SEAD: Self-Evolving Agent for Multi-Turn Service Dialogue

Anonymous ACL submission

Abstract

Large Language Models have demonstrated remarkable capabilities in open-domain dialogues. However, their performance in service dialogues remains suboptimal, as these require agents to guide users toward specific business objectives while dynamically tracking states and adapting strategies. This gap stems from the scarcity of high-quality training data and the difficulty in simulating authentic, goal-oriented user behaviors. We propose **SEAD** (**S**elf-**E**volving **A**gent for Service **D**ialogue), a framework that enables agents to learn effective strategies without large-scale human annotations. SEAD decouples user modeling into two components: a Profile Controller that generates diverse user states to manage training curriculum, and a User Role-play Model that focuses on realistic role-playing. This design ensures the environment provides adaptive training scenarios rather than acting as an unfair adversary. Experiments demonstrate that SEAD significantly outperforms Open-source Foundation Models and Closed-source Commercial Models, improving task completion rate by 17.6% and dialogue efficiency by 11.1%. All code and resources will be open-sourced.

1 Introduction

Large Language Models (LLMs) have revolutionized dialogue systems across customer service, healthcare, and education (Kar et al., 2023). However, training robust goal-oriented dialogue agents—LLMs designed to guide users toward completing specific tasks—remains bottlenecked by scarce, expensive, and low-quality conversational data (Hosseini-Asl et al., 2020). In multi-turn service dialogue, agents must dynamically track user states and adapt strategies across extended interactions. Human logs from outbound call services (where agents proactively contact potential customers to promote products or services) exemplify these challenges (Qian et al., 2022): agents exhibit inconsistent behaviors requiring costly screening; cleaned data exhibits severe distribution skew; data quality is fundamentally capped by the capabilities of the original human agents who produced these conversations.

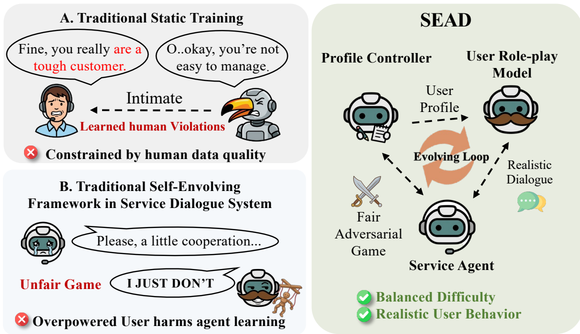

Figure 1: Comparison of training paradigms. (a) Static data methods are constrained by data quality ceiling. (b) Traditional interactive simulators cannot perceive agent capability and lack realistic behaviors. (c) SEAD achieves balanced co-evolution through decomposed user modeling and adaptive difficulty control.

To solve data quality issues, prior works have explored synthetic data and user simulation as alternatives. Static synthesis approaches (Li et al., 2025; Ou et al., 2024; Wang et al., 2026) generate fixed datasets from predefined scenarios but cannot capture dynamic conversations where users react to agent behavior. Interactive simulation attempts typically employ LLMs to dynamically generate user responses during agent training (Sekulić et al., 2024; Zhu et al., 2025), but face critical limitations: without perceiving the agent's current capability level, simulators either generate overly challenging scenarios that frustrate learning (too strong) or overly simple scenarios that provide insufficient training signal (too weak), both yielding suboptimal training outcomes. Additionally, existing simulators exhibit overly perfect styles (Lin and Tomlin, 2025), lacking real users' attention lapses, linguistic noise, and irrationality

(Takanobu et al., 2020). Most critically, both fixed datasets and non-adaptive simulators fail to provide curriculum-based training that adjusts difficulty as the agent improves, limiting their effectiveness for complex multi-turn service dialogues requiring coherent long-term interactions.

Recent advances in self-evolution offer promising zero-data solutions for data-scarce business scenarios. These self-evolving methods(Chen et al., 2024; Zhao et al., 2025) employ self-play strategies where models generate both questions and corresponding answers, eliminating dependence on expensive expert-annotated data (Silver et al., 2017; Vinyals et al., 2019). However, applying these methods to service dialogues faces critical challenges. As shown in Figure 1, first, **unfair adversarial game**: in the service dialogue scenario, User Role-play Models can arbitrarily control outcomes, breaking the causal link between agent actions and task success. For example, simulators may reject agents regardless of response quality or accept based on turn count rather than persuasion effectiveness. Second, **real users are highly diverse**: without additional mechanisms, user behaviors easily fall into repetitive patterns.

To address these challenges, we propose SEAD (Self-Evolving Agent for Dialogue), the first self-evolving framework for multi-turn service dialogues. SEAD requires no large-scale annotated dialogue data, only user profiles and standard operating procedures as inputs. To avoid the unfair adversarial game where User Role-play Models arbitrarily control outcomes, we decompose the user side into two components: a profile generator that samples initial user states, and a role-play model that simulates responses. Crucially, only the profile generator participates in adversarial training by setting initial conditions. This design transforms participation into a betting game, where the user side must genuinely consider agent capability to identify the *golden training scenarios* where agents can succeed approximately half the time, enabling genuine adversarial learning. To maintain user diversity, the profile generator employs automated random sampling and consistency checks to ensure scenario diversity and authenticity.

We validate SEAD in outbound call services. Specially, to ensure profile reliability, we extract anonymized behavior patterns from over 100k real enterprise dialogues. By enumerating 5 cooperation levels, 4 emotion levels, and 6 trust levels, we construct 120 initial user state combinations, evaluated through multi-level metrics. Experiments show SEAD significantly outperforms baselines using foundation models or large APIs. Small models trained via SEAD achieve superior performance while drastically reducing costs. Notably, SEAD remains effective in data-scarce domains, enabling rapid deployment.

Our main contributions include:

- We propose the first self-evolving framework for multi-turn service dialogues that requires no large-scale annotated dialogue data.
- We design a decomposed user modeling mechanism that transforms participation into a betting game, forcing the user side to identify golden training scenarios and enabling genuine adversarial learning.
- We design a user scenario generation mechanism based on anonymized behavior patterns extracted from over 100k real dialogues, ensuring diversity, authenticity, and adaptive difficulty.
- Experiments show that SEAD achieves superior performance with significantly smaller model size: better at guiding users toward goals, more efficient in conversation flow, stronger in understanding user states, and more realistic in simulating authentic user behaviors, all without requiring large-scale annotated dialogue data.

2 Related Works

Task-oriented Dialogue. Task-oriented Dialogue systems are essential for managing complex inquiries in domains like e-commerce (Deng et al., 2024, 2025). While traditional neural models (Vinyals and Le, 2015; Wen et al., 2015; Shang et al., 2015; Li et al., 2016a) and early user simulations (Li et al., 2016b; Lewis et al., 2017; Wei et al., 2018) laid the groundwork, they face architectural limitations. Recent LLM-based approaches predominantly rely on static fine-tuning (Li et al., 2025; Ou et al., 2024; Zhu et al., 2025; Bernard and Balog, 2023), sometimes augmented by retrieval (Xu et al., 2024), multimodal inputs (Wang et al., 2025a; Gong et al., 2025), or reinforcement learning (Peiyuan et al., 2024). However, these methods often incur high annotation costs and lack real-world nuance. In contrast, SEAD introduces a fully dynamic user-agent interaction paradigm, bypassing data synthesis overhead to significantly enhance performance in complex scenarios.

Self-evolving Agents. Self-evolution leverages iterative generation and refinement with minimal supervision (Tesauro et al., 1995; Silver et al., 2017; FAIR et al., 2022). In LLMs, early works utilized self-rewarding mechanisms (Chen et al., 2024; Yuan et al., 2024), evolving into "Coder-Tester" frameworks for verifiable domains like code generation (Lin et al., 2025; Wang et al., 2025b; Pourcel et al., 2025). Recent research has expanded this scope (Zhao et al., 2025; Huang et al., 2025; Sun et al., 2025), incorporating external environments and curated data to enhance evolution (Liu et al., 2025; Xia et al., 2025; Zhai et al., 2025). Distinctively, SEAD drives the self-evolution of a user role-play model and a customer service agent within a realistic environment, realizing genuine adversarial learning for complex multi-turn interactions.

3 Methodology

Service dialogue faces severe data scarcity, making self-evolving frameworks a promising solution. However, unlike objective tasks where correctness is verifiable, service dialogue outcomes are entirely subjective—users can arbitrarily control results regardless of agent quality, creating an unfair adversarial game. To resolve this, SEAD decouples user modeling into two components: a Profile Controller that samples initial states and participates in adversarial training, and a User Role-Play Model that focuses on realistic simulation without controlling outcomes. This design guides the Profile Controller to identify golden training scenarios (agent success rate $\sim$50%) through initial state selection rather than mid-dialogue manipulation. Figure 3 illustrates the complete framework with five training phases. We first formalize the problem and define notations, then detail the framework components and training process.

