Breaking Down the Middle: User-Simulation-Based Interactive Agent Trajectory Generation

Anonymous ACL submission

Abstract

Training specialized task-oriented dialogue agents requires high-quality multi-turn interaction traces, yet acquiring real annotated dialogues is costly, privacy-constrained, and difficult to cover the long tail of real user behaviors. LLM-based user simulation offers a scalable alternative, but existing simulators exhibit systematic fidelity gaps: they produce users that are overly verbose, behaviorally homogeneous, and distributionally misaligned with real service traffic. We propose a three-level alignment framework for high-fidelity agentic trace generation that jointly models scenario coverage, behavioral composition, and linguistic realism, grounding each level in empirical distributions from real production data. Beyond one-shot data synthesis, we formulate agent training as an **iterative loop**: the simulator generates realistic multi-turn trajectories; fidelity and reward checks select training-ready traces; SFT→GRPO optimization improves the agent; and evaluation failures guide the next round of data generation. Experiments on a multilingual task-oriented dialogue benchmark demonstrate that this loop reduces distributional divergence from real user traffic and enables a specialized 9B model to approach the performance of large frontier models. These results highlight the effectiveness of enhanced-simulation loops for training task-specialized agents.

1 Introduction

General-purpose large language models (LLMs) have made agents highly capable in open-ended dialogue, where success is often measured by fluency, instruction following, and broad knowledge coverage (Ouyang et al., 2022; Zeng et al., 2024). However, these strengths do not make them directly deployable in specialized, high-volume task-completion settings (Wang et al., 2025; Gao et al., 2026). In online service handling, for example, an agent must not only understand user intent, but also ground its response in order status and service policies, choose context-appropriate wording, handle short and emotional expressions, and avoid over-promising outcomes that violate constraints. These requirements expose a deployment gap between general conversational ability and scenario-specific task competence. Simply prompting large frontier models is often impractical for high-concurrency service systems, where latency and cost constraints are stringent. We therefore focus on training specialized small models from high-quality generated data, so that domain workflows, user behavior patterns, and regional language norms can be internalized for targeted tasks.

Training such scenario-specialized agents requires high-quality multi-turn interaction trajectories. Ideally, these trajectories should come from real user-agent interactions (Gao et al., 2025a), because only such data reflects how human users express complaints, change their attitude across turns, and decide whether a resolution is acceptable. However, collecting and annotating real multi-turn service dialogues is difficult to scale: it requires human participation, strict privacy control, domain expertise, and coverage of many long-tail conditions and regional linguistic patterns (Gao et al., 2025b). Synthetic data generation offers a scalable alternative, but its usefulness depends critically on the fidelity of the agent-user interaction environment. A simulator must reproduce not only fluent utterances, but also user scenarios, behaviors, emotions, and linguistic features from turn level to session level (Gao et al., 2026). This is challenging because complaint users are often terse, fragmented, emotionally charged, region-specific, and driven by latent goals that are not explicitly expressed in dialogue states. As a result, naively prompting LLMs to play users often yields unrealistic simulations, causing agents trained on such data to overfit dis-

torted synthetic distributions.

To address these limitations, we propose a high-fidelity agentic trace simulation framework for generating quality-validated synthetic interaction data. Instead of relying on manually enumerated scenarios or uniformly sampled personas, our framework aligns user simulation with the empirical distribution of real service agent contexts at three levels: **Scenario coverage**, **Behavioral composition**, and **Linguistic patterns**.

For downstream agentic training, high-fidelity simulation cannot be guaranteed solely by generation rules. A simulator may satisfy prompt-level constraints while still producing unrealistic users or trajectories that are harmful for training (Shi et al., 2026). Therefore, we propose quality-gated evaluation metrics for data filtering and construction. The evaluation metrics focus on two complementary perspectives: *simulation fidelity evaluation*, which measures whether synthetic users match real scenario distribution, persona behavior, and linguistic statistics, and *agent-quality evaluation*, which measures whether models trained on the filtered data improve on downstream task completions, including skill alignment, state-machine compliance, and rule-based business cases. This converts synthetic data generation from a one-shot prompting process into an iterative data production loop, where simulation quality and agent performance jointly determine which data are retained for SFT and GRPO training.

We test the performance of our proposed approach on a benchmark grounded in online food-delivery service interactions, including multilingual complaints spanning seven markets. Compared with GPT-5.4, GPT-4.1, DeepSeek-V4-Flash, Qwen3.5-9B, and a Qwen3.5-9B-SFT baseline, the trained Qwen3.5-9B model matches the best tool-call validity and achieves the highest progressiveness and simulator-derived user satisfaction. Our main contributions are:

- We identify limited diversity and distributional distortion as key limitations of existing user simulation frameworks, and propose a three-level (scenario, behavioral, linguistic) alignment strategy grounded in real production data to generate realistic and diverse interaction trajectories.
- We introduce an end-to-end data skill that transforms predefined user personas and scenario definitions into training-ready interaction traces, with built-in format conversion and automatic evaluation, reducing cold-start data construction to a configurable workflow.
- We demonstrate that a specialized 9B model trained with our SFT→GRPO pipeline matches large frontier models on executable tool use and achieves the strongest progressiveness and simulator-derived satisfaction in multilingual customer-service complaint handling.

