

# 000 $\tau^2$ -BENCH: EVALUATING CONVERSATIONAL AGENTS 001 IN A DUAL-CONTROL ENVIRONMENT 002 003 004

005 **Anonymous authors**

006 Paper under double-blind review  
007  
008  
009

## ABSTRACT

012 Existing benchmarks for conversational AI agents simulate *single-control* environments,  
013 where only the AI agent can use tools to interact with the world, while the  
014 user remains a passive information provider. This differs from real-world scenarios  
015 like technical support, where users need to actively participate in modifying the  
016 state of the (shared) world. In order to address this gap, we introduce  $\tau^2$ -bench,  
017 with four key contributions: 1) A novel Telecom dual-control domain modeled as  
018 a Dec-POMDP, where both agent and user make use of tools to act in a shared,  
019 dynamic environment that tests both agent coordination and communication, 2)  
020 A compositional task generator that programmatically creates diverse, verifiable  
021 tasks from atomic components, ensuring domain coverage and controlled com-  
022 plexity, 3) A reliable user simulator tightly coupled with the environment, whose  
023 behavior is constrained by tools and observable states, improving simulation fi-  
024 delity, 4) fine-grained analysis of agent performance through multiple ablations  
025 including separating errors arising from reasoning vs communication/coordination.  
026 In particular, our experiments show significant performance drops when agents  
027 shift from no-user to dual-control, highlighting the challenges of guiding users.  
028 Overall,  $\tau^2$ -bench provides a controlled testbed for agents that must both reason  
029 effectively and guide user actions.  
030

## 1 INTRODUCTION

031 Existing benchmarks for conversational AI agents are designed to test their abilities to communicate  
032 effectively with a user and perform the right sequence of actions to solve tasks (Yao et al., 2024; Lu  
033 et al., 2024; Xiao et al., 2024; Prabhakar et al., 2025). These benchmarks are inherently *single-control*  
034 environments, where the AI agent is able to interact with the world but the (simulated) user is limited  
035 to providing information about preferences and goals. In  $\tau$ -bench (Yao et al., 2024) for example, the  
036 *retail* and *airline* domains test the agent’s ability to solve constraint satisfaction tasks, where  
037 constraints stem from a combination of a domain policy that the agent must follow and the user’s  
038 cognitive state including beliefs, goals and preferences.  
039

040 In such settings, the user’s understanding of the agent’s environment (including the set of actions  
041 the agent can take) is provided through carefully crafted natural language instructions that help  
042 ensure a single, solvable path for the constraint satisfaction problem. While this enables easier task  
043 specification, the user can only perceive the agent’s actions through communication and reason about  
044 the environment state solely based on the initial instructions. This is quite different from real-world  
045 scenarios like technical customer support, where the user has to actively participate in taking some  
046 actions to diagnose and solve problems when asked to do so, such as *restarting their phone* or *turning*  
047 *off airplane mode*.  
048

049 In order to capture this additional real-world complexity, we introduce  $\tau^2$ -bench with a *dual-control*  
050 environment, where (LLM-simulated) users can take actions and call tools in addition to commu-  
051 nicipating with the agent. The dual-control model provides a few advantages in improving the user  
052 simulation, such as selective information hiding, non-verbal manipulation of the environment, and  
053 easier specification of the user’s task or scenario, especially in cases where the structured format of  
tools is preferred to unstructured natural language.

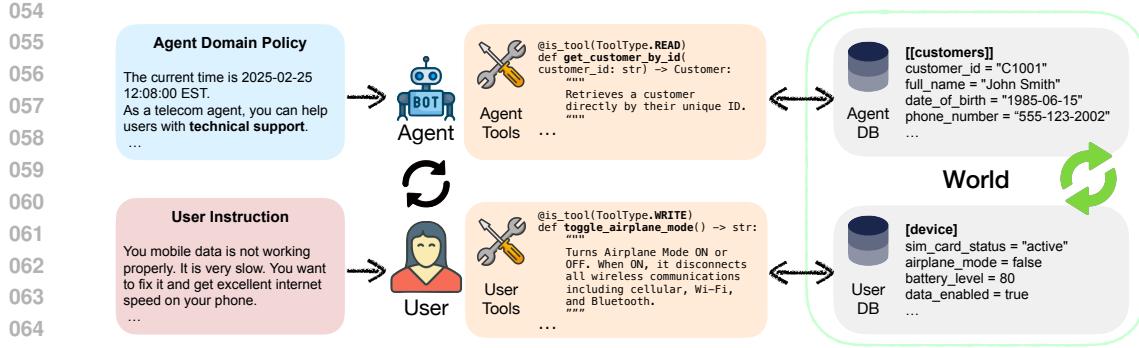


Figure 1: Supporting dual-control environment in  $\tau^2$ -bench telecom. The agent has access to a set of tools that interact with a database, and is tasked with resolving the user’s request via Tool-Agent-User (TAU) interactions while adhering to the domain policy. To test real-world scenarios, the user is simulated by another AI agent given a scenario-based instruction and a set of tools that interact with its own database. The simulated user can be regarded as handling an easier version of the TAU interaction in a dual format (Tool-User-Agent), where it only need to follow instructions but does not need to reason about solutions for the task.

The primary challenge in a dual-control setup lies in granting the user simulator meaningful agency via tools while maintaining the “complexity asymmetry” between the agent and the user simulator – that is, making sure the user’s affordances are limited and the user still requires support from the agent in order to solve issues. While tools for the user simulator could disrupt this balance by increasing its capabilities, we find that carefully designing the tools can help constrain user behavior, leading to more reliable simulations. We achieve this by ensuring user tools yield only human-readable outputs, limiting user planning to reactive tool use based on agent requests, and tightly constraining user behavior through the environment. This results in a controlled testbed for agents that must both think and effectively guide a user to take the right actions.

Overall,  $\tau^2$ -bench makes four key contributions:

**Dual-control environment.** We introduce a novel dual-control setup and instantiate `telecom` domain within  $\tau^2$ -bench, where, unlike previous benchmarks, both the AI agent and the user possess distinct tools to observe, act upon, and verify the state of a shared, dynamic environment. This is formalized using a Decentralized Partially Observable Markov Decision Process (Dec-POMDP) (Oliehoek et al., 2016). This dual-control setup is designed to accurately represent real-world collaborative scenarios and exposes crucial agent coordination and communication challenges absent from evaluations where users have limited agency. Experiments show state-of-the-art LLMs struggle significantly in this domain (e.g., pass<sup>1</sup> of 34% for `gpt-4.1` (OpenAI, 2025a), 42% for `o4-mini` (OpenAI, 2025b), and 49% for `claude-3.7-sonnet` (Anthropic, 2025) on new tasks).

**Compositional task generator.**  $\tau^2$ -bench incorporates a programmatic task generator that automatically composes a vast and diverse set of verifiable tasks from a small set of atomic base scenarios (defined by initialization, solution, and assertion functions). This method ensures provable correctness of tasks, provides complete domain coverage, allows for explicit control over task complexity (e.g., by number of solution steps or issue type), and removes the manual effort and potential brittleness associated with hand-crafted task suites.

**Reliable user simulator.** We enhance the reliability of the user simulation by tightly coupling the user simulator to the environment. User behavior is constrained by the available tools and the observable state of the environment, leading to more predictable and consistent interactions. This approach significantly alleviates the need for complex natural language prompting to guide the user simulator and results in substantially higher reliability (e.g., the `telecom` domain’s user simulator shows a 16% error rate with 6% critical errors, compared to 40% error rate with 12% critical errors in the `retail` domain from  $\tau$ -bench).

**Fine-grained diagnosis of agent failures.**  $\tau^2$ -bench enables a decomposed diagnosis of agent performance by evaluating task success in different modes: (i) a fully autonomous mode (“no-user mode”), where the agent controls all tools, isolating its reasoning capabilities, and (ii) the standard dual-control mode, which introduces communication and coordination requirements. Our findings reveal a substantial performance decrease (around 20% pass<sup>1</sup>) when agents must shift from

108 autonomous operation to guiding a user. This clearly distinguishes pure reasoning failures from those  
 109 arising from communication and decentralized control, pinpointing the latter as a critical bottleneck.  
 110

## 111 2 RELATED WORK

112 **Benchmarks for Conversational AI Agents.** Following a long line of research into language  
 113 agents (Yao et al., 2022; Zhou et al., 2023; Jimenez et al., 2023; Liu et al., 2023; Ruan et al.,  
 114 2023), LLM tool use (Yan et al., 2024; Qin et al., 2024; Huang et al., 2023), and task-oriented  
 115 dialog (Chen et al., 2021; Budzianowski et al., 2018; Andreas et al., 2020; Schatzmann et al., 2007;  
 116 Gür et al., 2018; He et al., 2018; Hu et al., 2023),  $\tau$ -bench (Yao et al., 2024) is a recently introduced  
 117 benchmark to measure the reliability of language agents in multi-turn task-oriented conversations  
 118 such as customer service workflows, while respecting domain rules. Each task in  $\tau$ -bench instantiates  
 119 a live conversation between a user simulator and the language agent, with tasks spread across two  
 120 domains – retail and airline. To quantify reliability the paper introduces  $\text{pass}^k$  metrics:  
 121 the fraction of  $k$  independent runs that succeed.