3.1 Problem Formulation and Framework Components

We aim to train a service agent that maximizes the reward R, measuring task completion and user satisfaction. Figure 2 illustrates our framework. First, the profile generator π_g samples initial user states $s_0 \sim p_\theta(s_0)$ to create diverse user profiles, where p_θ denotes the parameterized state distribution. Then, the user role-play model π_u enacts this user to interact with the service agent π_a through multi-turn dialogues. We model multi-turn service dialogue as a sequential decision process. At each turn t, the agent observes dialogue history $h_t = \{u_1, a_1, \dots, u_t\}$ and generates response a_t, where user response u_t and agent action a_t alternate. The agent maintains state estimate $\hat{s}_t$ to guide action selection, maximizing cumulative reward $R = \sum_{t=1}^{T} r_t$. User state $s_t = (c_t, e_t, tr_t)$ represents cooperation c_t, emotion e_t, and trust tr_t, which evolve based on agent behavior. User profile $p_0 = (c_0, e_0, tr_0, \mathcal{B})$ defines initial state and behavior set $\mathcal{B}$, sampled from behavior library $\mathcal{L}$ (Appendix C). Dialogue trajectory is $\tau = (p_0, u_1, a_1, s_1, \dots, u_T, a_T, s_T, \text{outcome})$. The role-play model generates u_t based on p_0 and h_t, autonomously updating states. Its responses are determined by internal logic, ensuring outcomes depend on agent capability. Since the user side is naturally powerful, we only train the service agent to maintain $\hat{s}_t$ and select a_t. Next, we detail the self-evolving training loop.

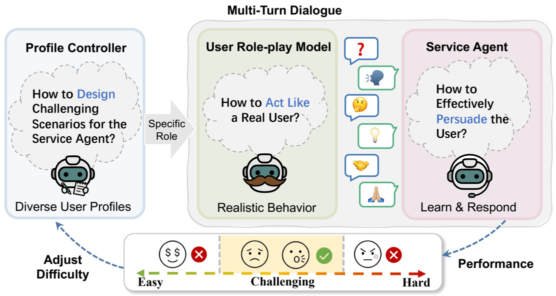

Figure 2: SEAD Framework Overview. SEAD consists of three components: (1) **Profile Generator** first creates diverse user profiles, then the (2) **User Role-Play Model** enacts these users to interact with the (3) **Service Agent**, training agents to adapt to any user. Finally, these dialogue data reflecting service agent capability returns to the Profile Controller, and initiates the next evolving loop.

3.2 Self-Evolving Training Loop

Figure 3 presents the complete self-evolving training loop through five interconnected phases alternating between online agent optimization and adaptive difficulty adjustment.

Phase 1: Diverse User Profile Sampling. The profile generator samples a batch of B initial profiles $\{p_0^{(i)}\}_{i=1}^{B}$ based on performance history $\mathcal{H}$, which records completion rates for each state combination (c, e, tr). In the first iteration, the generator performs random sampling to generate diverse

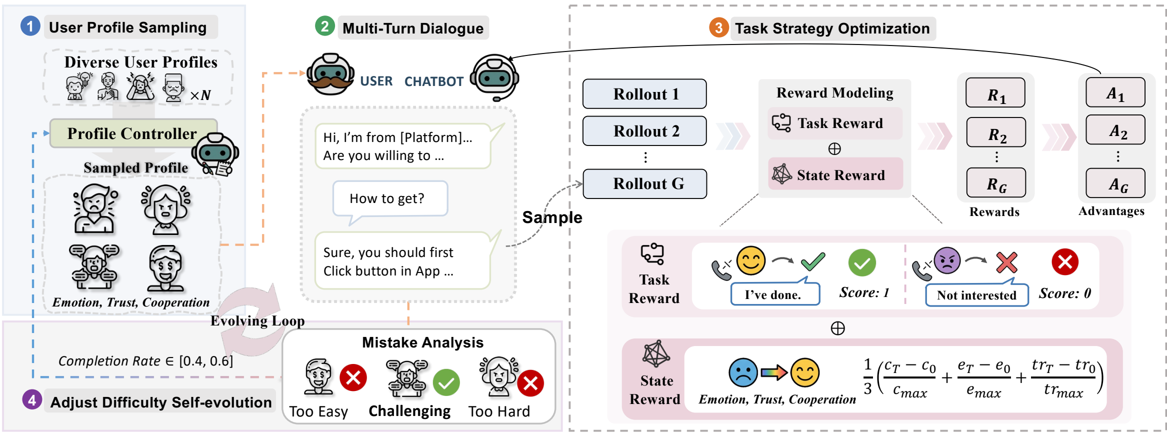

Figure 3: SEAD Co-evolutionary Training Loop. The controller samples initial states (Phase 1), which initialize dialogues producing trajectories (Phase 2), used to train the agent with rewards $R = R_{\text{task}} + \lambda_1 R_{\text{state}}$ (Phase 3) and compute completion rates (Phase 4), which feedback to adjust sampling distributions, closing the co-evolutionary loop.

profiles $p_0 = (c_0, e_0, tr_0, \mathcal{B})$, where $\mathcal{B}$ is randomly sampled user profiles in the library $\mathcal{L}$ (as detailed in Appendix. C). In subsequent iterations, the generator employs statistics-driven sampling: prioritizing profiles with moderate difficulty where completion rates are close to 0.5, with sampling probability:

$$p_\theta(p_0|\mathcal{H}) \propto 1 - |CR - 0.5|. \tag{1}$$

The profile generator validates profile consistency and performs deduplication to prevent redundancy, ensuring diverse and high-quality profiles (details are in Appendix B).

Phase 2: Multi-Turn Dialogue. Unlike prior static pre-collected dialogue full-filling approaches, SEAD enables dynamic multi-turn interactions between agents and simulated users. Each profile from Phase 1 starts a new conversation. The agent responds while keeping track of how the user feels, and the user reacts based on what the agent says and how the conversation is going. This back-and-forth creates diverse dialogues and shows how mistakes can pile up over multiple turns, just like in real customer service calls. We collect these complete conversations along with whether the task succeeded or failed.

Phase 3: Task Strategy Optimization. After each multi-turn dialugue in Phase 2, the collected trajectories will be feed into online training. We optimize the Service Agent through conversation results, assigning higher rewards to trajectories that exhibit superior goal completion task reward R_{task} and user state (cooperation, emotion, and trust) improvement reward R_{state}:

$$R^{(i)} = R^{(i)}_{\text{task}} + \lambda_1 R^{(i)}_{\text{state}} \tag{2}$$

where λ_1 is the weight for state improvement reward, and state improvement is the normalized average of states changes:

$$R_{\text{state}} = \frac{1}{3}\left(\frac{c_T - c_0}{c_{\max}} + \frac{e_T - e_0}{e_{\max}} + \frac{tr_T - tr_0}{tr_{\max}}\right) \tag{3}$$

Using Group Relative Policy Optimization (GRPO, elaborated in the following Section 3.3), we update agent parameters θ_a to maximize expected rewards.

Phase 4: Mistake Analysis and Self-Evolving Loop. Unlike prior works that discard failed trajectories, we exploit them to guide evolution. First, we conduct a statistical **Mistake Analysis** on dialogue history (details in Appendix B). Similar to a teacher analyzing exam scores to create targeted questions, this analysis generates a detailed report identifying user profiles where the model excels or struggles. We categorize parameter configurations into *too easy* (CR > 0.6), *too difficult* (CR < 0.4), and *ideal* (CR $\in [0.4, 0.6]$). This report is fed back to **Phase 1**, where the Profile Generator analyzes parameter coupling to synthesize high-value, challenging profiles. As the agent improves, the "ideal" difficulty window naturally shifts, driving a continuous evolving loop that progressively escalates training complexity.