2 Related Work

2.1 User Simulation for Agentic Data

Agent data construction has increasingly relied on synthetic instructions, personas, and user simulators to reduce the cost of collecting multi-turn interaction traces. General synthetic-data pipelines such as Self-Instruct and PersonaHub scale instruction or persona diversity (Wang et al., 2023; Ge et al., 2024), while task-oriented dialogue research has long used agenda-based simulators and Wizard-of-Oz corpora to train dialogue policies (Schatzmann et al., 2007; Budzianowski et al., 2018). More recent LLM-based simulators improve flexibility through in-context role playing (Davidson et al., 2023), domain-aware tuning (Sekulic et al., 2024), user-diversity prompting (Ahmad et al., 2025), and dual-model consistency checking (Luo et al., 2024). ***Different from work that mainly optimizes dialogue plausibility***, our framework treats user simulation as fidelity-constrained agentic data generation, aligning traces with real service traffic at the scenario, behavioral, and linguistic levels before post-training.

2.2 Interactive Agent Evaluation

Although recent agent benchmarks have begun to evaluate tool calls, API use, task completion, and service-domain constraints (Yao et al., 2023; Qin et al., 2023; Yao et al., 2024; Wang et al., 2025; Shi et al., 2026), they are mainly designed to measure the final capability of a trained agent. Related dialogue benchmarks further cover multilingual interaction and customer-support quality (Hu et al., 2023, 2024; Mendonça et al., 2023), but they rarely use evaluation signals to guide how synthetic interaction data should be generated, filtered, or regenerated. ***Our evaluation is process-facing rather than only outcome-facing***: simulation-fidelity metrics filter and regenerate

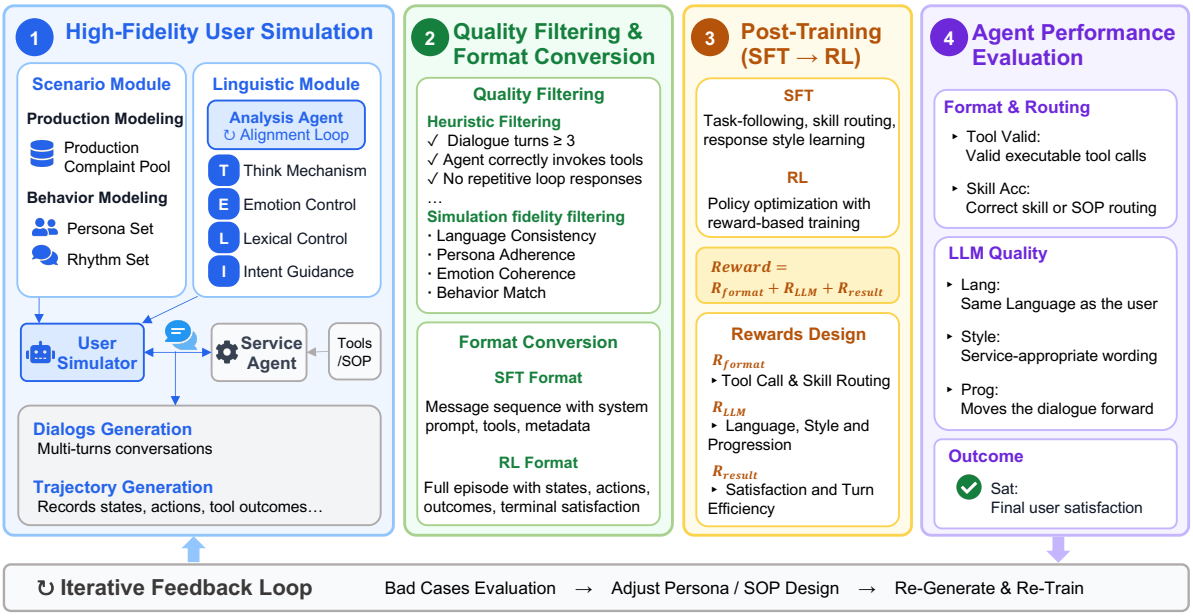

Figure 1: Overview of our end-to-end pipeline for user simulation-based agent trajectory generation. The pipeline comprises five phases: (1) role-play dialogue generation via a configurable user simulator; (2) heuristic and simulation fidelity filtering to retain high-quality trajectories; (3) training format conversion into SFT and RL (GRPO) formats; (4) model post-training via SFT→GRPO; and (5) automatic agent performance evaluation with an iterative feedback loop.

low-quality traces, agent-quality metrics diagnose tool routing, language, progress, and outcome failures, and the resulting feedback guides both subsequent data generation and SFT→GRPO optimization. In this way, evaluation improves the quality of generated trajectories and ultimately supports stronger trained-agent performance.