122 Several follow-ups to  $\tau$ -bench have explored variations of the basic setup. FlowBench (Xiao et al.,  
 123 2024) isolates the planning step of tool-using agents by injecting explicit workflow knowledge into  
 124 the prompt, using natural language, python-like pseudocode or mermaid flowcharts. IntellAgent (Levi  
 125 & Kadar, 2025) provides an evaluation pipeline to programmatically build synthetic test suites from  
 126 structured policy graphs that encode domain rules and their co-occurrence statistics. IntellAgent  
 127 explicitly uses  $\tau$ -bench as an external gold standard, reporting a high Spearman correlation between  
 128 the two score distributions, and acts as a fast, synthetic proxy task. APIGen-MT (Prabhakar et al.,  
 129 2025) explores the idea of fine-tuning tool-calling agents for  $\tau$ -bench. They generate data by creating  
 130 conversation blueprints which are sequences of tool calls that depend on each other, followed by  
 131 simulating conversational traces based on each blueprint. ToolSandbox (Lu et al., 2024) focuses on  
 132 creating stateful tools in order to evaluate agent progress in a more fine-grained manner.

133 Our work extends the  $\tau$ -bench paradigm and generalizes it to allow for both the user and agent to  
 134 have state-changing abilities (via tool calls) over a shared world. As demonstrated in the results,  
 135 this allows us to build more complex domains to test conversational agents, while also providing the  
 136 opportunity for fine-grained analysis of agent failure points that can be improved upon.

137 **User Simulation for Conversational Agents.** The reliability of user simulation has been a key  
 138 concern for benchmarks like  $\tau$ -bench (Yao et al., 2024). While most efforts have focused on  
 139 introducing supervision for the user simulator, for instance, by using a generic LLM to generate or  
 140 validate user responses (Prabhakar et al., 2025), less attention has been paid to the possibility of  
 141 using the environment itself to constrain and shape user simulator behavior for increased reliability, a  
 142 core tenet of our approach. This concern has been extensively studied in the context of task-oriented  
 143 dialogue systems, with early work by (Pietquin & Hastie, 2013) providing a comprehensive survey  
 144 of metrics for evaluating user simulations. More recently, (Kazi et al., 2024) has demonstrated how  
 145 LLMs can be effectively used as user-agents for evaluating task-oriented dialogue systems, showing  
 146 that careful prompting and state tracking can lead to more reliable and context-aware user simulations.

147 **Multi-Agent Benchmarks.** Our work is also related to efforts to build multi-agent frameworks  
 148 and evaluate them (Zhu et al., 2025). While we can consider the user and the agent in our paper as  
 149 forming a multi-agent system, the key difference in our case is that the final evaluation still focuses  
 150 on the agent’s ability to elicit the right information from the user and perform the correct actions to  
 151 solve the task. This introduces an inherent asymmetry between the agent and the user; our focus is  
 152 not on solving a pure multi-agent problem but rather on the agent’s capability to effectively guide and  
 153 collaborate with a user who also possesses agency. In this sense, the framework can be collaborative  
 154 (e.g., troubleshooting), competitive (e.g., negotiating a subscription), or a hybrid, requiring the agent  
 155 to identify and navigate the scenario appropriately, even accounting for user mistakes or errors.

## 156 3 $\tau^2$ -BENCH: EVALUATING AGENTS IN A DUAL-CONTROL ENVIRONMENT

157  $\tau^2$ -bench serves as a platform for a systematic study of multi-turn interactions between a conversa-  
 158 tional AI agent and a simulated user. Dual-control interactions are formulated as a Decentralized

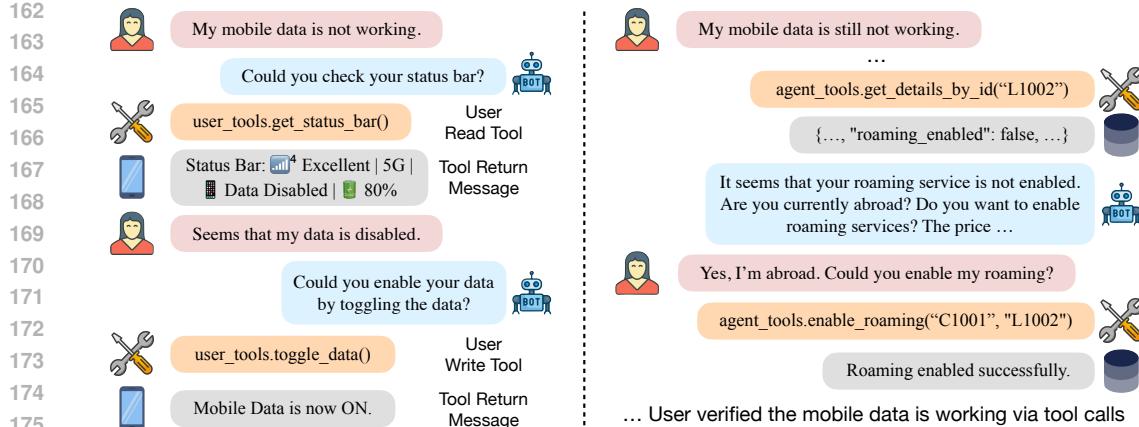


Figure 2: An example agent-user interaction trajectory ( $S_{history}$ ) of  $\tau^2$ -bench in the `telecom` domain. By controlling the implementation of the user tools (the mocked phone), we can reliably simulate the user’s response to agent’s actionable instructions like “checking the status bar” and “toggling data” based on the underlying database state. On the right half, we show the possibility of modeling the impact of agent’s tool calls on the user’s database state, where the roaming service for the user is enabled on the agent’s end and therefore allows the user’s phone to roam.

Partially Observable Markov Decision Process (Dec-POMDP) (Oliehoek et al., 2016) to define the evaluation environment, where both the agent and user can communicate, use tools, and receive observations. This allows us to simulate complex scenarios like technical troubleshooting where agent and user must coordinate their actions to solve the task.

### 3.1 THE DEC-POMDP FORMALISM

As illustrated in Figure 1, the Dec-POMDP in  $\tau^2$ -bench involves two players: an agent and a user. The entire process is formally defined by the tuple  $(\mathcal{S}, \{\mathcal{A}_i\}, \{\mathcal{O}_i\}, \mathcal{T}, \mathcal{R}, \mathcal{U}, \mathcal{M})$ , where  $i \in \{\text{agent}, \text{user}\}$  denotes players and each component in the tuple is detailed below, with illustrative examples drawn from the new `telecom` domain.

**Message space ( $\mathcal{M}$ ):** The set of all possible (natural language) messages exchanged between the agent and the user. For example, the user could say “I cannot use mobile data.” and the agent could respond with “Could you check whether your airplane mode is on?”

**State space ( $\mathcal{S}$ ):** The global state  $\mathcal{S} = \mathcal{S}_{world} \otimes \mathcal{S}_{history}$ , where  $\mathcal{S}_{world} = \mathcal{S}_{db,agent} \otimes \mathcal{S}_{db,user}$  represents the *underlying* database states for the agent and user, and  $\mathcal{S}_{history}$  logs all interaction events (actions, observations, messages). For example, in the `telecom` domain,  $\mathcal{S}_{db,agent}$  might be CRM data (customer profiles, lines), while  $\mathcal{S}_{db,user}$  could be phone status.

**Action spaces ( $\mathcal{A}_i$ ):** Player  $i$ ’s action  $a_i \in \mathcal{A}_i$  is either a tool call  $a_{i,tool} \in \mathcal{A}_{i,tool}$  (interacting with  $\mathcal{S}_{db,i}$  via function calls like `tool_name(**kwargs)`) or a message  $m_i \in \mathcal{M}$ . Only one player acts per turn. In the `telecom` domain, the agent can access tools like `get_customer_by_id(id)` and the user can access tools like `toggle_airplane_mode()`.

**Observation spaces ( $\mathcal{O}_i$ ):** Player  $i$ ’s observation  $o_i \in \mathcal{O}_i$  is either a tool observation  $o_{i,tool}$  (e.g., data, messages, or errors from  $a_{i,tool}$ ) or a message  $m_j \in \mathcal{M}$  from player  $j \neq i$ . Only one player receives an observation per turn. In the `telecom` domain, the agent might observe customer details from `get_customer_by_id`, and the user might observe a message indicating the airplane mode has been turned off from `toggle_airplane_mode`.

**Transition function ( $\mathcal{T}$ ):** Defines system dynamics via  $\mathcal{T} : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S} \times \mathcal{O}$ . Given current state  $s \in \mathcal{S}$  and joint action  $a = (a_{agent}, a_{user})$ , it yields a new state  $s' \in \mathcal{S}$  and joint observation  $o = (o_{agent}, o_{user})$ . Calling a tool  $a_{i,tool} \in \mathcal{A}_{i,tool}$  may change  $\mathcal{S}_{world}$  and yield  $o_i \in \mathcal{O}_{i,tool}$ . Sending a message  $m_i \in \mathcal{M}$  yields  $o_j = m_i$  for  $j \neq i$ . In both cases,  $s'$  includes updated  $\mathcal{S}_{world}$  and  $\mathcal{S}_{history}$ . For example, an agent’s action `enable_roaming(customer_id, line_id)` would update the world state (the roaming service for the specific line number is enabled), and a user’s action `toggle_airplane_mode` would update the status of the mocked phone.