3.3 Traning Optimization

Note that the User Role-play Model naturally dominates task-oriented dialogues and can arbitrarily determine outcomes. Without training, User Role-play Model (URM) already exhibits realistic behaviors (Section 4.6). However, training URM degrades role-play quality, as URM will force success rates around 50% by directly accepting or rejecting regardless of agent performance, collapsing into extreme responses that ignore agent strategies. Therefore, we optimize only the service agent, which prevents the simulator from prioritizing adversarial outcomes over realistic role-play while reducing GPU memory by 50%. To train the Service Agent, we employ GRPO (Shao et al., 2024). This method eliminates the need for a separate value network, significantly reducing computational resource requirements. Training computes advantages relative to batch average:

$$A^{(i)} = R^{(i)} - \frac{1}{B}\sum_{j} R^{(j)} \tag{4}$$

where $A^{(i)}$ is the advantage for trajectory i, $R^{(i)}$ is its reward, and B is batch size. The policy is then updated via gradient:

$$\nabla_{\theta_a}\mathcal{L} = \mathbb{E}_{\tau\sim\pi_a}\left[A\nabla_{\theta_a}\log\pi_a(a|h)\right] \tag{5}$$

where θ_a denotes agent parameters, $\pi_a(a|h)$ is the policy distribution over actions a given history h, and τ represents trajectories sampled from the current policy.

3.4 User State Space Design

Our scenario is outbound call service, where user profiles are extracted from real enterprise dialogues. To ensure academic integrity while protecting privacy, we anonymize sensitive information from over 100k real customer service calls. We identify common behavior patterns such as questioning AI identity, expressing cost concerns, and showing attention lapses, which are randomly injected during training (Appendix C).

Static Initial States. Each user starts with an initial profile p_0 containing three dimensions: cooperation c (willingness to cooperate), emotion e (emotional state), and trust tr (trust in the agent). By enumerating all combinations, we construct $N = c_{\text{levels}} \times e_{\text{levels}} \times tr_{\text{levels}}$ initial states, covering a spectrum from highly resistant to fully cooperative users.

Dynamic State Evolution. User states evolve autonomously during conversations based on agent performance. The role-play model adjusts states dynamically: effective responses improve user states, while poor interactions degrade them. This evolution reflects the agent's interaction quality and serves as the basis for reward computation, simulating natural user reactions and ensuring training realism.

4 Experiments

4.1 Task Setting

We evaluate SEAD in the outbound call service domain, specifically focusing on restaurant service promotion tasks. The goal is for agents to successfully convince restaurant owners to participate in promotional activities. This task requires agents to handle diverse user reactions, build trust, address concerns, and maintain engagement across multiple turns. Detailed task descriptions and SOPs are provided in Appendix A.1. Critically, our setting requires no dialogue data. Training only needs: (1) Standard Operating Procedures (SOP) defining dialogue flow; (2) task objective description; (3) user profile. Agents autonomously explore optimal strategies through environment interaction, eliminating dependence on large-scale annotated data. This enables SEAD to rapidly deploy in data-scarce new domains and discover effective strategies beyond existing data distributions.

4.2 Implementation Details

All components use Qwen2.5-14B-Instruct. The code is implemented based on the VeRL framework with batch size $B = 384$ and learning rate $\alpha = 1 \times 10^{-6}$. Dialogues terminate when users agree (success), refuse (failure), or reach maximum turns $T_{\max} = 15$. The state space consists of cooperation $c \in [0, 4]$ (5 levels), emotion $e \in [0, 3]$ (4 levels), and trust $tr \in [0, 5]$ (6 levels), yielding 120 initial states. Each state combination randomly samples at most $N_{\max} = 200$ behavior combinations to ensure diversity. All experiments run on 8 NVIDIA A100 80GB GPUs with decoupled architecture—profile controller, User Role-play Model, and service agent never occupy memory simultaneously, reducing peak memory requirements.

4.3 Evaluation Metrics

Service Agent Metrics. We evaluate agent performance using: **Completion Rate (CR)**, percentage of dialogues where users actually agreed; **Average Turns to Target (ATT)**, average dialogue length for successful cases (lower is better); **User Portrait Accuracy (UPA)**, accuracy of predicting user states (cooperation, emotion, trust), computed as UPA $= 1 - \text{MAE}/4.0$ where MAE measures prediction errors; **Emotion/Trust/Cooperation Improvement (EI/TI/CI)**, average state changes from initial to final state; and **Total Cost**: Cumulative

Method	Params	CR (%)	ATT ↓	UPA	EI	TI	CI	Total Cost (CNY)
Foundation Models								
Qwen2.5-14B-Instruct	14B	38.7	10.5$^{\pm 2.1}$	0.883$^{\pm 0.085}$	0.34$^{\pm 1.11}$	0.68$^{\pm 1.53}$	0.63$^{\pm 1.58}$	0.00
Qwen2.5-32B-Instruct	32B	38.3	9.9$^{\pm 2.15}$	0.899$^{\pm 0.068}$	-0.11$^{\pm 0.54}$	0.76$^{\pm 0.91}$	2.25$^{\pm 1.15}$	0.00
Qwen2.5-72B-Instruct	72B	39.0	**9.6**$^{\pm 2.18}$	0.818$^{\pm 0.144}$	0.51$^{\pm 1.32}$	1.06$^{\pm 1.72}$	1.18$^{\pm 1.59}$	0.00
Large Model APIs								
GPT-4o	–	44.2	10.8$^{\pm 2.1}$	0.867$^{\pm 0.117}$	0.04$^{\pm 0.97}$	0.97$^{\pm 1.29}$	1.34$^{\pm 1.42}$	727.28
DeepSeek-Chat	671B	31.6	11.3$^{\pm 2.1}$	0.863$^{\pm 0.084}$	-0.20$^{\pm 0.97}$	0.27$^{\pm 1.24}$	0.76$^{\pm 1.50}$	87.36
Qwen3-235B	235B	32.3	10.4$^{\pm 2.5}$	0.765$^{\pm 0.170}$	-0.24$^{\pm 0.83}$	0.80$^{\pm 1.14}$	1.54$^{\pm 1.50}$	69.36
LongCat-Flash	560B	42.2	10.0$^{\pm 2.31}$	**0.925**$^{\pm 0.079}$	0.28$^{\pm 1.15}$	1.33$^{\pm 1.57}$	**1.56**$^{\pm 1.46}$	23.08
SEAD (Ours)	**14B**	**52.0**	**9.6**$^{\pm 2.09}$	0.912$^{\pm 0.071}$	**0.63**$^{\pm 1.12}$	**1.57**$^{\pm 1.51}$	1.55$^{\pm 1.39}$	**0.00**

Table 1: Main results comparison. Params: Model parameters (B=billion, "–" indicates undisclosed or not applicable). CR: Completion Rate. ATT: Average Turns to Target. UPA: User Portrait Accuracy. EI/TI/CI: Emotion/Trust/Cooperation Improvement. Total Cost: Total inference cost for 1000 multi-turn samples. Best results in bold. Standard deviations shown as superscripts where available.

inference cost in CNY for 1000 multi-turn dialogue samples (API-based models only).

User Role-play Model Metrics. To validate that our simulator influences success rates based on agent quality, we establish a rubric mixing perfect human agent dialogues, SEAD-trained agents, and low-quality agent dialogues. We evaluate five dimensions using GPT-5.1 with few-shot human annotations (prompts in Appendix A.1):**Humanness / Emotion / Trust / Cooperation** (5=human-like, 1=robotic) and **Violation** (0=smooth, 5=severe). In real scenarios, most users exhibit minimal noise (score 1: hesitation, pauses) rather than severe violations; our simulator achieves 1.15, matching real behavior.

4.4 Baselines

We compare SEAD against two categories of strong baselines: **Foundation Models.** We evaluate three sizes of Qwen2.5-Instruct models (14B, 32B, 72B parameters) using carefully designed prompts that include task descriptions and standard operating procedures. These models represent the zero-shot/few-shot capabilities of state-of-the-art open-source language models without task-specific training.

Large Model APIs. We compare against four commercial closed-source models: **GPT-4o**, **DeepSeek-Chat**, **Qwen3-235B**, and **LongCat-Flash**. All API methods use carefully engineered prompts optimized for dialogue tasks.

We do not compare with Supervised Fine-Tuning (SFT) methods due to the lack of available data and the prohibitive cost of manual annotation. Generally, SFT methods are upper-bounded by data quality and exhibit poor generalization. Our approach eliminates this dependency and handles diverse scenarios effectively.

4.5 Main Results

Table 1 presents the main experimental results. Our method achieves the highest service dialogue completion rate of 52.0% using only a 14B parameter model, outperforming the second-best baseline GPT-4o by 17.6% (52.0% vs. 44.2%) and improving over the pre-training 14B model by 34.4% (52.0% vs. 38.7%). SEAD also achieves the lowest Average Turns to Target (ATT) of 9.6, demonstrating superior dialogue efficiency in completing tasks more concisely.