2.3 Multi-Turn Agent Optimization

Recent agent post-training increasingly combines SFT with RL: SFT teaches response formats, tool schemas, and basic agent behaviors (Ouyang et al., 2022; Zeng et al., 2024), while RL or preference optimization improves long-horizon interaction outcomes (Glaese et al., 2022; Rafailov et al., 2023; Shani et al., 2024; Gao et al., 2025c; Xiang et al., 2026; Shao et al., 2024). This trend has also appeared in service-agent optimization and self-evolving dialogue agents (Dai et al., 2026; Gao et al., 2026). ***However, data generation and agentic RL are inseparable***: stronger RL requires trajectories that accurately cover target scenarios, user behaviors, and language patterns. Our method therefore couples SFT→GRPO training with fidelity-gated data generation, using evaluation feedback to regenerate targeted traces before further optimization.

3 Method

As illustrated in Figure 1, our method defines an end-to-end trajectory generation and post-training pipeline that integrates high-fidelity user simulation, automatic trace transformation, model optimization, and closed-loop evaluation. Instead of relying on manually annotated multi-turn dialogues, we synthesize complaint-handling interactions from production-grounded scenarios, convert the resulting trajectories into supervised and reinforcement learning formats, and use evaluation failures to iteratively improve the training corpus. Section 3.1 introduces the user simulator and its scenario-, behavior-, and language-level alignment mechanisms; Section 3.2 describes trajectory filtering and training-format conversion; Section 3.3 presents the SFT→GRPO training procedure; Section 3.4 defines the evaluation metrics and feedback loop aligned with our reward design; and Section 3.5 describes how the complete workflow is packaged as a reusable data skill.

3.1 High-Fidelity User Simulation

Existing LLM-based user simulators tend to produce fluent, verbose, and multi-intent dialogues that diverge from real complaint messages, which are typically terse, fragmented, and goal-driven.

To address this, we propose a modular user simulation framework with three alignment levels implemented by two grounding modules: the **Scenario Module** anchors each dialogue in real production data and market-weighted behavior profiles, while the **Linguistic Module** constrains each message to match the language patterns of real users.

Scenario Module. It contains two modeling methods. In Production Modeling, each dialogue is grounded in a real production complaint order $o_i \sim \hat{p}_{\text{prod}}(\mathcal{O})$, where $\mathcal{O}$ is the de-identified complaint pool. This ensures the simulator covers long-tail combinations of market, order state, issue type, and risk constraints at their empirical frequency, rather than the simplified scenarios LLM generators tend to default to. In Behavior Modeling, a persona $p_i \sim \hat{p}_{\text{pers}}(\mathcal{P})$ and dialogue rhythm $r_i \sim \hat{p}_{\text{rhy}}(\mathcal{R})$ are sampled from market-weighted distributions estimated from real traffic, where $\mathcal{P}$ and $\mathcal{R}$ denote the persona set and rhythm set, respectively. The persona p_i controls communication style, cooperation level, and core intent, while the rhythm r_i controls interaction patterns such as burst, fragmented, passive, and persistent styles.

Linguistic Module. Even with accurate scenario and behavioral grounding, LLM-generated users exhibit systematic linguistic biases. To reduce this gap, we construct the linguistic constraint set $\mathcal{C}$ through an iterative corpus-alignment process. An analysis agent compares generated conversations against 20,441 real complaint messages, identifies distributional mismatches, and incrementally introduces targeted constraints: First, generated messages are substantially longer than real dialogues, motivating per-language length bounds paired with an internal thinking mechanism (**T**) that suppresses long messages. Second, the frequency of emotionally exaggerated acts far exceeds that of the reference corpus, motivating an emotion vocabulary restriction (**E**) that limits extreme negative terms. In addition, LLMs tend to generate stereotyped phrases, which motivates the lexical mechanism (**L**) for region-specific filler injection and filtering of high-frequency LLM-stereotyped expressions. Generated interactions habitually pack multiple intents, whereas real users overwhelmingly express a single goal per turn, motivating single-intent turn guidance (**I**). These four constraints systematically narrow the gap between simulated and real user language across several dimensions. The contribution of each mechanism is quantified in Table 1. This process is extensible, allowing new mechanisms to be added as additional markets become available.

Trajectory generation. The two modules are composed into a simulation context c_i.

$$c_i = g(o_i, p_i, r_i, \mathcal{C}) \tag{1}$$

Under c_i, the simulated user interacts with service agent π_θ to produce a trajectory d_i:

$$d_i = \{(u_t, a_t, s_t, z_t)\}_{t=1}^{T_i} \tag{2}$$

where u_t denotes the simulated user utterance, $a_t \sim \pi_\theta(\cdot \mid h_t)$ denotes the service-agent response, s_t denotes the operational state, and $z_t \in \{\text{Positive}, \text{Negative}, \text{Silent}, \text{TransToHuman}\}$ denotes the user feedback state.

3.2 Quality Filtering and Format Conversion

After role-play generation, each simulated trajectory passes through a quality filtering stage corresponding to Phase 2 in Figure 1. We first apply rule-based checks to remove traces that are too short, contain invalid or hallucinated skill calls, repeat the same response pattern, violate key SOP steps, or mismatch the assigned user language and persona. We then score the simulated user with fidelity-oriented metrics, including language consistency, persona adherence, emotion coherence, and behavior match, and retain only trajectories that satisfy both heuristic and simulation-quality thresholds.