216  
217  
218 Table 1: Key statistics for the  $\tau^2$ -bench domains.  
219  
220  
221

|                        | <b>retail</b>                        | <b>airline</b>                             | <b>telecom</b>                |
|------------------------|--------------------------------------|--|-------------------------------|
| <b>Agent Databases</b> | 500 users, 50 products, 1,000 orders | 500 users, 300 flights, 2,000 reservations | 5 plans, 9 lines, 4 customers |
| <b>Agent Tools</b>     | 7 write, 6 read                      | 6 write, 6 read                            | 6 write, 7 read               |
| <b>User Tools</b>      | -                                    | -  | 15 write, 15 read             |
| <b>Tasks</b>           | 115                                  | 50   | 114 (full: 2285)              |

222  
223 **Reward function ( $\mathcal{R}$ ):** A function  $\mathcal{R} : \mathcal{S} \rightarrow [0, 1]$  providing a global reward based on the overall  
224 state  $s \in \mathcal{S}$  (database states, history), signaling task success or failure. For example, in `telecom`,  
225 the agent is rewarded if the issue ("no mobile data") is fixed, as verified by the user's database state.

226 **Instruction space ( $\mathcal{U}$ ):** The instruction space  $\mathcal{U}$  defines the scenario guiding realistic user simulation,  
227 as well as the domain policies to which the agent must adhere when assisting the user.

228 The Dec-POMDP formalism offers key advantages for simulating complex, interactive scenarios (see  
229 Figure 2 for an example trajectory of interactions). It enables realistic simulations of collaborative  
230 environments, such as technical support, where users perform actions guided by agents. This presents  
231 agents with crucial coordination and communication challenges. In addition, the formalism enhances  
232 the reliability and control of user simulation. By predefining user tools and their effects on user states,  
233 user behavior becomes more controllable and less reliant on extensive natural language prompting.

### 235 3.2 DOMAIN AND TASK CREATION

236 Similar to  $\tau$ -bench, we adopted a multi-stage creation process to build domain-specific materials for  
237 new domains. This process, illustrated using the `telecom` domain, involves the following stages:

238 **Stage 1: Creating agent's database schema and tools.** We begin by prompting Large Language  
239 Models (LLMs) to generate a Product Requirements Document (PRD) that outlines the domain's core  
240 business logic. This PRD specifies the database schema and necessary functions. In the `telecom`  
241 domain, this involved defining a customer CRM system with schemas for customers and lines, along  
242 with functions to manage them. An LLM then generates function implementations, a mock database,  
243 and unit tests based on the PRD. We manually refine the generated code until all unit tests pass.

244 **Stage 2: Creating user's database schema and tools.** For troubleshooting scenarios, we similarly  
245 use an LLM to define the user's database schema and tools. In the `telecom` domain, this included  
246 implementing a mocked user phone device with status (e.g., signal strength) and functions (e.g.,  
247 toggling airplane mode). Again, an LLM generates implementations, a mock database, and unit tests,  
248 which are then manually refined until all tests pass.

249 **Stage 3: Programmatic task creation.** We employ a combinatorial approach to generate diverse,  
250 verifiable tasks from atomic building blocks (see Appendix A.3 for details on our task factorization  
251 framework).

252 Each atomic subtask  $t$  is about a specific problem to be resolved, for example, airplane mode on leads  
253 to mobile data not working. Specifically, each subtask  $t$  is defined as  $(\{f_{t,k}^{init}\}, \{f_{t,k}^{sol}\}, \{f_{t,k}^{assert}\})$ ,  
254 where  $f_{t,k}$  is the  $k$ -th function call of the subtask  $t$  that interacts with the agent's or user's database:

- 255 • **Initialization functions**  $f_{t,k}^{init}$  specify calls to set up the initial task state, typically by  
256 updating the database values. For instance, in `telecom`, an initialization might be  
257 `set_airplane_mode(True)`.
- 258 • **Solution functions**  $f_{t,k}^{sol}$  specify tool calls to resolve issues introduced by initialization. For  
259 example, `toggle_airplane_mode()` could be a solution for the initialization example given  
260 above. Note that these must be tools available to the agent or user.
- 261 • **Assertion functions**  $f_{t,k}^{assert}$  specify conditions the final state  $\mathcal{S}$  must meet for the task to be  
262 considered solved. For instance, `assert_service_status("connected")` checks if the  
263 user's service is active in `telecom`.

264 While solution functions  $f_{t,k}^{sol}$  are restricted to agent or user tools, initialization and assertion functions  
265 can be any function in the relevant database.

266 Atomic subtasks are grouped such that mutually exclusive or alternative subtasks are in the same  
267 group. A composite task is created by selecting at most one subtask from each group, concatenating

270  
271  
272  
273  
274  
275  
276  
277  
278  
279  
280  
281  
282  
283  
284  
285  
286  
287  
288  
289  
290  
291  
292  
293  
294  
295  
296  
297  
298  
299  
300  
301  
302  
303  
304  
305  
306  
307  
308  
309  
310  
311  
312  
313  
314  
315  
316  
317  
318  
319  
320  
321  
322  
323  
Table 2: Comparison of data curation processes between  $\tau$ -Bench and  $\tau^2$ -Bench on AI usage and human efforts. This process will become increasingly automated with the progress of stronger AI models and refined pipelines, further reducing the entry barrier for creating new domains.

| Phase                | $\tau$ -Bench<br>(AI Usage)               | $\tau$ -Bench<br>(Human Efforts)                      | $\tau^2$ -Bench<br>(AI Usage)                                 | $\tau^2$ -Bench<br>(Human Efforts)       |
|----------------------|---|---|---|--|
| Agent Schema & Tools | Low - Limited assistance                  | High - Manually hand-crafted                          | High - Generates PRD, schemas, and code                       | Medium - PRD and Code refinement         |
| User Schema & Tools  | N/A                                       | N/A   | High - Generates schemas and tools                            | Medium - PRD and Code refinement         |
| Task Creation        | Low - No automation                       | High - Manually writes task case with unique solution | None - Programmatic verifiable generation via atomic subtasks | Low - Defines atomic subtasks and logic  |
| Agent Policy         | High - Generates domain-specific policies | Medium - Manually design business rules               | High - Generates domain-specific policies and user manuals    | Low - Refines logic and details          |
| Manual Refinement    | None                                      | High - debug cases                                    | Low - help with code and document refinements                 | High - Joint refinement of all materials |

their respective function calls. Task correctness is automatically verified by checking if the final state  $s \in \mathcal{S}$  satisfies all assertion functions after applying initialization and then solution functions. We also verify that the task is not resolved until all solution functions are applied.

In the `telecom` domain, we developed 15 atomic subtask groups for 3 user intents of increasing complexity: `service_issue`, `mobile_data_issue`, and `mms_issue`. Combining these subtasks in a programmatic way yields 2285 tasks. We then subsample 114 tasks to form a balanced distribution over different intents and numbers of subtasks (details in Appendix A). The number of subtasks in a task serves as a proxy for difficulty as more diagnostic and resolution steps are required.

**Stage 4: Creating domain-specific agent policy.** Based on the curated tasks and their solutions, we prompt LLMs to generate domain-specific policies for the agent. For troubleshooting, these policies guide the agent in diagnosing and resolving user issues, often outlining step-by-step procedures for common problems related to each user intent, details in Appendix D.2.

**Stage 5: Manual refinement.** We jointly refine all the domain materials including tools, policy and atomic subtasks to improve the quality of the domain.

As summarized in Table 2, curating domains in  $\tau^2$ -bench involves more AI usage and less human efforts compared to the original  $\tau$ -bench. Moreover,  $\tau^2$ -bench enables a developer to specifically associate a task with a Persona (a brief description of the user’s identity). We put this to use in our new domain. Each `telecom` domain task was randomly assigned one of the following personas: None, Easy, and Hard. The None persona means that no specific persona is provided to the user simulator. The Easy persona describe the profile of a user who is rather familiar with the domain while the Hard persona represents a more challenging user with low technical knowledge (see Appendix A.1).

### 3.3 TASK EVALUATION

The success of a task can be defined by different criteria: DB check, status assertions, natural language assertions, communication info check, and action matching. The DB check and communication info check are the same as the original  $\tau$ -bench. The status assertion involves verifying specific conditions in the final world state  $\mathcal{S}_{world}$  using the predefined assertion functions (e.g., checking if a service is connected). The natural language assertion involves verifying specific conditions in the final history state  $\mathcal{S}_{history}$  using a natural language description, like “the agent diagnosed the cause of the issue.”

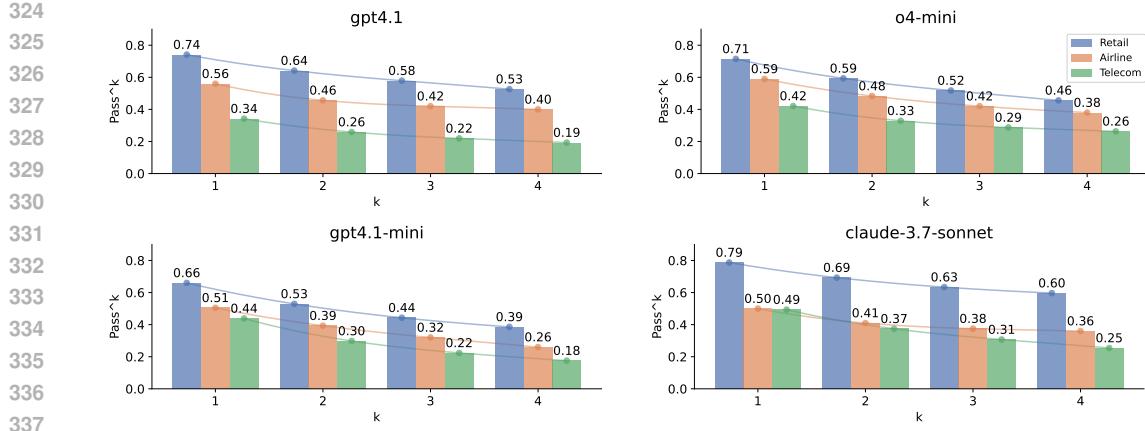


Figure 3:  $\text{pass}^k$  metrics across all evaluated domains (airline, retail, telecom). **Top Left:** gpt-4.1, **Top Right:** o4-mini, **Bottom Left:** gpt-4.1-mini, **Bottom Right:** claude-3.7-sonnet.