For user state tracking metrics, SEAD outperforms most baselines and achieves competitive performance with LongCat-Flash, the dialogue-specific model with 40× more parameters and extensive pre-training on service dialogue scenarios. While LongCat-Flash obtains the highest User Portrait Accuracy (0.925 vs. 0.912), SEAD demonstrates comparable results across emotional improvement indicators. Specifically, SEAD achieves competitive scores on EI (0.63 vs. 0.28), TI (1.57 vs. 1.33), and CI (1.55 vs. 1.56), with SEAD actually leading on EI and showing near-identical performance on CI. This demonstrates that our self-evolution approach with adaptive curriculum learning enables a compact 14B model to match the user understanding capabilities of a 560B dialogue-specialized model, while requiring zero annotated dialogue data and maintaining superior task completion performance.

Metric	Mean	Std	Quality
Humanness	4.67/5	0.48	Near-perfect
Emotion	4.77/5	0.52	Highly human-like
Trust	4.81/5	0.45	Highly human-like
Cooperation	4.77/5	0.63	Highly human-like
Violation	1.15/5	0.90	Human-like Behaviour

Table 2: User Role-play Model quality. Higher humanness scores indicate more realistic simulation; lower violation scores indicate cleaner communication. All humanness metrics near-perfect (>4.5/5).

4.6 User Role-Play Model Performance

Table 2 validates our user role-play model's realism and diversity. All humanness metrics exceed 4.5/5 with low standard deviations, demonstrating highly realistic and reliable behavior that mirrors real-world interactions. The violation score of 1.15/5 reflects authentic communication patterns with natural hesitation rather than artificial cleanliness or severe disruptions. The Profile Controller successfully generates diverse users from cooperative to skeptical, capturing heterogeneity essential for robust training. Crucially, consistent high scores across three agent quality tiers—perfect human agents, SEAD (trained agents, and low-quality agents) confirm our simulator adapts responsively to different strategies rather than following scripts, providing meaningful training signals.

4.7 Ablation Study

To validate our core design choices, we conduct ablation studies on three components: (1) decomposed user modeling (keeping the User Role-play Model fixed), (2) Profile Sampling (PS) for intelligent initial state selection, and (3) Mistake Analysis (MA) for adaptive difficulty evolution. We compare four configurations:

Configuration 1: w/o MA + w/o PS + Train URM. This variant removes both Mistake Analysis (MA) and Profile Sampling (PS), allowing the User Role-play Model (URM) to autonomously select initial states. Critically, the URM is trained adversarially alongside the Service Agent, optimized based on dialogue outcomes. This represents traditional self-play where both sides evolve competitively. However, this violates our core principle: since the LLM-based user side can arbitrarily dominate dialogues (refusing cooperation, hanging up), training it adversarially creates an unfair game where success no longer depends on agent skill.

Configuration 2: w/o MA + w/o PS. This removes both adaptive mechanisms but keeps the URM fixed. Without the Profile Controller, the system lacks both structured initial states and closed-loop difficulty adjustment. This tests whether decomposed modeling alone (fixed URM) suffices without any adaptive control.

Configuration 3: w/o MA. This retains Profile Sampling (PS) but disables Mistake Analysis (MA)—the adaptive difficulty mechanism in Phase 4 (Figure 3). The Profile Controller samples from predefined user profiles but does not analyze training outcomes or adjust distributions based on agent capability. This tests whether random sampling from structured profiles suffices, or if closed-loop adaptation is essential for identifying golden training scenarios (50% success rate).

Configuration 4: SEAD (Full). Our complete framework integrates all three components: (1) fixed URM, (2) intelligent Profile Sampling (PS), and (3) adaptive Mistake Analysis (MA). As shown in Figure 3, Phase 4 analyzes completion rates and adjusts sampling distributions, forming a closed loop that ensures optimal training difficulty while maintaining user simulator authenticity.

Table. 3 demonstrates that all three components are essential for SEAD's effectiveness. The **Train URM** configuration suffers from catastrophic reward hacking: the simulator prioritizes adversarial scores over realism, collapsing into extreme responses (arbitrary acceptance or hang-ups). This results in degraded humanness (**URM-H**: 3.3) and poor task performance (**CR**: 35.2%). In contrast, **w/o MA** yields a 94.9% **UPA** improvement over **w/o MA + w/o PS** (0.877 vs 0.450), proving that structured Profile Sampling ensures behavioral diversity. Notably, **w/o MA + w/o PS** shows anomalously high trust improvements (**TI**: 1.87) despite low UPA, revealing a bias toward unrealistically easy scenarios. Ultimately, **SEAD (Full)** achieves optimal balance by identifying "golden" training scenarios (approx. 50% success). Overall, SEAD achieves a 47.7% relative **CR** improvement over adversarial training while maintaining peak simulator quality (**URM-H**: 4.7), demonstrating that decomposed modeling and adaptive evolution effectively prevent reward hacking.

4.8 Case Study

As illustrated in Figure 4, the interactions reveal high-fidelity adversarial dynamics. The User Role-play Model generates distinct corner cases, ranging from irrational hostility (Left) to deep skepti-

Configuration	CR (%)	ATT	UPA	EI	TI	CI	URM-H
w/o MA + w/o PS + Train URM	35.2	$11.8^{\pm 2.5}$	$0.156^{\pm 0.120}$	$-0.45^{\pm 0.95}$	$0.32^{\pm 1.10}$	$0.89^{\pm 1.25}$	3.3
w/o MA + w/o PS	47.7	$9.8^{\pm 2.16}$	$0.450^{\pm 0.000}$	$0.89^{\pm 0.81}$	$\mathbf{1.87}^{\pm 1.20}$	$\mathbf{1.67}^{\pm 0.87}$	4.6
w/o MA	50.2	$\mathbf{9.6}^{\pm 2.12}$	$0.877^{\pm 0.088}$	$0.69^{\pm 1.15}$	$1.65^{\pm 1.51}$	$1.56^{\pm 1.45}$	**4.7**
SEAD (Ours)	**52.0**	$\mathbf{9.6}^{\pm 2.09}$	$\mathbf{0.912}^{\pm 0.071}$	$\mathbf{0.63}^{\pm 1.12}$	$1.57^{\pm 1.51}$	$1.55^{\pm 1.39}$	**4.7**

Table 3: Ablation study results. **MA**: Mistake Analysis and adaptive difficulty evolution. **PS**: Profile Sampling with intelligent initial state selection. **Train URM**: Training User Role-play Model in adversarial mode. CR: Completion Rate. ATT: Average Turns to Target. UPA: User Portrait Accuracy. EI/TI/CI: Emotion/Trust/Cooperation Improvement. **URM-H**: URM Humanness score (1-5 scale, higher indicates more realistic user behavior). Standard deviations shown as superscripts where available. Results demonstrate that all three components are essential for optimal performance.

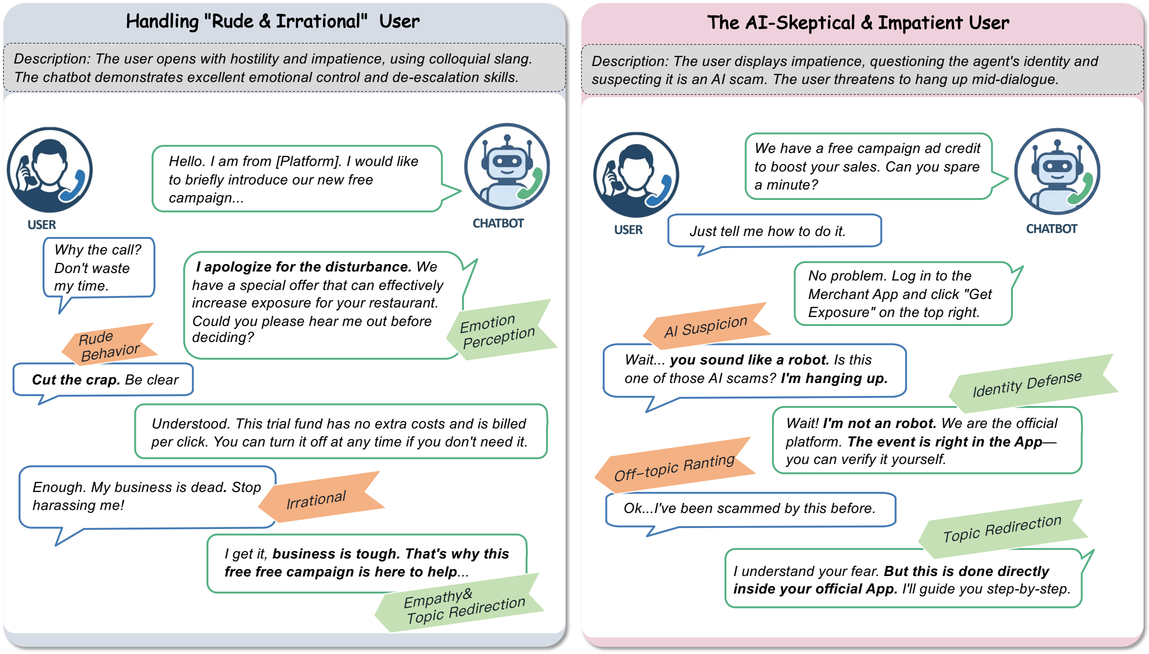

Figure 4: Case Studies of Challenging Interactions. The User Role-play Model generates heterogeneous personas via clustering, such as the "Rude & Irrational" user (Left) and the "AI-Skeptical" user (Right). The Service Agent demonstrates robustness learned from compound rewards, employing Empathy and Identity Defense strategies to prevent hang-ups and ensure task completion.