The accepted trajectories are then converted into training-ready formats. For SFT, each dialogue is serialized as a message sequence with system instructions, user turns, agent responses, tool calls, and scenario metadata. For RL, complete episodes are preserved with round-level states, skill decisions, tool outcomes, and terminal satisfaction signals, so that rewards can be computed over the full interaction.

3.3 Post-Training Pipeline

We train the dialogue agent in two stages. In the SFT stage, the model is fine-tuned on the filtered trajectories converted in Section 3.2, learning basic task-following, skill routing, and response style. In the GRPO stage (Shao et al., 2024), the model is further optimized with multi-dimensional reward signals computed from complete interaction episodes.

The total reward combines three complementary components:

$$R = \alpha_{\text{fmt}} R_{\text{format}} + \alpha_{\text{llm}} R_{\text{LLM}} + \alpha_{\text{res}} R_{\text{result}}, \tag{3}$$

each targeting a distinct aspect of agent behavior.

$R_{\text{format}} \in [0, 1]$ evaluates whether the agent emits a valid tool or skill call and routes it to the intent-matched SOP branch; $R_{\text{LLM}} \in [0, 1]$ uses an LLM judge to assess language consistency, service-appropriate wording, and response progress; and R_{result} measures episode-level outcome quality by combining simulator-derived satisfaction with a resolution-efficiency term. Let $R_{\text{sat}} \in [0, 1]$ denote the satisfaction score at episode end. We compute the outcome reward as

$$R_{\text{result}} = R_{\text{sat}} - \lambda(1 - R_{\text{sat}}) \frac{T}{T_{\max}}, \tag{4}$$

where T is the number of dialogue turns and $T_{\max}$ is the maximum allowed turn budget, so inefficient unresolved interactions receive lower outcome reward.

3.4 Automatic Evaluation

We evaluate trained agents with metrics that mirror the three reward components while remaining interpretable at test time. **Tool-Call Validity** (*Tool Valid.*) measures the proportion of agent turns that emit a well-formed tool call or skill invocation under the required schema. **Skill-Routing Accuracy** (*Skill Acc.*) measures whether the selected skill matches the complaint intent and the expected SOP branch. Together, these two metrics evaluate the behavior targeted by R_{format}.

Language Consistency (*Lang.*) first checks whether the response follows the user's language choice across the whole dialogue, rather than switching language in isolated turns. **Style Appropriateness** (*Style*) then evaluates whether the wording is suitable for the service context, including empathy, politeness, cultural fit, and avoidance of overly templated responses. **Progressiveness** (*Prog.*) further judges whether the response advances the current complaint state, for example by collecting missing information, making a valid compensation decision, or moving the user toward closure instead of repeating generic reassurance. These criteria are evaluated with the dialogue history and the current operational state, so fluent but state-irrelevant responses are penalized. Together, these metrics provide a state-aware language-quality view and correspond to R_{LLM}.

User Satisfaction (*Sat.*) is computed from the final satisfaction signal emitted by the user simulator and corresponds to the outcome-oriented component of R_{result}.

The same evaluation also drives iterative data improvement: low-scoring cases are grouped by failure mode, such as skill misrouting, SOP violation, incorrect branch decision, language mismatch, or low satisfaction, and each group is mapped to a targeted update of scenario coverage, persona constraints, SOP design, or SFT→GRPO training data. In this way, evaluation functions as a regeneration signal rather than a terminal report, closing the loop between agent testing, data generation, and model optimization.

3.5 End-to-End Data Skill

The complete pipeline is finally abstracted as a reusable **data skill**. The goal of this abstraction is not merely to wrap data generation scripts, but to standardize the full generate–filter–train–evaluate–regenerate cycle required for agent post-training. In practice, agent-data construction usually involves fragmented prompt templates, ad hoc filtering scripts, separate training recipes, and disconnected evaluation reports. Our skill packages these components into a single configuration-driven workflow, so that the user only needs to specify the task-facing interfaces while the system executes the operational details.

Configurable interfaces. The skill exposes six key interfaces: scenario schemas, user persona distributions, tool/SOP definitions, reward functions, filtering rules, and evaluation metrics. These interfaces serve as a declarative contract between domain experts and the training pipeline. Domain experts specify what kinds of users, intents, tools, constraints, and success criteria should be simulated, while the skill standardizes how these specifications are instantiated as executable interaction traces. This design keeps the workflow portable: moving to a new task mainly requires replacing the configuration layer rather than rewriting the simulator, converter, trainer, and evaluator.

Automated execution and feedback reuse. Given the configuration, the skill automatically runs role-play generation, quality filtering, training-format conversion, SFT→GRPO optimization, agent evaluation, and targeted data re-

Table 1: Fidelity evaluation of the proposed user simulator. Linguistic Module shows incremental ablation; T=Think, E=Emotion, L=Lexical, I=Intent. green = best, red = worst per metric. Subscripts denote signed deviation from Real. ↓ = lower is better.