The action matching involves verifying if every solution function  $f_{t,k}^{sol}$  exists in the actual agent-user interaction trajectory. Practically, each task can specify a subset of these criteria based on its features. In telecom, only assertion functions are used to evaluate task success.

## 4 EXPERIMENTS

### 4.1 AGENT SETTINGS

All LLM API calls are implemented using the `Litellm` package (BerriAI, 2025). We evaluated four large language models: gpt-4.1-mini-2025-04-14, gpt-4.1-2025-04-14, o4-mini-2025-04-16, and claude-3-7-sonnet-20250219. The user simulator is implemented using gpt-4.1-2025-04-14. Each task is run four times, maintaining a consistent LLM temperature of 0 to promote deterministic outputs. Both the agents and the user simulator are implemented as function-calling agents. All tools are provided to LLMs in the OpenAI tools format. The agent prompt includes generic guidelines along with domain-specific policies. Similarly, the user prompt contains generic guidelines supplemented by task-specific instructions. Both domain policies and prompts are available in Appendices C and D.

When the gpt-4.1-2025-04-14 agent is paired with the gpt-4.1-2025-04-14 user simulator, the average agent/user simulation costs are \$0.086/\$0.059 per task, respectively. The cost of running all domains for 1 trial per task is approximately \$40.

### 4.2 RESULTS

**Pass^k scores.** We computed performance metrics on the verified  $\tau^2$ -bench domains (retail and airline) and on our new telecom domain (see Figure 3). Our findings indicate that the telecom domain presents a greater challenge, exhibiting an overall lower success rate compared to other domains. gpt-4.1 pass^1 drops from 74%/56% for retail and airline respectively to 34% for telecom. gpt-4.1-mini, o4-mini, and claude-3.7-sonnet perform better with pass^1 of around 50% for telecom. In the case of claude-3.7-sonnet, the pass^1 score for telecom (49%) is on par with airline. However, as k increases, the pass^k scores decline more rapidly for telecom compared to airline, suggesting less consistent performance on the telecom domain.

**Ablation analysis.** An agent’s success in  $\tau^2$ -bench depends on two things – 1) how well it can communicate and collaborate with the user at solving the issue, and 2) how well it can reason over and apply the domain guidelines specified in the policy document. In order to understand the impact of each of these components—reasoning and communication for dual control—we perform an ablation study. Specifically, we evaluate performance in the telecom domain across three distinct settings:

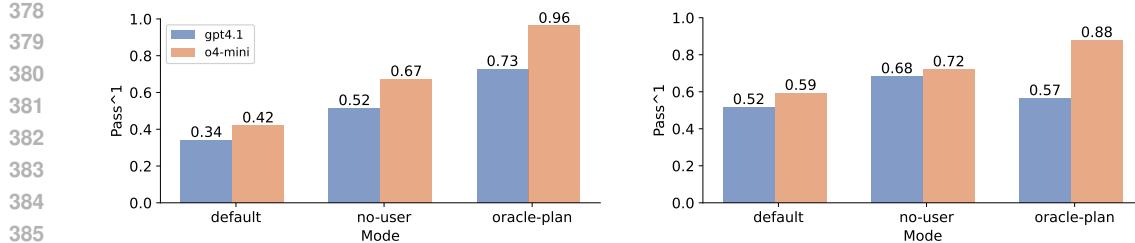


Figure 4:  $\text{pass}^1$  metrics for the Telecom domain across different operational modes (Default, No-User, Oracle Plan) using the Default user simulation. **Left:** original policy. **Right:** workflow-based policy. This figure illustrates the impact of reasoning load and decentralized control on agent performance.

- **Default:** The default agent and user simulator configuration where the agent and user collaborate in a *dual-control* setup.
- **No-User:** The agent is provided with a ticket summarizing the user’s problem and success criteria. The agent controls all tools, including those typically operated by the user, and is solely responsible for solving the problem. This setting tests the agent’s reasoning and tool-calling capability independently of its capacity to interact with the user.
- **Oracle Plan:** The agent is provided with the sequence of tool calls required to solve the problem, encompassing actions for both the agent and the user. This setup alleviates the agent’s reasoning load, focusing on its ability to collaborate with the user to execute a known plan.

Figure 4 (left) reports performance across these settings for gpt-4.1 and o4-mini, revealing key insights. The difference between Oracle Plan and Default configurations highlights the impact of the reasoning load on agent performance. Unsurprisingly, providing the ground truth leads to better performance than the Default setting. But it is notable that this effect is larger for the o4-mini than for gpt-4.1, suggesting that o4-mini is better able to make use of the ground truth information.

The comparison between No-User and Default modes illustrates the impact of dual control and the associated communication overhead on agent error rates and overall success. For both models, shifting from no user operation (No-User) to a collaborative setup (Default) where the agent must guide the user results in a substantial drop in  $\text{pass}^1$  (18% drop for gpt-4.1 and 25% drop for o4-mini). This underscores that LLMs still face significant challenges when solving problems with an active user who shares control of the environment.

**Impact of policy document on performance.** Figure 4 also allows us to analyze the impact of the policy document on task success. Specifically, we created an alternate policy document that provides more specific details on the workflow required to solve each type of task, with the rationale that having the workflow provides more concrete guidance to the agent (see Appendices D.2.2 and D.2.3). We observe that this is indeed the case and slightly improves agent performance using the workflow policy (right) over the original one (left) under the Default and No-User modes. Surprisingly, workflow policy document hurts performance on Oracle Plan for both gpt-4.1 and o4-mini. Our hypothesis here is that since the agent already has the ground truth action sequence, providing it the workflow might lead to confusion and hurt its performance rather than help.

**Impact of number of actions and sub-tasks.** Figure 5 breaks down the  $\text{pass}^1$  scores across various tasks in telecom binned by the number of actions required to solve them (left) or the number of different sub-tasks that need to be addressed (right). *transfer* refers to the special case of a task that requires to be transferred to a human and cannot be solved by the agent alone.

As expected, regardless of the base model being gpt-4.1 or o4-mini, agent performance drops as the number of actions increases, reaching close to zero for  $> 7$  actions in Default mode. The No-User mode results in higher scores overall, although the gap reduces (from about 0.3-0.4 to  $< 0.2$ ) as the number of actions increases. This hints that maintaining reliability over longer-horizon tasks remains a challenge under both settings and communication with the user is not the only bottleneck. Interestingly, for No-User, gpt-4.1 performs better at the tail end (10 actions or more) than o4-mini.

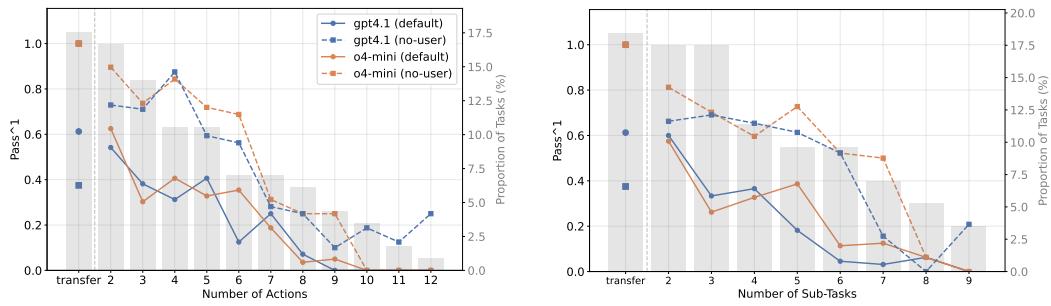


Figure 5:  $\text{pass}^1$  scores across various tasks in `telecom` binned by the number of actions required to solve them (left) or the number of different issues that need to be addressed (right). *transfer* refers to the special case of a task that requires to be transferred to a human and cannot be solved by the agent alone. (Grey bars indicates the proportion of the tasks that fall into that bin.)

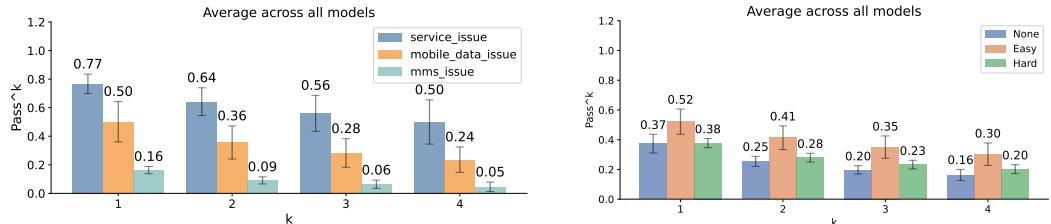


Figure 6:  $\text{pass}^k$  per issue (left) and persona (right) type for the `telecom` domain averaged across all models. Performance is shown for `service_issue`, `mobile_data_issue`, and `mms_issue` issue types and across different persona types, highlighting how different issue and persona types affect success rates.

We observe a similar trend with increase in the number of distinct sub-tasks per task – performance trends downwards for both base models, with the `no-user` mode generally being higher than the `default` mode. Both these results validate that our domain design and task creation process provide a natural path to scaling complexity via combining different sub-tasks into a single task.