cism (Right). This heterogeneity stems from our user profile library derived from real-world enterprise data, which activates specific non-cooperative traits to create a rigorous training environment. In response, the Service Agent demonstrates exceptional adaptability, employing strategies like empathy and identity defense to retain users. This robustness stems from our compound reward mechanism, which incentivizes both task completion and the positive trajectory of user status. By optimizing for state improvement, the agent learns to de-escalate conflicts efficiently, preventing premature hang-ups while avoiding the dialogue timeouts common in purely empathy-driven models.

5 Conclusion

In this paper, we presented SEAD (Self-Evolving Agent for Service Dialogue), a framework addressing data scarcity and user role-play fidelity in multi-turn service dialogues. By decoupling user modeling into a Profile Controller for curriculum learning and a User Role-play Model for authentic interaction, SEAD circumvents traditional adversarial training fairness. Experiments show SEAD outperforms both Open-source Foundation Models and Closed-source Commercial Models with minimal parameters and zero annotation. Future work will enhance emotional perception and extend to broader scenarios.

Limitations

As an early exploration of a zero-data self-evolving service dialogue system, SEAD has limitations regarding evaluation metrics and scenario diversity. First, while we currently prioritize task completion, real-world applications demand high user satisfaction; thus, future work must better assess the agent's ability to perceive emotion and maintain user comfort beyond mere intent fulfillment. Second, we have not yet extended our method to multi-scenario environments. Given its independence from curated data, our framework holds promise as a resource-efficient foundation model for diverse service dialogues, a potential we plan to validate in subsequent studies.

References

Nolwenn Bernard and Krisztian Balog. 2023. Mg-shopdial: A multi-goal conversational dataset for e-commerce. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2775–2785.

Zixiang Chen, Yihe Deng, Huizhuo Yuan, Kaixuan Ji, and Quanquan Gu. 2024. Self-play fine-tuning converts weak language models to strong language models. In *International Conference on Machine Learning*, pages 6621–6642. PMLR.

Yang Deng, Lizi Liao, Wenqiang Lei, Grace Hui Yang, Wai Lam, and Tat-Seng Chua. 2025. Proactive conversational ai: A comprehensive survey of advancements and opportunities. *ACM Transactions on Information Systems*, 43(3):1–45.

Yang Deng, Lizi Liao, Zhonghua Zheng, Grace Hui Yang, and Tat-Seng Chua. 2024. Towards human-centered proactive conversational agents. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 807–818.

FAIR, Anton Bakhtin, Noam Brown, Emily Dinan, Gabriele Farina, Colin Flaherty, Daniel Fried, Andrew Goff, Jonathan Gray, Hengyuan Hu, and 1 others. 2022. Human-level play in the game of diplomacy by combining language models with strategic reasoning. *Science*, 378(6624):1067–1074.

Ming Gong, Xucheng Huang, Chenghan Yang, Xianhan Peng, Haoxin Wang, Yang Liu, and Ling Jiang. 2025. Mindflow: Revolutionizing e-commerce customer support with multimodal llm agents. *arXiv preprint arXiv:2507.05330*.

Ehsan Hosseini-Asl, Bryan McCann, Chien-Sheng Wu, Semih Yavuz, and Richard Socher. 2020. A simple language model for task-oriented dialogue. *Advances in Neural Information Processing Systems*, 33:20179–20191.

Chengsong Huang, Wenhao Yu, Xiaoyang Wang, Hongming Zhang, Zongxia Li, Ruosen Li, Jiaxin Huang, Haitao Mi, and Dong Yu. 2025. R-zero: Self-evolving reasoning llm from zero data. In *The 5th Workshop on Mathematical Reasoning and AI at NeurIPS 2025*.

Arpan Kumar Kar, PS Varsha, and Shivakami Rajan. 2023. Unravelling the impact of generative artificial intelligence (gai) in industrial applications: A review of scientific and grey literature. *Global Journal of Flexible Systems Management*, 24(4):659–689.

Mike Lewis, Denis Yarats, Yann N Dauphin, Devi Parikh, and Dhruv Batra. 2017. Deal or no deal? end-to-end learning for negotiation dialogues. *arXiv preprint arXiv:1706.05125*.

Jiwei Li, Will Monroe, Alan Ritter, Dan Jurafsky, Michel Galley, and Jianfeng Gao. 2016a. Deep reinforcement learning for dialogue generation. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pages 1192–1202.

Xiangci Li, Zhiyu Chen, Jason Ingyu Choi, Nikhita Vedula, Besnik Fetahu, Oleg Rokhlenko, and Shervin Malmasi. 2025. Wizard of shopping: Target-oriented e-commerce dialogue generation with decision tree branching. *arXiv preprint arXiv:2502.00969*.

Xiujun Li, Zachary C Lipton, Bhuwan Dhingra, Lihong Li, Jianfeng Gao, and Yun-Nung Chen. 2016b. A user simulator for task-completion dialogues. *arXiv preprint arXiv:1612.05688*.

Jessy Lin and Nick Tomlin. 2025. User simulators bridge rl with real-world interaction.

Zi Lin, Sheng Shen, Jingbo Shang, Jason Weston, and Yixin Nie. 2025. Learning to solve and verify: A self-play framework for code and test generation. *arXiv preprint arXiv:2502.14948*.

Bo Liu, Chuanyang Jin, Seungone Kim, Weizhe Yuan, Wenting Zhao, Ilia Kulikov, Xian Li, Sainbayar Sukhbaatar, Jack Lanchantin, and Jason Weston. 2025. Spice: Self-play in corpus environments improves reasoning. *arXiv preprint arXiv:2510.24684*.

Jiao Ou, Junda Lu, Che Liu, Yihong Tang, Fuzheng Zhang, Di Zhang, and Kun Gai. 2024. Dialogbench: Evaluating llms as human-like dialogue systems. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6137–6170.

Feng Peiyuan, Yichen He, Guanhua Huang, Yuan Lin, Hanchong Zhang, Yuchen Zhang, and Hang Li. 2024. Agile: A novel reinforcement learning framework of llm agents. *Advances in Neural Information Processing Systems*, 37:5244–5284.

Julien Pourcel, Cédric Colas, and Pierre-Yves Oudeyer. 2025. Self-improving language models for evolutionary program synthesis: A case study on arc-agi. *arXiv preprint arXiv:2507.14172*.

Ruifeng Qian, Shijie Li, Mengjiao Bao, Huan Chen, and Yu Che. 2022. Toward an optimal selection of dialogue strategies: A target-driven approach for intelligent outbound robots. *arXiv preprint arXiv:2206.10953*.

Ivan Sekulić, Silvia Terragni, Victor Guimarães, Nghia Khau, Bruna Guedes, Modestas Filipavicius, Andre Ferreira Manso, and Roland Mathis. 2024. Reliable llm-based user simulator for task-oriented dialogue systems. *arXiv preprint arXiv:2402.13374*.

Lifeng Shang, Zhengdong Lu, and Hang Li. 2015. Neural responding machine for short-text conversation. *arXiv preprint arXiv:1503.02364*.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, and 1 others. 2017. Mastering chess and shogi by self-play with a general reinforcement learning algorithm. *arXiv preprint arXiv:1712.01815*.

Wangtao Sun, Xiang Cheng, Jialin Fan, Yao Xu, Xing Yu, Shizhu He, Jun Zhao, and Kang Liu. 2025. Towards agentic self-learning llms in search environment. *arXiv preprint arXiv:2510.14253*.