Dimension	Metric	Real	Scenario Module		Linguistic Module				
			w/o Production	w/o Behavioral	w/o TELI	w/o ELI	w/o LI	w/o I	Full
Linguistic Fidelity	Avg Words	4.78	$6.74_{+1.96}$	$\mathbf{5.80}_{+1.02}$	$24.38_{+19.60}$	$9.28_{+4.50}$	$10.01_{+5.23}$	$9.95_{+5.17}$	$6.87_{+2.09}$
	P90 Words	10.00	$11.00_{+1.00}$	$9.00_{-1.00}$	$38.00_{+28.00}$	$15.00_{+5.00}$	$15.00_{+5.00}$	$15.00_{+5.00}$	$\mathbf{10.00}_{0.00}$
	Avg Clause Len	4.49	$4.14_{-0.35}$	$4.05_{-0.44}$	$8.27_{+3.78}$	$4.88_{+0.39}$	$5.15_{+0.66}$	$5.24_{+0.75}$	$\mathbf{4.34}_{-0.15}$
	FTA Penetration (%)	1.30	$5.76_{+4.46}$	$10.32_{+9.02}$	$21.40_{+20.10}$	$8.49_{+7.19}$	$6.24_{+4.94}$	$6.81_{+5.51}$	$\mathbf{5.52}_{+4.22}$
	Mixed Speech Act (%)	1.54	$15.15_{+13.61}$	$14.14_{+12.60}$	$44.45_{+42.91}$	$22.71_{+21.17}$	$21.01_{+19.47}$	$17.53_{+15.99}$	$\mathbf{11.81}_{+10.27}$
	Bigram Entropy (bits)	14.05	$11.13_{-2.92}$	$10.05_{-4.00}$	$12.21_{-1.84}$	$12.45_{-1.60}$	$11.71_{-2.34}$	$\mathbf{13.01}_{-1.04}$	$12.47_{-1.58}$
Semantic Distribution	SDR (×10) ↓	–	8.13	8.57	8.21	8.07	8.14	8.05	**8.01**
	SCE_{KL} ↓	–	28.92	3.58	4.19	3.57	6.21	3.86	**3.56**

generation. It also preserves intermediate artifacts, including rejected traces, reward breakdowns, failed tool calls, low-satisfaction sessions, and evaluation failure clusters. These artifacts make the loop actionable: skill-routing failures can trigger additional SOP-branch traces, language mismatches can update linguistic constraints, and low-satisfaction sessions can guide persona or reward revisions. Thus, evaluation is not treated as a terminal report; it becomes a regeneration signal that continuously improves both the synthetic data distribution and the trained agent.

4 Experiments

4.1 Implementation Details

We implement our framework on a multilingual food-delivery complaint-handling setting covering seven markets. Qwen3.5-9B is used as the open-source backbone. Training data are generated by the user simulator in Section 3.1, filtered and converted following Section 3.2, and then used for two-stage post-training: SFT first teaches executable tool use, skill routing, and response style, while GRPO further optimizes the reward in Eq. 3. For fair comparison, all large frontier models and open-source baselines are evaluated with the same agent instruction, tool/SOP specification, user simulator, and turn-level evaluation protocol. We report the reward-aligned metrics defined in Section 3.4, covering executable tool use and skill routing (*Tool Valid.*, *Skill Acc.*), conversational quality (*Lang.*, *Style*, *Prog.*), and final task outcome (*Sat.*).

4.2 Simulation Fidelity Evaluation

We start by evaluating the simulation performance of the proposed user simulator, focusing on whether generated users match real-user distributions across scenario and linguistic dimensions. Table 1 reports fidelity metrics for the complete simulator configuration (**Full**) and several diagnostic variants.

We compare **Full** with controlled variants that isolate the contribution of each simulator component. For the Scenario Module, **w/o Production** replaces real orders with LLM-generated orders, and **w/o Behavior** removes persona settings p_i and rhythm sampling r_i. The Linguistic Module is evaluated through incremental variants (**w/o TELI** → **w/o ELI** → **w/o LI** → **w/o I** → **Full**), where each step re-introduces one additional constraint discussed in Section 3.1.

Evaluation metrics. We evaluate against a real-user reference corpus (**Real**) including 20,441 dialogues from 7 markets across 4 languages using eight metrics.

Scenario distribution metrics measure scenario fidelity. *SDR* (Semantic Distance from Real) is the distance between simulated and real dialogues:

$$\text{SDR} = 1 - \frac{1}{NM} \sum_{i,j} \cos\left(\mathbf{e}_i^{\text{sim}}, \mathbf{e}_j^{\text{real}}\right) \quad (5)$$

where $\mathbf{e}_i^{\text{sim}}$ and $\mathbf{e}_j^{\text{real}}$ denote the sentence embeddings of dialogues, and N, M are the corpus sizes. Lower SDR indicates closer semantic proximity to real users. SCE_{KL} measures the distribution of market-complaint joint patterns between simulated and actual data, defined as:

$$\text{SCE}_{\text{KL}} = \sum_{(r,f)} p_{\text{real}}(r,f) \log \frac{p_{\text{real}}(r,f)}{\hat{p}(r,f)} \quad (6)$$

where r denotes market and f denotes complaint type. Lower SCE_{KL} indicates better distribution alignment with real data.