**Impact of issue and persona types.** The `telecom` domain is organized around three primary user intents and personas reflecting different difficulties. Figure 6 provides a breakdown analysis of performance by issue and persona types. We observe that the  $\text{pass}^k$  scores for all LLMs are driven by higher failure rates on more complex issue types (`mobile_data_issue` and `mms_issue`), while the agent tends to perform better on tasks associated with the `Easy` persona compared to those associated with the `Hard` and (interestingly) the `None` one. Refer to Appendix A.2 for more details.

#### 4.3 HOW DOES DUAL-CONTROL IMPACT BENCHMARK RELIABILITY?

Ensuring the reliability of conversational agent benchmarks is paramount. Three primary sources of uncertainty can impact benchmark reliability: **implementation errors**, **task specification errors**, and **user simulator errors**. While the user simulator is often cited as a critical component requiring careful evaluation, its assessment can be confounded by issues in the benchmark’s implementation or task definitions. Therefore, we first address these potential error sources before evaluating the user simulator itself. This is detailed in Appendix B.

**User simulator quality evaluation.** Having minimized errors in the benchmark implementation and task specifications, we evaluated the user simulator quality. To assess the quality of the user simulator across domains, we manually annotated interaction traces generated using `gpt-4.1` for both the User Simulator and the Agent. Each conversation was reviewed by two separate annotators tasked with identifying user simulator errors. Annotators were given the User Simulator Guidelines, the specific User Instructions for this conversation (see Appendix C.2), descriptions of the available User Tools (if any), and the complete conversation trajectory (messages and tool calls). Annotators assessed each user turn against four criteria: adherence to **User Simulator Guidelines**, adherence to **User Instructions**, correct use of **User Tools**, and generation of a **natural and consistent** conversational

486  
 487 Table 3: User simulator error count (*rates*) across domains. Only critical user errors prevent the agent  
 488 from solving the task while benign errors do not affect task completion. New `telecom` domain  
 489 shows much lower error rate and no critical errors were reported. (See Appendix E for more details).

| 490 Domain  | 491 Num Conversations | 492 Critical Errors | 493 Benign Errors | 494 Total Errors |
|-------------|-----------------------|---------------------|-------------------|------------------|
| 491 airline | 492 100               | 493 13 (13%)        | 494 34 (34%)      | 495 47 (47%)     |
| 492 retail  | 493 50                | 494 6 (12%)         | 495 14 (28%)      | 496 20 (40%)     |
| 493 telecom | 494 50                | 495 3 (6%)          | 496 5 (10%)       | 497 8 (16%)      |

498 continuation. Errors were categorized as either (1) **task-critical** errors: high-severity failures that  
 499 preclude task completion (e.g., generating an intent that contradicts the user goal, or causing an  
 500 irrecoverable state transition), or as (2) **task-benign** errors: Errors that do not prevent the task from  
 501 being completed.

502 **Reliability of the user simulator.** As shown in Table 3, our analysis of user simulator behavior  
 503 reveals significant improvements in reliability for the new `telecom` domain. While for the `retail`  
 504 and `airline` domains we recorded a 40% and 47% error rate for the user simulator (with 12% and  
 505 13% being critical errors that prevent task completion), this rate is much lower for the `telecom`  
 506 domain, only 16% with 6% critical errors reported. This substantial improvement in reliability can  
 507 be attributed to the domain design, which shapes and tightly constrains user behavior through its  
 508 environment and available affordances. Rather than relying heavily on natural language specifications  
 509 to guide behavior, the `telecom` domain’s structured interface and clear action space naturally guide  
 510 the user simulator toward correct interactions, resulting in more consistent and predictable behavior.

## 5 CONCLUSION

511 We present  $\tau^2$ -bench that generalizes  $\tau$ -bench by introducing the dual-control setting and found  
 512 a substantial performance drop in LLMs due to coordination and communication requirements,  
 513 highlighting these as critical bottlenecks over pure reasoning capabilities for solving user requests.

514 More work remains to be done to improve the user simulator. Although we have shown that  
 515 augmenting users with curated tools can help avoid critical errors, we have not yet investigated how  
 516 this method could be applied to the existing `airline` and `retail` domains. Doing so would pave  
 517 the way towards a more generic solution to ensuring high quality user simulator. Extending domain  
 518 coverage for the benchmark still heavily relies on human experts. For benchmarking methods to  
 519 be adopted by industry, providing much needed standards, it is critical to further investigate how to  
 520 automate the domain curation process.

521 One important limitation of  $\tau^2$ -bench is that it does not explicitly model the expert-novice gap inherent  
 522 to most customer support tasks. When interacting with a naive user, an expert must understand the  
 523 user’s mental model and adapt explanations accordingly. Assessing and improving the AI agent’s  
 524 abilities to bridge this gap is a promising direction for future work and  $\tau^2$ -bench provides a strong  
 525 starting point for such explorations.

526  
 527  
 528  
 529  
 530  
 531  
 532  
 533  
 534  
 535  
 536  
 537  
 538  
 539

540 REPRODUCIBILITY STATEMENT  
541542 We have provided all codes and data in the supplementary materials and will make them open source.  
543 The experiments can be reproduced by following the experimental settings and our README in the  
544 code.  
545546 ETHICS STATEMENT  
547548 The development of standardized benchmarks for Large Language Models (LLMs) and AI agents is  
549 crucial for ensuring societal control and fostering fairness amidst rapidly advancing technologies.  
550 Such benchmarks not only provide a framework for transparent evaluation but also enable research  
551 groups to coordinate their efforts around common tasks, thereby accelerating the overall progress  
552 in the field. While this work itself may not have direct negative societal implications, it contributes  
553 to the development of real-world agents, which will invariably have diverse economic and societal  
554 consequences. Therefore, it is also of paramount importance that AI agents are designed to collab-  
555 orate effectively and safely with human users, a prerequisite for their responsible integration into  
556 commercial settings and everyday life.  
557558 REFERENCES  
559560 Jacob Andreas, John Bufe, David Burkett, Charles Chen, Josh Clausman, Jean Crawford, Kate Crim,  
561 Jordan DeLoach, Leah Dorner, Jason Eisner, et al. Task-oriented dialogue as dataflow synthesis.  
562 *Transactions of the Association for Computational Linguistics*, 8:556–571, 2020.563 Anthropic. Claude 3.7 Sonnet, 2025. URL <https://www.anthropic.com/news/claude-3-7-sonnet>. Model release: 2025-02-24.  
564565 BerriAI. litellm, 2025. URL <https://github.com/BerriAI/litellm>.  
566567 Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Inigo Casanueva, Stefan Ultes, Osman  
568 Ramadan, and Milica Gašić. Multiwoz—a large-scale multi-domain wizard-of-oz dataset for  
569 task-oriented dialogue modelling. *arXiv preprint arXiv:1810.00278*, 2018.570 Derek Chen, Howard Chen, Yi Yang, Alex Lin, and Zhou Yu. Action-based conversations dataset: A  
571 corpus for building more in-depth task-oriented dialogue systems. *arXiv preprint arXiv:2104.00783*,  
572 2021.  
573574 Izzeddin Gür, Dilek Hakkani-Tür, Gokhan Tür, and Pararth Shah. User modeling for task oriented  
575 dialogues. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pp. 900–906, 2018. doi:  
576 10.1109/SLT.2018.8639652.577 He He, Derek Chen, Anusha Balakrishnan, and Percy Liang. Decoupling strategy and generation in  
578 negotiation dialogues. *arXiv preprint arXiv:1808.09637*, 2018.  
579580 Zhiyuan Hu, Yue Feng, Anh Tuan Luu, Bryan Hooi, and Aldo Lipani. Unlocking the potential  
581 of user feedback: Leveraging large language model as user simulators to enhance dialogue  
582 system. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge  
583 Management, CIKM '23*. ACM, October 2023. doi: 10.1145/3583780.3615220. URL <http://dx.doi.org/10.1145/3583780.3615220>.  
584585 Yue Huang, Jiawen Shi, Yuan Li, Chenrui Fan, Siyuan Wu, Qihui Zhang, Yixin Liu, Pan Zhou, Yao  
586 Wan, Neil Zhenqiang Gong, et al. Metatool benchmark for large language models: Deciding  
587 whether to use tools and which to use. *arXiv preprint arXiv:2310.03128*, 2023.588 Carlos E Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik  
589 Narasimhan. Swe-bench: Can language models resolve real-world github issues? *arXiv preprint  
590 arXiv:2310.06770*, 2023.  
591592 Taaha Kazi, Ruiliang Lyu, Sizhe Zhou, Dilek Hakkani-Tur, and Gokhan Tur. Large Language  
593 Models as User-Agents for Evaluating Task-Oriented-Dialogue Systems, November 2024. URL  
594 <http://arxiv.org/abs/2411.09972>. arXiv:2411.09972.

594 Elad Levi and Ilan Kadar. Intellagent: A multi-agent framework for evaluating conversational ai  
595 systems. *arXiv preprint arXiv:2501.11067*, 2025.

596

597 Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding,  
598 Kaiwen Men, Kejuan Yang, et al. Agentbench: Evaluating llms as agents. *arXiv preprint*  
599 *arXiv:2308.03688*, 2023.

600 Jiarui Lu, Thomas Holleis, Yizhe Zhang, Bernhard Aumayer, Feng Nan, Felix Bai, Shuang Ma, Shen  
601 Ma, Mengyu Li, Guoli Yin, et al. Toolsandbox: A stateful, conversational, interactive evaluation  
602 benchmark for llm tool use capabilities. *arXiv preprint arXiv:2408.04682*, 2024.

603 Frans A Oliehoek, Christopher Amato, et al. *A concise introduction to decentralized POMDPs*,  
604 volume 1. Springer, 2016.