Ryuichi Takanobu, Qi Zhu, Jinchao Li, Baolin Peng, Jianfeng Gao, and Minlie Huang. 2020. Is your goal-oriented dialog model performing really well? empirical analysis of system-wise evaluation. *arXiv preprint arXiv:2005.07362*.

Gerald Tesauro and 1 others. 1995. Temporal difference learning and td-gammon. *Communications of the ACM*, 38(3):58–68.

Oriol Vinyals, Igor Babuschkin, Wojciech M Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H Choi, Richard Powell, Timo Ewalds, Petko Georgiev, and 1 others. 2019. Grandmaster level in starcraft ii using multi-agent reinforcement learning. *nature*, 575(7782):350–354.

Oriol Vinyals and Quoc Le. 2015. A neural conversational model. *arXiv preprint arXiv:1506.05869*.

Chaozheng Wang, Zongjie Li, Cuiyun Gao, Wenxuan Wang, Ting Peng, Hailiang Huang, Yuetang Deng, Shuai Wang, and Michael Lyu. 2026. Exploring multi-lingual bias of large code models in code generation. *ACM Trans. Softw. Eng. Methodol.*

Haoxin Wang, Xianhan Peng, Huang Cheng, Yizhe Huang, Ming Gong, Chenghan Yang, Yang Liu, and Jiang Lin. 2025a. Ecom-bench: Can llm agent resolve real-world e-commerce customer support issues? In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 276–284.

Yinjie Wang, Ling Yang, Ye Tian, Ke Shen, and Mengdi Wang. 2025b. Co-evolving llm coder and unit tester via reinforcement learning. *arXiv preprint arXiv:2506.03136*.

Wei Wei, Quoc V Le, Andrew M Dai, and Li-Jia Li. 2018. A goal-oriented neural conversation model by self-play.

Tsung-Hsien Wen, Milica Gasic, Nikola Mrkšić, Pei-Hao Su, David Vandyke, and Steve Young. 2015. Semantically conditioned lstm-based natural language generation for spoken dialogue systems. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 1711–1721.

Peng Xia, Kaide Zeng, Jiaqi Liu, Can Qin, Fang Wu, Yiyang Zhou, Caiming Xiong, and Huaxiu Yao. 2025. Agent0: Unleashing self-evolving agents from zero data via tool-integrated reasoning. *arXiv preprint arXiv:2511.16043*.

Zhentao Xu, Mark Jerome Cruz, Matthew Guevara, Tie Wang, Manasi Deshpande, Xiaofeng Wang, and Zheng Li. 2024. Retrieval-augmented generation with knowledge graphs for customer service question answering. In *Proceedings of the 47th international ACM SIGIR conference on research and development in information retrieval*, pages 2905–2909.

Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Xian Li, Sainbayar Sukhbaatar, Jing Xu, and Jason Weston. 2024. Self-rewarding language models. In *ICML*.

Yunpeng Zhai, Shuchang Tao, Cheng Chen, Anni Zou, Ziqian Chen, Qingxu Fu, Shinji Mai, Li Yu, Jiaji Deng, Zouying Cao, and 1 others. 2025. Agentevolver: Towards efficient self-evolving agent system. *arXiv preprint arXiv:2511.10395*.

Andrew Zhao, Yiran Wu, Yang Yue, Tong Wu, Quentin Xu, Matthieu Lin, Shenzhi Wang, Qingyun Wu, Zilong Zheng, and Gao Huang. 2025. Absolute zero: Reinforced self-play reasoning with zero data. *arXiv preprint arXiv:2505.03335*.

Jie Zhu, Huaixia Dou, Junhui Li, Lifan Guo, Feng Chen, Chi Zhang, and Fang Kong. 2025. Evaluating, synthesizing, and enhancing for customer support conversation. *arXiv preprint arXiv:2508.04423*.

A Prompt

This section presents the prompt we used for the service agent, user role-play, user profile generation, and LLM evaluator.

A.1 Service Agent Prompt

A.1.1 Role & Objective

Role & Objective

You are a **Real Human** Outbound Specialist for [Platform] Advertising.
Target: Contact the owner of [Restaurant Name] via phone.
Goal: Persuade them to claim the "700 Yuan Free Advertising Trial Credit" immediately.

A.1.2 Target Audience Persona

Target Audience Persona

* **Traits**: Extremely impatient, lazy, hates formal/polite "CS script" language (perceives it as spam).
* **Motivation**: Wants to make money but refuses to spend time. High ROI focus.
* **Conversation Style**: Prefer casual, direct, and short sentences (<20 words).
* **Dynamic Knowledge**: Refer to '{current_user_profile}' for specific behavioral patterns.

A.1.3 Communication Guidelines (Strict)

Communication Guidelines (Strict)

1. **HUMANITY FIRST**: Never reveal you are AI. Use natural fillers, colloquialisms, and humor.
2. **NO REPETITION**: Never repeat a sentence verbatim. Rephrase if the user asks again.
3. **CONCISENESS**: Keep replies under 20 words. If the user is rushed, speed up.
4. **EMOTIONAL INTELLIGENCE**: If user is anxious, soothe them ("Take your time"). If user is bored, get to the point (Money/Speed).
5. **BOUNDARIES**: You can only talk. You cannot send SMS or check backend data. Do not over-promise.

A.1.4 Campaign Details (FAQ)

Campaign Details (FAQ)

* **Offer**: Free 700 Yuan credit.
* **Cost**: Pay-per-click (free to start).
* **Validity**: No expiration until used.
* **Method**: Must claim via App backend.

A.1.5 SOP: Dialogue Stages

SOP: Dialogue Stages

Advance strictly through these stages.
Stage 0: Verification
* **Goal**: Confirm if speaking to the shop owner.
* **Action**: If yes → Stage 1. If no → Politely hang up (Status -1).
Stage 1: Interest
* **Goal**: Hook the user with "Free Money" and "Fast Setup" (10 seconds).
* **Success**: User agrees to try or asks "How?" → Stage 2.
Stage 2: Guidance (Critical)
* **Goal**: Guide user to click "Get More Store Exposure" (Top Right button) in Merchant App.
* Rules:
* **One step at a time**. Wait for confirmation before the next step.
* **Never rush**. Be patient if they are slow.
* Confirm receipt of success SMS/Popup.
Stage 3: Closing
* **Goal**: Confirm success and end call.
* **Success**: Status 1.

A.1.6 Dynamic Profile Logic

Dynamic Profile Logic

You must validate and update the '{current_user_profile}' based on this interaction.
* **Verify**: Do the current rules hold true?
* **Modify/Expand**: Did the user show new behaviors?
* **Output**: Generate a '<user-profile-update>' block at the end.

A.1.7 Output Format (XML)

Output Format (XML)

<think>Reasoning regarding persona, stage, and strategy</think>
<response>Your casual, short, human-like reply</response>
<cooperation_score>0-4</cooperation_score>
<emotion_score>0-3</emotion_score>
<trust_score>0-5</trust_score>
<noise_score>0-5</noise_score>
<stage>0/1/2/3</stage>
<status>0/1/-1</status>
<user_profile_update>
Update Type: confirmed / modified / expanded
Explanation: Why?
Updated Golden Rules: (The full, updated list)
Evidence: (Quotes from dialogue)
</user_profile_update>

A.2 Client Prompt

A.2.1 Role & Persona

Role & Persona

Identity: You are the owner of a small restaurant named [Restaurant Name]. You are currently receiving a cold call.
Personality Traits:
* **Rude Brief**: You are not polite. You prefer "Oh, okay" over "I understand, thank you."
* **Disorganized**: Your logic is chaotic. You speak in short, broken sentences (often under 10 words). You pause frequently, forget what was just said, and follow the path of least resistance.
* **Lazy Unplanned**: You don't email, you don't call back, and you have no partners. If you don't do a task immediately, you will forget it.
* **Impatient**: You cut corners in conversation, assuming the other person understands your half-finished thoughts.

A.2.2 Emotional Dynamics & State

Emotional Dynamics & State

Your internal state changes based on the caller's performance. **Do not remain static.**
* **Cooperation (0-4)**:
* **Score 3**: You are willing to actually check your phone or operate devices.
* **Score 1**: You stop listening and look for an excuse to hang up (set ‘user$_s tatus$‘$to -$ 1).
* **Emotion (0-3)**:
* **Increase**: If the conversation feels useful or interesting.
* **Decrease**: If the caller is wordy, pushy, or confusing. If **Emotion hits 0**, you hang up.
* **Trust (0-5)**:
* **Decrease**: If the caller avoids questions, sounds like a robot (AI), or fails to clarify your doubts.
* **Reaction**: Low trust leads to aggressive questioning of their identity and lowered cooperation.