Linguistic fidelity metrics measure alignment with language patterns of real users. Avg Words and P90 Words represent the average sentence word count and 90th percentile of sentence word

Table 2: Distribution of generated dialogues by region, complaint type, and dialogue statistics.

Category	Value	Count	%
Region	Saudi Arabia (SA)	4,737	47.4
	UAE (AE)	1,509	15.1
	Kuwait (KW)	1,461	14.6
	Brazil (BR)	1,394	13.9
	Qatar (QA)	403	4.0
	Bahrain (BH)	284	2.8
	Hong Kong (HK)	206	2.1
Complaint Type	General inquiry (no category)	2,977	29.8
	Other clear intent	2,887	28.9
	Missing items	1,669	16.7
	Wrong items	849	8.5
	Food spillage / damage	371	3.7
	Late delivery	336	3.4
	Wrong order	171	1.7
	Other seven types	734	7.3
Dialogue Stats	Turns per session	3.81	-
	Tokens per turn	17.90	-
	Words per turn	6.87	-

count. Avg Clause Len is the mean word count per clause after splitting on punctuation, reflecting syntactic fragmentation. FTA Penetration is the fraction of messages containing face-threatening acts such as accusations and threats. Mixed Speech Act denotes the proportion of messages containing multiple act types in a single turn. Bigram Entropy is the Shannon entropy for lexical diversity.

Analysis of Scenario Module Ablation. Removing Production Modeling (w/o Production) causes a sharp increase in SCE_{KL} from 3.56 to 28.92, confirming that LLM-generated orders fail to reproduce the long-tail distribution of real data. However, it has negligible impact on linguistic metrics. Removing the persona and rhythm system (w/o Behavior) yields the worst SDR, indicating that without Behavioral Modeling, the simulated interaction dialogues are semantically furthest from real users, and also produces the lowest Bigram Entropy, reflecting the collapse to uniform, formulaic language when user diversity is removed.

Analysis of Linguistic Module Ablation. Reintroducing think mechanism **T** substantially reduces message length, with Avg Words dropping from 24.38 to 9.28 and P90 Words from 38 to 15. The Emotion control mechanism **E** suppresses face-threatening language, reducing FTA Penetration from 8.49% to 6.24%. Lexical mechanism **L** improves lexical diversity, raising Bigram Entropy to its peak. Intent guidance **I** resolves speech-act mixing, reducing Mixed Speech Act from 17.53% to 11.81%, reducing P90 Words to the real value (10.0), and moving Avg Words closer to Real (from 9.95 to 6.87). This suggests that intent mixing may be a direct contributor to long message length.

Review of generated dialogues. Table 2 summarizes the distribution of the generated dialogues. The corpus contains 9994 sessions spanning 7 markets and 4 languages. By region, Saudi Arabia accounts for the largest share (47.4%), followed by UAE (15.1%) and Kuwait (14.6%), reflecting the market distribution of the deployment environment. Complaint types are dominated by general inquiries (29.8%) and other clear-intent complaints (28.9%), with item-related complaints (missing items 16.7%, wrong items 8.5%) forming the next largest group. On average, each session consists of 3.81 turns, with 17.90 tokens and 6.87 words per turn, confirming the terse, fragmented nature of real complaint messages.

4.3 Agent Performance Evaluation

Table 3 presents the overall performance comparison on our internal multilingual food-delivery complaint benchmark. On executable format, our model reaches 100.0 tool-call validity, matching GPT-5.4 and Qwen3.5-9B-SFT and substantially improving over the base Qwen3.5-9B model (52.5). This indicates that the filtered traces are effective for teaching the model to produce valid tool or skill invocations. For skill routing, our model obtains 93.3, slightly below GPT-5.4 (97.0) but clearly above Qwen3.5-9B-SFT (81.8), GPT-4.1 (67.5), and DeepSeek-V4-Flash (44.1), showing that GRPO further improves decision quality beyond format learning.

The language-quality metrics reveal a more nuanced pattern. GPT-5.4 remains strongest on style appropriateness (92.4), and Qwen3.5-9B-SFT achieves the highest language consistency (100.0). However, strong local language scores do not necessarily translate into better task progress: DeepSeek-V4-Flash reaches 99.8 on language consistency but only 6.2 on progressiveness and 3.3 on final satisfaction. In contrast, our model achieves the best progressiveness (80.6) and final satisfaction (71.3), outperforming GPT-5.4 on the two outcome-oriented dimensions despite using a much smaller backbone. This separation between format accuracy and outcome quality supports the role of the SFT→GRPO design: SFT stabilizes executable behavior, while GRPO uses interaction-level feedback to refine decisions that

Table 3: Internal reward-aligned evaluation on the multilingual food-delivery complaint-handling benchmark. Metrics mirror the training rewards, covering format/routing correctness, LLM-judged response quality, and simulator-derived outcomes. Best results are **bolded**; second best are underlined. ↑: higher is better; ↓: lower is better.