605

606 OpenAI. gpt-4.1, 2025a. URL <https://openai.com/index/gpt-4-1/>. Model release:  
607 2025-04-14.

608 OpenAI. o4-mini, 2025b. URL <https://openai.com/index/o3-o4-mini-system-card/>. Model release: 2025-04-16.

609

610 Olivier Pietquin and Helen Hastie. A survey on metrics for the evaluation of user simulations. *The  
611 knowledge engineering review*, 28(1):59–73, 2013. URL <https://www.cambridge.org/core/journals/knowledge-engineering-review/article/survey-on-metrics-for-the-evaluation-of-user-simulations/602976EC6417B5BAA1719D0876FB5611>. Publisher: Cambridge University Press.

611

612 Akshara Prabhakar, Zuxin Liu, Weiran Yao, Jianguo Zhang, Ming Zhu, Shiyu Wang, Zhiwei Liu,  
613 Tulika Awalgona, Haolin Chen, Thai Hoang, et al. Apigen-mlt: Agentic pipeline for multi-turn  
614 data generation via simulated agent-human interplay. *arXiv preprint arXiv:2504.03601*, 2025.

615

616 Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru  
617 Tang, Bill Qian, et al. Toolllm: Facilitating large language models to master 16000+ real-world  
618 apis. In *ICLR*, 2024.

619

620 Yangjun Ruan, Honghua Dong, Andrew Wang, Silviu Pitis, Yongchao Zhou, Jimmy Ba, Yann Dubois,  
621 Chris J Maddison, and Tatsunori Hashimoto. Identifying the risks of lm agents with an lm-emulated  
622 sandbox. *arXiv preprint arXiv:2309.15817*, 2023.

623

624 Jost Schatzmann, Daniel Jurafsky, Michael Galley, and David Trevillian. Evaluating agenda-based  
625 user simulation for reinforcement learning of dialogue management. In *Speech Communication*,  
626 volume 47, pp. 95–121, 2007.

627

628 Ruixuan Xiao, Wentao Ma, Ke Wang, Yuchuan Wu, Junbo Zhao, Haobo Wang, Fei Huang, and  
629 Yongbin Li. Flowbench: Revisiting and benchmarking workflow-guided planning for llm-based  
630 agents. *arXiv preprint arXiv:2406.14884*, 2024.

631

632 Fanjia Yan, Huanzhi Mao, Charlie Cheng-Jie Ji, Tianjun Zhang, Shishir G. Patil, Ion Stoica,  
633 and Joseph E. Gonzalez. Berkeley function calling leaderboard. [https://gorilla.cs.berkeley.edu/blogs/8\\_berkeley\\_function\\_calling\\_leaderboard.html](https://gorilla.cs.berkeley.edu/blogs/8_berkeley_function_calling_leaderboard.html),  
634 2024.

635

636 Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. Webshop: Towards scalable  
637 real-world web interaction with grounded language agents. *Advances in Neural Information  
638 Processing Systems*, 35:20744–20757, 2022.

639

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

642

643 Shuyan Zhou, Frank F Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng,  
644 Yonatan Bisk, Daniel Fried, Uri Alon, et al. WebArena: A Realistic Web Environment for Building  
645 Autonomous Agents. *arXiv preprint arXiv:2307.13854*, 2023.

646

647 Kunlun Zhu, Hongyi Du, Zhaochen Hong, Xiaocheng Yang, Shuyi Guo, Zhe Wang, Zhenhailong  
648 Wang, Cheng Qian, Xiangru Tang, Heng Ji, et al. Multiagentbench: Evaluating the collaboration  
649 and competition of llm agents. *arXiv preprint arXiv:2503.01935*, 2025.

648 APPENDIX  
649650 A TELECOM DOMAIN  
651652 The number of tasks spanning different user intents and number of tasks are shown in Table 4.  
653654 Table 4: Number of tasks sampled for each intent and number of subtasks.  
655

| 656 Number of subtasks | 657 <b>service_issue</b> | 658 <b>mobile_data_issue</b> | 659 <b>mms_issue</b> | 660 total |
|------------------------|--------------------------|------------------------------|----------------------|-----------|
| 658 2                  | 659 9                    | 660 8                        | 661 8                | 662 25    |
| 659 3                  | 660 9                    | 661 8                        | 662 9                | 663 26    |
| 660 4                  | 661 9                    | 662 6                        | 663 6                | 664 21    |
| 661 5                  | 662 2                    | 663 6                        | 664 5                | 665 13    |
| 662 6                  | 663 -                    | 664 5                        | 665 6                | 666 11    |
| 663 7                  | 664 -                    | 665 3                        | 666 5                | 667 8     |
| 664 8                  | 665 -                    | 666 -                        | 667 4                | 668 4     |
| 665 9                  | 666 -                    | 667 -                        | 668 6                | 669 6     |
| 666 total              |                          | 667 29                       | 668 36               | 669 49    |
| 670                    |                          | 671 114                      |                      |           |

672 There are 3 different intents in the telecom domain: `service_issue`, `mobile_data_issue`,  
673 and `mms_issue`. How the number of actions required to solve the issues varies between intents is  
674 shown in Table 5.  
675676 Table 5: Number of actions required to solve the issues.  
677

| 678 Intent                         | 679 Mean | 680 Std  | 681 Min | 682 Max |
|------------------------------------|----------|----------|---------|---------|
| 683 <code>service_issue</code>     | 684 2.31 | 685 2.25 | 686 1   | 687 8   |
| 688 <code>mobile_data_issue</code> | 689 4.31 | 690 1.79 | 691 2   | 692 8   |
| 693 <code>mms_issue</code>         | 694 6.00 | 695 2.85 | 696 2   | 697 12  |

698 A.1 USER PERSONA  
699700 We define two distinct user personas to represent different levels of technical expertise and comfort  
701 with technology:  
702703 **Persona 1: Easy** As a 41-year-old office administrator, you use your cellphone daily for  
704 both work and personal tasks. While you're familiar with common phone functions, you  
705 wouldn't call yourself a tech enthusiast.  
706707 Your technical skills are average - you handle standard smartphone features like calls,  
708 texts, email, and basic apps with ease. You understand the fundamental settings, but  
709 prefer clear, step-by-step guidance when trying something new.  
710711 In interactions, you're naturally friendly and patient. When receiving help, you listen at-  
712 tentively and aren't afraid to ask questions. You make sure to confirm your understanding  
713 and provide detailed feedback on each instruction you receive.  
714715 **Persona 2: Hard** At 64 years old, you're a retired librarian who keeps your phone use  
716 simple - mainly for calls, texts, and capturing photos of your grandchildren. Technology  
717 in general makes you feel uneasy and overwhelmed.  
718719 Your technical knowledge is quite limited. Step-by-step instructions often confuse you,  
720 and technical terms like "VPN" or "APN" might as well be a foreign language. You only  
721 share information when specifically asked.  
722723 When dealing with technology, you tend to get flustered quickly. You need constant  
724 reassurance and often interrupt with anxious questions. Simple requests like "reboot the  
725 phone" can trigger worries about losing precious photos.  
726

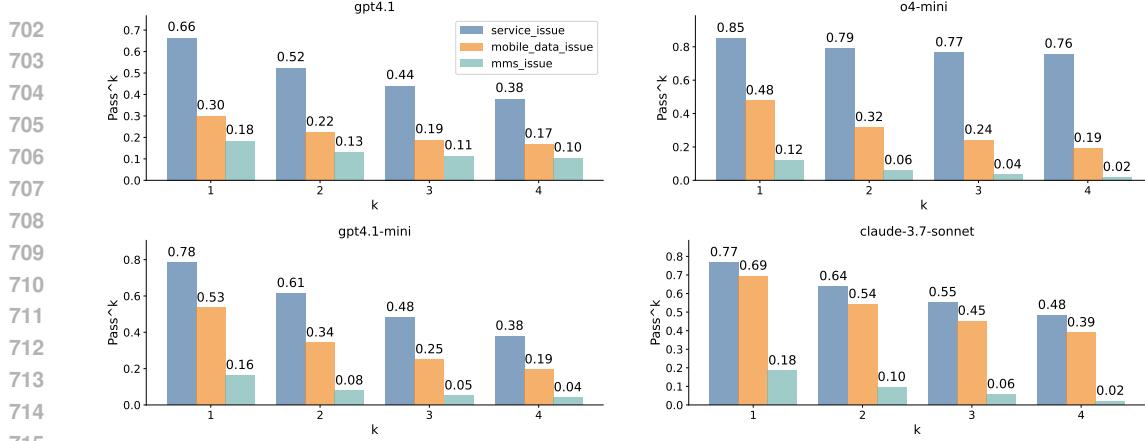


Figure 7: gpt-4.1 (top left), o4-mini (top right), gpt-4.1-mini (bottom left), and claudie-3.7-sonnet (bottom right)  $\text{Pass}^k$  per issue type for the telecom domain. Performance is shown for service\_issue, mobile\_data\_issue, and mms\_issue issue types, highlighting how different issue types affect success rates.

## A.2 MORE EXPERIMENTAL ANALYSIS

**Impact of issue types.** The telecom domain is organized around three primary user intents reflecting the three different issues that can be encountered: service\_issue, mobile\_data\_issue, and mms\_issue, each of which contains specific procedures laid out by the domain policy. These issue types are designed to have an inherent difficulty hierarchy. For instance, service\_issue tasks can typically be resolved independently through a straightforward sequence of actions. In contrast, successfully addressing mobile\_data\_issue or mms\_issue often requires first checking for and potentially resolving underlying service\_issue problems. This dependency creates a natural ordering in task difficulty, with service\_issue being the easiest, while mobile\_data\_issue and mms\_issue represent more complex, multi-stage problems.