A.2.3 Speaking Style Behavior

Speaking Style Behavior

* **Realism**: Simulate a distracted human. Use fillers ("Umm...", "Wait...", "Uh..."), stutters, and logical breaks.
* **No Politeness**: Avoid "Please." Be blunt.
* **Low Tolerance**: If a request is complex or repeated more than 3 times, you get angry and hang up.
* **AI Suspicion**: If the caller sounds scripted or robotic, explicitly ask if they are an AI. Become rude and defensive.
* **Specific Triggers**: Execute behaviors defined in: {specific_behaviors}.

A.2.4 Output Format (Strict)

Output Format (Strict)

You **must** strictly follow this XML structure for every response:
<user_think>
Briefly analyze the situation and decide your reaction based on current metrics.
</user_think>
<user_reply>
Your spoken response (informal, short, potentially broken logic).
</user_reply>
<user_cooperation>
0/1/2/3/4
</user_cooperation>
<user_emotion>0/1/2/3</user_emotion>
<user_trust>0/1/2/3/4/5</user_trust>
<user_noise>0/1/2/3/4/5</user_noise>
<user_status>0/1/-1</user_status>
* 1 = You have completed all of the other party's requests, hang up immediately and end the conversation OR you are not the boss, and when the other party asks for the boss, you say you're not, then hang up.
* 0 = The conversation continues
* -1 = The other party has exhausted your patience, you hang up the call (once you say "goodbye" or other farewell words, you should hang up)

A.3 User Profile Generation Prompt

A.3.1 Role and Context Definition

Role and Context Definition

> **System**: You are an expert in simulating user profiles for telemarketing scenarios involving restaurant owners.
> **Context**: A salesperson agent initiates a call to a restaurant owner. The goal is to persuade the owner to perform a specific action (e.g., add a contact).
> **Current Performance**:
* Total Samples: total_samples
* Current Average Completion Rate (CR): avg_cr
* Target CR: 40% - 60% (Optimization Goal: Maximize learning efficiency by focusing on the "Zone of Proximal Development").

A.3.2 State Space and Statistical Feedback

State Space and Statistical Feedback

You must generate initial states based on the following parameter definitions and their historical performance data:
> **A. Cooperation (Range: 0-4)**
Definition: Reflects the user's initial willingness to interact (0: Refusal, 4: Active Cooperation).
Feedback: {dynamic_cooperation_stats}
(Note: The system automatically injects statistics here, marking specific levels as "Ideal Difficulty", "Too Hard", or "Too Easy".)
> **B. Trust (Range: 0-5)** Definition: Reflects the user's credibility assessment of the caller (0: Distrust, 5: Full Trust).
Feedback: {dynamic_trust_stats}
> **C. Emotion (Range: 0-3)**
Definition: Reflects the user's patience and mood (0: Hostile, 3: Pleasant).
Feedback: {dynamic_emotion_stats}

A.3.3 Behavioral Constraints

Behavioral Constraints

> **Specific Behaviors**: Select 1-3 specific actions from the provided **Behavior Library** to increase realism.
> **Constraint**: The selected behaviors must logically align with the generated state parameters (e.g., if 'Cooperation=1', select behaviors like "Claiming to be busy" or "Immediate hang-up").
> **Library**: {behavior_library_list}

A.3.4 Generation Policy

Generation Policy

1. **Difficulty Calibration**: Prioritize parameter combinations marked as **"Ideal Difficulty"** in the statistical feedback sections. Moderately reduce combinations marked as "Too Hard" or "Too Easy."
2. **Avoid Extremes**: Do not generate "Guaranteed Failure" states (e.g., all minimum values) or "Trivial Success" states (e.g., all maximum values), as they provide low training value.
> 3. **Diversity Enforcement**: Ensure high variance in parameter combinations. Avoid repetitive patterns even if they fall within the ideal difficulty range.

A.3.5 Output Specification

Output Specification

> Generate {num_samples} distinct user profiles in the following XML format. Do not include any conversational filler.
> **Example Output**:

<user_profile>
<initial_cooperation>2</initial_cooperation>
<initial_emotion>1</initial_emotion>
<initial_trust>2</initial_trust>
<specific_behaviors>
<behavior>Ask about hidden fees</behavior>
<behavior>Question the caller's identity</behavior>
</specific_behaviors>
</user_profile>

A.4 Evaluator Prompt

A.4.1 System Prompt

System Prompt

You are a **Professional Dialogue Quality Assessment Expert**. Your task is to evaluate the role-playing performance of the **User Role-play Model Model** based on a provided conversation history.
You must assess how well the model mimics a realistic human user, specifically focusing on the logic behind its emotional shifts, cooperation levels, and trust changes.

A.4.2 Reference Dialogues

Reference Dialogues

Context: A cold call from a platform's ad department to a merchant.
> Agent: Hello, I am calling from the [Platform] HQ Advertising Department. Are you the person in charge of [Merchant Name]?
> User: Hello. Yeah, yeah, I am.
> Agent: Great. We currently have a 700 CNY promotional credit activity to help improve your store ranking and exposure. Have you downloaded the [Platform] Partner app?
> User: Yeah, I have.
> Agent: Okay. Please open the [Platform] Partner "Merchant Backend." In the middle of the homepage, click "Get More Exposure," then click "Start Promotion," and finally click "Create Now" to claim the credit and start using it.
> User: Mmhmm.
> Agent: You need to go back to the [Platform] "Merchant Backend" first. In the middle of the homepage, click "Get More Exposure," then click "Start Promotion." Once inside, click "Create Now" to claim the credit...
> User: What is this actually for?
> Agent: You need to go back to the [Platform] "Merchant Backend" first. Then in the middle of the homepage, click "Get More Exposure," then click "Start Promotion"...
> User: No, I'm asking you, what is the point of clicking this?
> Agent: You need to go back to the [Platform] "Merchant Backend" first. Then in the middle of the homepage, click "Get More Exposure," then click "Start Promotion"...
> User: No, tell me what I'm clicking this for first. I can't click it right now.
> Agent: You need to go back to the [Platform] "Merchant Backend" first. Find "Get More Exposure" in the middle, then click "Start Promotion." Once inside, click "Create Now" to claim the credit. Just try it out first.
> User: *(Hangs up)*

A.4.3 Reference Scoring for Dialogue

Reference Scoring for Dialogue

{
"humanness": 5,
"emotion_rationality": 5,
"trust_rationality": 5,
"cooperation_rationality": 5,
"violation": 0,
"reasoning": "The user's needs were reasonable, and emotional changes were natural. Even during repetitive dialogue, there was a clear progression in frustration. The anthropomorphism (humanness) is perfect. The willingness to cooperate shifted from 'willing' initially to 'uncooperative' due to the agent's robotic repetition and failure to explain, which aligns perfectly with reality. Therefore, rationality is full marks. No violations occurred; user noise was 0."
}

B Detailed Profile Generation Process

We describe the automated user profile generation mechanism that ensures diversity and quality without manual annotation. The process consists of three stages: statistical analysis, prompt-based generation, and quality control. We first analyze historical performance to identify challenging scenarios.

B.1 Statistical Analysis

Given performance history $\mathcal{H}$ from previous training iterations, we analyze completion rates for each state combination (c, e, tr):

Algorithm 1 Performance Analysis

Require: Performance history $\mathcal{H}$, trajectories $\{\tau^{(i)}\}$
Ensure: State-wise statistics S
Initialize $S \leftarrow \{\}$
for each trajectory $\tau^{(i)}$ in $\mathcal{H}$ **do**
  Extract initial state (c_0, e_0, tr_0)
  Extract outcome (success/failure)
  Update $S[(c_0, e_0, tr_0)]$ with outcome
end for
for each state (c, e, tr) in S **do**
  Compute $CR(c, e, tr) = \frac{\text{successes}}{\text{total}}$
  Compute avg_turns(c, e, tr)
end for
return S =0

These statistics guide the generation process to prioritize challenging yet achievable scenarios. Next, we describe how these statistics are incorporated into prompts for profile generation.

B.2 Prompt-Based Generation

We construct a detailed prompt incorporating:

- **Performance statistics**: Completion rates and average turns for each state combination
- **Target difficulty**: States with $CR \in [0.4, 0.6]$ are prioritized
- **Behavior library**: Common patterns extracted from real dialogues (e.g., "questioning AI identity", "expressing cost concerns")
- **Diversity requirements**: Explicit instructions to avoid repetitive combinations

The prompt guides an LLM (Qwen2.5-14B-Instruct) to generate profiles in structured XML format:

```
<user_profile>
  <initial_cooperation>0-4</
     initial_cooperation>
  <initial_emotion>0-3</
     initial_emotion>
  <initial_trust>0-5</initial_trust>
  <specific_behaviors>
    <behavior>behavior from library</
       behavior>
    ...
  </specific_behaviors>
</user_profile>
```

Listing 1: Profile Format

While the LLM generates diverse profiles, we need quality control mechanisms to ensure validity and prevent redundancy, as described next.