Group	Method	Format & Routing		LLM Quality			Outcome
		Tool Valid.↑	Skill Acc.↑	Lang.↑	Style↑	Prog.↑	Sat.↑
Large Models	GPT-5.4	**100.0**	**97.0**	99.6	**92.4**	73.1	66.9
	GPT-4.1	53.6	67.5	96.5	66.1	40.0	30.7
	DeepSeek-V4-Flash	9.7	44.1	99.8	32.8	6.2	3.3
Foundation Models	Qwen3.5-9B	52.5	70.4	96.7	60.3	31.6	21.2
	Qwen3.5-9B-SFT	**100.0**	81.8	**100.0**	80.7	62.5	55.2
Ours	SFT → GRPO	**100.0**	93.3	97.6	87.5	**80.6**	**71.3**

Table 4: Internal business-oriented dynamic evaluation on deployment-facing service metrics. The metrics correspond to the operational indicators discussed in Section 4.4. All methods are evaluated with the same user simulator. Best results are **bolded**; second best are underlined.

Group	Method	Comp. Sat.↑	Comp. Xfer.↓	No-Comp. Sat.↑	No-Comp. Xfer.↓	Comp. Rate↓
Large Models	GPT-5.4	54.7	**5.4**	13.8	**55.2**	20.0
	GPT-4.1	52.8	38.1	8.7	72.2	25.2
	DeepSeek-V4-Flash	46.3	50.6	7.7	83.4	25.9
Foundation Models	Qwen3.5-9B	37.7	25.5	4.8	90.2	20.6
	Qwen3.5-9B-SFT	22.2	61.8	8.9	78.9	36.9
Ours	SFT → GRPO	**76.0**	12.0	**29.5**	58.5	**8.1**

affect task progress and user satisfaction. These results suggest that high-fidelity interaction traces and reward-aligned optimization mainly improve the agent's ability to make task progress and reach satisfactory resolutions, rather than merely imitating response format.

4.4 Business-Oriented Dynamic Evaluation

Beyond the reward-aligned metrics, we further evaluate business-critical session outcomes that are directly monitored in deployment: satisfaction and transfer-to-human rates under compensated (Comp. Sat. and Comp. Xfer.) and non-compensated cases (No-Comp. Sat. and No-Comp. Xfer.), together with the overall compensation rate (Comp. Rate). These metrics test whether the skill-based data generation and SFT→GRPO training process improves not only model-facing behavior but also operational decision quality.

Table 4 shows strong gains across deployment-facing indicators. In compensated sessions, our model achieves the highest satisfaction score (76.0) and the second-lowest transfer-to-human rate (12.0), behind only GPT-5.4. In non-compensated sessions, it also obtains the best satisfaction score (29.5) and the second-lowest transfer rate (58.5), while maintaining the lowest overall compensation rate (8.1). One possible explanation for GPT-5.4's especially low transfer rates is that a large frontier model may be better at sustaining a conversation and avoiding escalation under ambiguous user feedback. However, its lower satisfaction scores and higher compensation rate suggest that avoiding transfer alone may not imply better domain-specific resolution quality, whereas our model appears to balance retention, compensation control, and user satisfaction more effectively.

5 Conclusion

We present a high-fidelity agentic trace simulation and post-training framework for building specialized dialogue agents. The framework aligns synthetic interactions with real service traffic at the scenario, behavioral, and linguistic levels, filters generated trajectories by simulation quality, and converts them into SFT and GRPO training data. We further package this process as a reusable data skill that exposes key configuration interfaces for personas, scenarios, rewards, and evaluation metrics. Experiments on multilingual food-delivery complaint handling show that the resulting traces are distributionally realistic and that the trained agent improves both reward-aligned metrics and business-oriented session outcomes.

Limitations

Our framework requires task-specific configurations, including scenario schemas, persona definitions, tool/SOP descriptions, reward functions, and evaluation metrics. This introduces some domain adaptation effort when moving to a new application, but it also makes the data generation and optimization process controllable, auditable, and aligned with deployment goals. In addition, although simulator-derived satisfaction and LLM-judged quality metrics provide scalable feedback for data filtering and model optimization, they cannot fully replace live user studies or deployment-side A/B testing. Future work can combine our offline evaluation loop with online feedback to further calibrate rewards and user simulators.

References

Adnan Ahmad, Stefan Hillmann, and Sebastian Möller. 2025. Simulating user diversity in task-oriented dialogue systems using large language models. *arXiv preprint arXiv:2502.12813*.

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. 2018. MultiWOZ - a large-scale multi-domain Wizard-of-Oz dataset for task-oriented dialogue modelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5016–5026, Brussels, Belgium. Association for Computational Linguistics.

Yuqin Dai, Ning Gao, Wei Zhang, Jie Wang, Zichen Luo, Jinpeng Wang, Yujie Wang, Ruiyuan Wu, and Chaozheng Wang. 2026. Sead: Self-evolving agent for multi-turn service dialogue. *arXiv preprint arXiv:2602.03548*.

Sam Davidson, Salvatore Romeo, Raphael Shu, James Gung, Arshit Gupta, Saab Mansour, and Yi Zhang. 2023. User simulation with large language models for evaluating task-oriented dialogue. *arXiv preprint arXiv:2309.13233*.

Ning Gao, Wei Zhang, Yuqin Dai, Ling Shi, Ziyin Wang, Yujie Wang, Wei He, Jinpeng Wang, and Chaozheng Wang. 2026. Reinforcing real-world service agents: Balancing utility and cost in task-oriented dialogue. *arXiv preprint arXiv:2602.22697*.