Figure 7 provides an breakdown analysis of performance by issue type. We observe that the  $\text{Pass}^k$  scores for all LLMs (gpt-4.1, o4-mini, claudie-3.7-sonnet, gpt-4.1-mini) are driven by higher failure rates on more complex issue types (mobile\_data\_issue and mms\_issue). This suggests that the multi-stage reasoning and conditional logic required for harder issue types pose a substantial challenge to the agents. We also notice that the spread across issue types differs slightly by the model. For instance, claudie-3.7-sonnet does better than o4-mini on mobile\_data\_issue but worse on service\_issue.

**Impact of user persona.** Figure 8 provides a breakdown analysis of performance by user persona. Results confirm that the agent tends to perform better on tasks associated with the Easy persona compared to those associated with the Hard one. Interestingly, the performances of the agent on tasks involving no persona information (None) tend to be be on par or lower to performances on tasks associated with the Hard persona. This highlights the critical importance of testing AI systems with well-defined user personas before real-world deployment.

## A.3 TASK FACTORIZATION FRAMEWORK

To ensure our tasks are both complex and verifiable, we use a formal root-cause factorization framework. This method systematically breaks down any high-level, user-visible issue into its independent, underlying causes, allowing us to programmatically control task structure and difficulty.

The framework consists of the following components:

**Issue Predicate ( $I\text{st}riggered}_I$ )** First, we define a high-level predicate that formally represents the user-visible problem. This function,  $I\text{st}riggered}_I(s)$ , evaluates to true if the issue is present in a given environment state  $s$ . For example, in the "No Service" task (Appendix A.1),  $I\text{st}riggered}_I(s)$  is true if a call to `check_status_bar(s)` indicates "No Service."

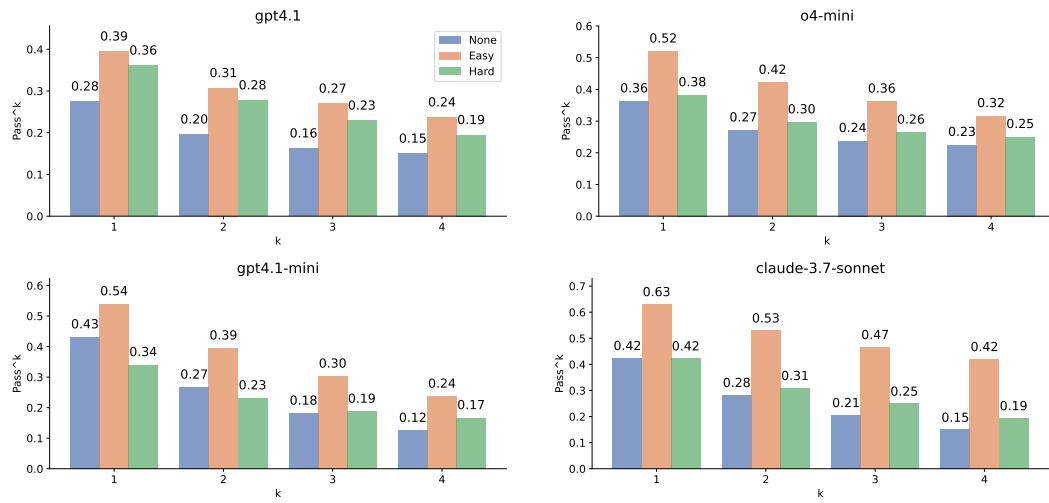


Figure 8: gpt-4.1 (top left), o4-mini (top right), gpt-4.1-mini (bottom left), and claude-3.7-sonnet (bottom right) pass<sup>k</sup> per persona type for the telecom domain. Performance is shown across different persona types, highlighting how different user characteristics affect success rates.

**Parameter Groups ( $G_j$ )** Next, we identify underlying system parameters that could cause the issue and partition them into disjoint, independently-acting groups ( $G_1, G_2, \dots, G_m$ ). Each group represents a distinct causal mechanism. Since we control the implementation of the environment, we can manually adjust it to ensure that it is amenable to such partitioning of the parameters. In our example, we identify two independent groups:  $G_1 = \{\text{airplane\_mode\_status}\}$  and  $G_2 = \{\text{sim\_card\_seated\_status}\}$ . The state of airplane mode does not affect the physical status of the SIM card, making them independent.

**Root-Cause Predicates ( $RC_j$ )** For each parameter group  $G_j$ , we define a corresponding low-level predicate,  $RC_j(s)$ , which evaluates to true only if that specific group is the cause of a problem. Following our example,  $RC_1(s)$  is true if `airplane_mode_status(s) == ON`, and  $RC_2(s)$  is true if `sim_card_seated_status(s) == UNSEATED`.

**The Decomposition Property** The power of this framework lies in formally connecting the high-level issue with its low-level root causes. The Decomposition Property guarantees that our factorization is complete and sound:

$$IsTriggered_I(s) \Leftrightarrow \bigvee_j RC_j(s)$$

This property states that the user-visible issue is present if and only if at least one of its independent root causes is active.

**Atomic Subtask Definition** With this framework in mind, we can clarify the definition of an atomic subtask. An atomic subtask is the fundamental building block in our benchmark, formally tied to a single root cause ( $RC_j$ ):

- **Initialization** ( $init_j$ ): A set of privileged functions whose goal is to activate a single root-cause predicate,  $RC_j$ . For example, the `turn_airplane_mode_on()` action activates the root cause where `airplane_mode_status` is ON.
- **Solution** ( $sol_j$ ): A sequence of tool calls available to the agent/user. The purpose of these actions is to deactivate the same root-cause predicate,  $RC_j$ . For instance, the `toggle_airplane_mode()` action deactivates the root cause by setting `airplane_mode_status` back to OFF.
- **Assertion** ( $assert_I$ ): A function that verifies if the overall issue  $I$  has been resolved (i.e.,  $\neg IsTriggered_I(s)$ ). This same assertion is used across all atomic subtasks related to the same

810 issue. For the "No Service" issue, the assertion verifies that the status bar shows a signal, regardless  
 811 of which specific root cause was just fixed.  
 812

#### 813 A.4 EXAMPLE TASK

```

815
816 # Task Details
817
818 ## ID
819 [service_issue]airplane_mode_on|unseat_sim_card
820
821 ## Description
822 - **Purpose**: Test resolution path: No Service/Connection Issues.
823
824 ## User Scenario
825 - **Instructions**:
826   - **Domain**: telecom
827   - **Reason for call**: Your phone has been showing 'No Service' for
     the past few hours.
828   - **Known info**: You are John Smith with phone number 555-123-2002.
829   - **Unknown info**: null
830   - **Task instructions**: If the agent suggests actions that don't
     immediately fix the issue,
     follow their guidance but express mild frustration after the first
     unsuccessful attempt.
     You will consider the issue resolved when the status bar shows that
     you have signal.
831     If the tool call does not return updated status information, you
     might need to perform
     another tool call to get the updated status.
832
833
834
835
836 ## Ticket
837 The user is experiencing issues with their phone service. They are
838 unable to make or receive
839 calls, and the status bar shows 'No Service'. Customer name: John Smith,
840 phone number:
841 555-123-2002. They will consider the issue resolved when the status bar
842 shows that they have
843 signal.
844
845 ## Initial State
846 - **Initialization Data**: null
847 - **Initialization Actions**:
848   1. **Action**: set_user_info
     - **Env Type**: user
     - **Arguments**:
       - name: John Smith
       - phone_number: 555-123-2002
   2. **Action**: turn_airplane_mode_on
     - **Env Type**: user
     - **Arguments**: {}
   3. **Action**: unseat_sim_card
     - **Env Type**: user
     - **Arguments**: {}
849
850
851
852
853
854
855
856 ## Evaluation Criteria
857 ### Actions
858 1. **Action ID**: toggle_airplane_mode_0
     - **Requestor**: user
     - **Name**: toggle_airplane_mode
     - **Arguments**: {}
859
860
861
862 2. **Action ID**: reseat_sim_card_1
     - **Requestor**: user
863

```

```

864
865     - **Name**: reseat_sim_card
866     - **Arguments**: {}
867
### Environment Assertions
868     - **Env Type**: user
869     - **Function**: assert_service_status
870     - **Arguments**:
871         - expected_status: connected
872     - **Assert Value**: true
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917

```

### A.5 EXAMPLE TRAJECTORY 1: DEFAULT MODE

Trajectory for the task in Appendix A.4 in the Default mode.