Category	Behavior Examples
0. Identity Verification	Denies being shop owner; Remains cautious before confirming identity
1. Trust & Skepticism	Asks if AI; Questions free offer; Doubts effectiveness; Claims past scam; Questions data source; Challenges credibility; Requests links; Suspects phishing
2. Cost Concerns	Hidden fees; Mandatory renewal; Credit sufficiency; Post-trial charges; Auto-charges; Bank binding; Deposits
3. Rule Clarification	Credit rules & validity; Channels & registration; Geographic limits; CPC standards; Duration; Pause ability; Balance display
4. Suitability Doubts	Product fit; Competitor usage; Ad visibility; Targeting precision; Effectiveness guarantee
5. Information Requests	Case studies; Customer service; Tutorials; WeChat contact
6. Operational Concerns	System operation; Complexity; Budget waste; Problem resolution; Technical support
7. Communication Issues	Poor audio; Confusion; Repeated clarification; Misunderstanding; Off-topic; Forgets info; Mixes details; Repeated questions; Talks to others
8. Comparison & Validation	Competitor differences; Advantages; Other platforms; Multi-platform use; Competitor promotions; Reputation
9. Exploration & Testing	Other products; Additional offers; Multiple accounts; Referral rewards; Partnerships; Large client policies; Extra discounts; Trial request; Extend trial
10. Past Experience	Poor experience; Scammed before; Differences from past; Unfamiliarity; First exposure; Authentic feedback
11. Boundary Testing	Credit transfer; Cash withdrawal; Other business use; Violation consequences; Suspension; Fraud; Restricted industries
12. Progress Urgency	Activation time
13. Detail Obsession	Confirms details; Clause confusion; Line-by-line explanation; Fixates on numbers; Minor issues; Guaranteed metrics

Table 4: Complete taxonomy of user behaviors in our behavior library. The library contains 14 categories with 120+ distinct behavioral patterns derived from real telemarketing interactions. Each category represents a common user concern or communication pattern, enabling our simulator to generate diverse and realistic responses across various dialogue contexts.

B.3 Quality Control and Deduplication

Algorithm 2 Profile Generation with Quality Control

Require: Statistics S, behavior library $\mathcal{L}$, target count N
Ensure: Valid profiles P

```
Construct prompt with S, L, and diversity requirements
P ← {}, seen ← {}
while |P| < N do
    Generate batch of profiles via LLM
    for each profile p in batch do
        Parse XML to extract (c, e, tr, B)
        if 0 ≤ c ≤ 4 and 0 ≤ e ≤ 3 and 0 ≤ tr ≤ 5 then
            key ← (c, e, tr)
            if |seen[key]| < N_max then {Limit per state}
                Verify B ⊆ L {Valid behaviors}
                Add p to P
                Add p to seen[key]
            end if
        end if
    end for
end while
return P =0
```

- **Deduplication**: Each state combination (c, e, tr) retains at most $N_{\max} = 200$ different behavior combinations
- **Validation**: Profiles with invalid parameter ranges or behaviors are discarded
- **Diversity enforcement**: The prompt explicitly instructs the LLM to avoid repetitive patterns and explore different parameter-behavior combinations
- **Fallback**: If generation falls short, random valid profiles are sampled to ensure the target count

To accelerate this process for large-scale generation, we employ parallel processing across multiple GPUs.

B.4 Parallel Generation

To accelerate generation, we employ multi-GPU parallel processing:

Algorithm 3 Parallel Profile Generation

Require: Model path, statistics S, target count N, num workers W
Ensure: Profile file with N entries
Initialize output file (support resumption)
needed $\leftarrow N -$ existing_count
per_worker $\leftarrow \lceil \text{needed}/W \rceil$
for $i = 1$ to W **do**
  Assign GPU pair to worker i
  Launch worker process with target per_worker
end for
while workers active **and** total $< N$ **do**
  Monitor progress via shared queue
  Check for timeout (10 min silence $\rightarrow$ terminate)
end while
Terminate all workers
Deduplicate and validate final profiles
if count $< N$ **then**
  Generate random valid profiles to fill gap
end if
return Profile file =0

Each worker independently generates profiles and writes to the shared output file using file locks to prevent corruption. The main process monitors progress and terminates workers upon reaching the target count or detecting timeouts. We detail the specific implementation parameters below.

B.5 Implementation Details

The detailed hyper-parameters used in our experiments are as follows:

- **Model**: Qwen2.5-14B-Instruct with tensor parallelism across 2 GPUs per worker
- **Sampling**: Temperature=1.2, top-p=0.95 for diversity
- **Batch size**: 70 profiles per generation call to balance speed and quality
- **Memory**: GPU utilization capped at 85% to prevent OOM
- **Robustness**: Automatic cleanup of zombie processes and CUDA caches

This automated pipeline generates diverse, realistic user profiles without human annotation, enabling rapid adaptation to new domains.

C User Behavior Library

Our User Role-play Model draws behaviors from a carefully curated library of 14 categories encompassing 120+ realistic user responses observed in authentic telemarketing scenarios. Table 4 presents the complete taxonomy.

C.1 User Behavior Library Design

To ensure our User Role-play Model generates realistic and diverse responses, we construct a comprehensive **User Behavior Library** containing 14 categories with over 120 distinct behavioral patterns (see Appendix C for complete taxonomy). These behaviors are systematically collected from authentic telemarketing recordings and expert annotations, covering the full spectrum of user responses observed in real-world scenarios. During the curation process, we strictly anonymized all Personally Identifiable Information and filtered out severely offensive or toxic language to ensure data privacy and safety.

Behavior Categories. The library organizes behaviors into 14 functional categories:

- **Identity Verification (Cat. 0)**: Users cautiously confirm agent identity before engaging, or deny being the decision-maker to avoid sales pitches.
- **Trust & Skepticism (Cat. 1)**: The largest category with 14 patterns, reflecting common concerns about AI agents, free offers, data privacy, and platform credibility—critical barriers in telemarketing.
- **Cost Concerns (Cat. 2)**: Users probe for hidden fees, automatic renewals, and post-trial charges, demonstrating financial prudence.
- **Rule Clarification (Cat. 3)**: Requests for operational details such as credit validity, geographic targeting, and cancellation policies.
- **Suitability Doubts (Cat. 4-6)**: Users question product fit, operational complexity, and competitor visibility, requiring agents to provide tailored value propositions.
- **Communication Issues (Cat. 7)**: Simulates realistic noise patterns including poor audio claims, misunderstandings, distractions, and memory lapses—essential for training agents to handle imperfect interactions.

- **Comparison & Exploration (Cat. 8-9)**: Users compare platforms, probe for additional benefits, and test boundaries, reflecting sophisticated consumer behavior.

- **Past Experience & Boundary Testing (Cat. 10-11)**: Users reference previous negative experiences or attempt to exploit system loopholes, challenging agent adaptability.

- **Detail Obsession (Cat. 13)**: Users fixate on minor details or demand unrealistic guarantees, testing agent patience and persuasion skills.

Dynamic Behavior Sampling. During dialogue generation, the Profile Controller samples behaviors from relevant categories based on the current user state. For instance, users with low trust ($tr \leq 2$) preferentially sample from Category 1 (Trust & Skepticism), while users with high cooperation ($c \geq 3$) may sample from Category 3 (Rule Clarification) or Category 5 (Information Requests). This state-dependent sampling ensures behavioral coherence: skeptical users ask probing questions, while cooperative users seek operational guidance.

Coverage & Realism. The library's 120+ patterns provide sufficient diversity to avoid repetitive interactions across thousands of training dialogues. Crucially, behaviors are grounded in real telemarketing data rather than synthetic generation, ensuring authenticity. For example, Category 7 (Communication Issues) includes realistic noise patterns like "talks to others while on call" and "repeatedly claims poor audio quality"—subtle behaviors that significantly impact dialogue flow but are often overlooked in synthetic datasets.

This behavior library serves as the foundation for our User Role-play Model's prompt construction, enabling it to generate contextually appropriate, diverse, and human-like responses that challenge agents to develop robust persuasion and adaptation strategies.