Rena Gao, Carsten Roever, and Jey Han Lau. 2025a. Interaction matters: An evaluation framework for interactive dialogue assessment on english second language conversations. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 10977–11012.

Rena Wei Gao, Xuetong Wu, Tatsuki Kuribayashi, Mingrui Ye, Siya Qi, Carsten Roever, Yuanxing Liu, Zheng Yuan, and Jey Han Lau. 2025b. Can llms simulate l2-english dialogue? an information-theoretic analysis of l1-dependent biases. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4355–4379.

Zhaolin Gao, Wenhao Zhan, Jonathan Chang, Gokul Swamy, Kianté Brantley, Jason Lee, and Wen Sun. 2025c. Regressing the relative future: Efficient policy optimization for multi-turn rlhf. In *International Conference on Learning Representations*, volume 2025, pages 8475–8514.

Tao Ge, Xin Chan, Xiaoyang Wang, Dian Yu, Haitao Mi, and Dong Yu. 2024. Scaling synthetic data creation with 1,000,000,000 personas. *arXiv preprint arXiv:2406.20094*.

Amelia Glaese, Nat McAleese, Maja Trębacz, John Aslanides, Vlad Firoiu, Timo Ewalds, Maribeth Rauh, Laura Weidinger, Martin Chadwick, Phoebe Thacker, and 1 others. 2022. Improving alignment of dialogue agents via targeted human judgements. *arXiv preprint arXiv:2209.14375*.

Songbo Hu, Xiaobin Wang, Zhangdie Yuan, Mete Hergul, Han Zhou, Ivan Vulić, and Anna Korhonen. 2024. DIALIGHT: Lightweight multilingual development and evaluation of task-oriented dialogue systems. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics*.

Songbo Hu, Han Zhou, Mete Hergul, Milan Gritta, Guchun Zhang, Ignacio Iacobacci, Ivan Vulić, and Anna Korhonen. 2023. Multi 3 WOZ: A multilingual, multi-domain, multi-parallel dataset for training and evaluating culturally adapted task-oriented dialog systems. *Transactions of the Association for Computational Linguistics*, 11:1396–1415.

Xiang Luo, Zhiwen Tang, Jin Wang, and Xuejie Zhang. 2024. Duetsim: Building user simulator with dual large language models for task-oriented dialogues. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 5414–5424.

John Mendonça, Patrícia Pereira, Miguel Menezes, Vera Cabarrão, Ana C Farinha, Helena Moniz, Alon Lavie, and Isabel Trancoso. 2023. Dialogue quality and emotion annotations for customer support conversations. In *Proceedings of the Third Workshop on Natural Language Generation, Evaluation, and Metrics (GEM)*, pages 9–21.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, and 1 others. 2023. ToolLLM: Facilitating large language models to master 16000+ real-world APIs. *arXiv preprint arXiv:2307.16789*.

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. *Advances in neural information processing systems*, 36:53728–53741.

Jost Schatzmann, Blaise Thomson, Karl Weilhammer, Hui Ye, and Steve Young. 2007. Agenda-based user simulation for bootstrapping a POMDP dialogue system. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Companion Volume, Short Papers*, pages 149–152, Rochester, New York. Association for Computational Linguistics.

Ivan Sekulic, 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. In *Proceedings of the 1st Workshop on Simulating Conversational Intelligence in Chat (SCI-CHAT 2024)*, pages 19–35, St. Julians, Malta. Association for Computational Linguistics.

Lior Shani, Aviv Rosenberg, Asaf Cassel, Oran Lang, Daniele Calandriello, Avital Zipori, Hila Noga, Orgad Keller, Bilal Piot, Idan Szpektor, and 1 others. 2024. Multi-turn reinforcement learning from preference human feedback. *arXiv preprint arXiv:2405.14655*.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Ling Shi, Yuqin Dai, Ziyin Wang, Ning Gao, Wei Zhang, Chaozheng Wang, Yujie Wang, Wei He, Jinpeng Wang, and Deiyi Xiong. 2026. Sage: A service agent graph-guided evaluation benchmark. *arXiv preprint arXiv:2604.09285*.

Haoxin Wang, Xianhan Peng, Huang Cheng, Yizhe Huang, Ming Gong, Chenghan Yang, Yang Liu, and Jiang Lin. 2025. 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.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13484–13508, Toronto, Canada. Association for Computational Linguistics.

Xingyuan Xiang, Xiangchen Pan, and Wei Wei. 2026. User simulator-guided multi-turn preference optimization. *arXiv preprint arXiv:2604.03671*.

Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. 2024. τ-bench: A benchmark for tool-agent-user interaction in real-world domains. *arXiv preprint arXiv:2406.12045*.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations*.

Aohan Zeng, Mingdao Liu, Rui Lu, Bowen Wang, Xiao Liu, Yuxiao Dong, and Jie Tang. 2024. AgentTuning: Enabling generalized agent abilities for LLMs. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 3053–3077, Bangkok, Thailand. Association for Computational Linguistics.