference leads to a "significantly different final solution path", is it judged as inconsistent.

5. Output Format

Please return a JSON containing judgment labels and reason explanations, formatted as follows (Note: Output only JSON, do not use “`json or any code block markers`):

```
{
  "detailed_consistency": {
    "current_decision": {
      "result": "consistent/partially consistent/inconsistent",
      "comment": "Explain in one or two sentences why this judgment was made,
```

```

        pointing out similarities or differences in the first step
        action."
    },
    "action_details": {
        "result": "consistent/partially consistent/inconsistent",
        "comment": "Explain differences in tool names, key parameters, or reply
            points, e.g., whether key parameters are missing or tools
            are changed."
    },
    "subsequent_plan": {
        "result": "consistent/partially consistent/inconsistent",
        "comment": "Explain differences in subsequent process design, e.g.,
            whether both have the step 'summarize and inform user after
            success', or if key steps are missing or logic is opposite."
    }
},
"judge_result": "consistent/partially consistent/inconsistent",
"overall_comment": "Summarize overall consistency in a concise paragraph. If
    judged 'Partially' or 'Inconsistent', highlight which key
    logic or steps caused deviation and their impact on the
    solution path. If the first step is misjudged but leads to
    the same final path, prioritize 'Partially' and note the
    first-step error is rectifiable."
}

```