**Assistant:** *Hi! How can I help you today?*

**User:** *Hi there! My phone has been showing "No Service" for the past few hours. I'm unable to make calls or use mobile data. Can you help me fix this?*

**Assistant:** *To assist you with your "No Service" issue, I need to identify your account first. Could you please provide one of the following:*

- The phone number associated with your account
- Your customer ID
- Your full name and date of birth

*Once I have this information, I can begin troubleshooting your issue.*

**User:** *Sure! My phone number is 555-123-2002, and my name is John Smith. Let me know if you need anything else to proceed.*

**Assistant:** *[System retrieves customer information]*

**Tool:** [Tool call: get\_customer\_by\_phone(phone\_number: "555-123-2002")]

```

896
897     {
898         "customer_id": "C1001",
899         "full_name": "John Smith",
900         "date_of_birth": "1985-06-15",
901         "email": "john.smith@example.com",
902         "phone_number": "555-123-2002",
903         "address": {
904             "street": "123 Main St",
905             "city": "Anytown",
906             "state": "CA",
907             "zip_code": "90210"
908         },
909         "account_status": "Active",
910         "payment_methods": [
911             {
912                 "method_type": "Credit Card",
913                 "account_number_last_4": "1235",
914                 "expiration_date": "12/2026"
915             },
916             {
917                 "line_ids": ["L1001", "L1002", "L1003"],
918                 "bill_ids": ["B1001", "B1002", "B1003"],
919                 "created_at": "2025-01-15 10:30:00",
920                 "last_extension_date": null,
921                 "goodwill_credit_used_this_year": 25.0
922             }
923     }

```



972  
 973     **User:** *I re-seated the SIM card, and now the status bar shows full signal strength*  
 974     *with a 5G connection. It looks like the issue is resolved! Thank you!*  
 975  
 976     **Assistant:** *I'm glad to hear that reseating the SIM card resolved your issue and*  
 977     *your phone now has full signal strength with a 5G connection.*  
 978     *If you experience any further issues or have other questions, feel free to reach out.*  
 979     *Have a great day!*  
 980  
 981     **User:** *Thank you so much for your help! Everything seems to be working perfectly*  
 982     *now. Have a wonderful day! [STOP]*  
 983

## 982     A.6 EXAMPLE TRAJECTORY 2: NO-USER MODE

983  
 984     Trajectory for the task in Appendix A.4 in the No-User mode.

985  
 986     **Assistant:** *[System retrieves customer information]*  
 987     **Tool:** [Tool call: get\_customer\_by\_phone(phone\_number:  
 988     "555-123-2002")]  
 989  
 990     {  
 991         "customer\_id": "C1001",  
 992         "full\_name": "John Smith",  
 993         "date\_of\_birth": "1985-06-15",  
 994         "email": "john.smith@example.com",  
 995         "phone\_number": "555-123-2002",  
 996         "address": {  
 997             "street": "123 Main St",  
 998             "city": "Anytown",  
 999             "state": "CA",  
 1000             "zip\_code": "90210"  
 1001         },  
 1002         "account\_status": "Active",  
 1003         "payment\_methods": [  
 1004             {"method\_type": "Credit Card",  
 1005             "account\_number\_last\_4": "1235",  
 1006             "expiration\_date": "12/2026"  
 1007         }],  
 1008         "line\_ids": ["L1001", "L1002", "L1003"],  
 1009         "bill\_ids": ["B1001", "B1002", "B1003"],  
 1010         "created\_at": "2025-01-15 10:30:00",  
 1011         "last\_extension\_date": null,  
 1012         "goodwill\_credit\_used\_this\_year": 25.0  
 1013     }  
 1014  
 1015     **Assistant:** *[System retrieves line details]*  
 1016     **Tool:** [Tool call: get\_details\_by\_id(id: "L1001")]  
 1017  
 1018     {  
 1019         "line\_id": "L1001",  
 1020         "phone\_number": "555-123-2001",  
 1021         "status": "Active",  
 1022         "plan\_id": "P1001",  
 1023         "device\_id": "D1001",  
 1024         "data\_used\_gb": 3.2,  
 1025         "data\_refueling\_gb": 0.0,  
 1026         "roaming\_enabled": false,  
 1027         "contract\_end\_date": "2026-12-31",  
 1028         "last\_plan\_change\_date": "2025-01-10",  
 1029         "last\_sim\_replacement\_date": null,

```

1026         "suspension_start_date": null
1027     }
1028
1029 Assistant: [System retrieves additional line details]
1030
1031 Tool: [Tool call: get_details_by_id(id: "L1002")]
1032 {
1033     "line_id": "L1002",
1034     "phone_number": "555-123-2002",
1035     "status": "Active",
1036     "plan_id": "P1002",
1037     "device_id": "D1002",
1038     "data_used_gb": 8.7,
1039     "data_refueling_gb": 0.0,
1040     "roaming_enabled": true,
1041     "contract_end_date": "2026-12-31",
1042     "last_plan_change_date": "2024-12-15",
1043     "last_sim_replacement_date": "2025-01-20",
1044     "suspension_start_date": null
1045 }
1046
1047 Assistant: [System retrieves device details]
1048
1049 Tool: [Tool call: get_details_by_id(id: "D1002")]
1050 {
1051     "device_id": "D1002",
1052     "device_type": "phone",
1053     "model": "Smartphone Pro Max",
1054     "imei": "234567890123456",
1055     "is_esim_capable": true,
1056     "activated": true,
1057     "activation_date": "2025-02-15 11:45:00",
1058     "last_esim_transfer_date": "2025-01-20 09:30:00"
1059 }
1060
1061 Assistant: [System checks network status]
1062
1063 Tool: [Tool call: get_network_status()]
1064
1065     Airplane Mode: ON
1066     SIM Card Status: invalid
1067     Cellular Connection: no_service
1068     Cellular Signal: none
1069     Cellular Network Type: none
1070     Mobile Data Allowed: Yes
1071     Roaming: No
1072     Data Roaming Allowed: No
1073     Wi-Fi Radio: OFF
1074     Wi-Fi Connected: No
1075
1076 Assistant: [System toggles airplane mode]
1077
1078 Tool: [Tool call: toggle_airplane_mode()]
1079
1080     Airplane Mode is now OFF.
1081     Status Bar: [No Signal] | [Battery 80%]
1082
1083 Assistant: [System checks SIM status]
1084
1085 Tool: [Tool call: get_sim_status()]

```

1080           *The SIM card is invalid or not recognized.*  
 1081  
 1082           **Assistant:** *[System reseats SIM card]*  
 1083           **Tool:** *[Tool call: reseat\_sim\_card() ]*  
 1084           SIM card re-seated successfully.  
 1085           Status Bar: *[Signal 4] Excellent | 5G | [Data] Enabled | [Battery 80%]*  
 1086  
 1087           **Assistant:** *[STOP]*  
 1088

## 1089           B VERIFYING ORIGINAL $\tau^2$ -BENCH

### 1090           B.1 VERIFYING IMPLEMENTATION

1093           To minimize implementation errors, we (1) provide a unified tool formalism across all domains,  
 1094           ensuring consistency in how agent capabilities are represented, (2) we ensure that each domain  
 1095           environment specifies a data model, and (3) introduced a mock domain specifically designed for unit  
 1096           testing, allowing for isolated verification of core benchmark functionalities.

### 1097           B.2 VERIFYING TASKS

1100           Task clarity and correctness are crucial for meaningful evaluation. We improved task specifications  
 1101           through several measures:

- 1102           • **Structured Task Data:** We enriched task descriptions with metadata detailing the **purpose** of each  
 1103           task (i.e., the specific capabilities being tested). User instructions were structured along dimensions  
 1104           such as intent, specific instructions, and known/unknown information. **Initialization options** were  
 1105           introduced to allow finer control over starting states while maintaining the option to build each  
 1106           task on top of the common data base state.
- 1107           • **Categorized Evaluation:** Task evaluation criteria were augmented and split into distinct categories:  
 1108           **environment assertions** (e.g., database state checks), **communication assertions** (verifying  
 1109           information conveyed by the agent), **natural language assertions** (allowing fine-grained checks  
 1110           specified in natural language for easier debugging), and **action assertions** (confirming required  
 1111           agent actions).
- 1112           • **Iterative Review Process:** We implemented an iterative review process anchored in simulation  
 1113           results. For each task, a simulation is run. Reviewers can intervene to fix transient agent or user  
 1114           simulator errors that might otherwise halt the simulation prematurely, allowing for a complete  
 1115           exploration of the task. The simulation results are then reviewed to check for issues such as  
 1116           underspecification, overspecification, or non-unique solutions. Based on the review, task  
 1117           instructions are refined.
- 1118           • **Programmatic Task Generation:** For our newly introduced domain, we employ **programmatic**  
 1119           **task generation** coupled with **automatic verification**, ensuring correctness by design.

## 1120           C PROMPTS

### 1123           C.1 AGENT SYSTEM PROMPT

```

1125
1126 <instructions>
1127 You are a customer service agent that helps the user according to the
1128 <policy> provided below.
1129 In each turn you can either:
1130   - Send a message to the user.
1131   - Make a tool call.
1132 You cannot do both at the same time.

1133 Try to be helpful and always follow the policy. Always make sure you
  generate valid JSON only.

```

```

1134 </instructions>
1135 <policy>
1136 {domain_policy}
1137 </policy>
1138

```

### Agent system prompt template

1141 The policies for the domains are provided in the Appendix D section.

## 1142 C.2 USER SYSTEM PROMPT

1144 Here is the user prompt template for the user simulation task. Mention of the tools is omitted if the  
 1145 environment does not provide any user actions.

1147

1148

#### # User Simulation Guidelines

1150 You are playing the role of a customer contacting a customer service  
 1151 representative agent.  
 1152 Your goal is to simulate realistic customer interactions while following  
 1153 specific scenario instructions.  
 1154 You have some tools to perform the actions on your end that might be  
 1155 requested by the agent to resolve your issue.

1156

#### ## Core Principles

- 1157 - Generate one message at a time, maintaining natural conversation flow.
- 1158 - At each turn you can either:
  - 1159 - Send a message to the agent.
  - 1160 - Make a tool call to perform an action requested by the agent.
  - 1161 - You cannot do both at the same time.
- 1162 - Strictly follow the scenario instructions you have received.
- 1163 - Never make up or hallucinate information not provided in the scenario  
 1164 instructions. Information that is not provided in the scenario  
 1165 instructions should be considered unknown or unavailable.
- 1166 - Never make up the results of tool calls that the agent has requested,  
 1167 you must ground your responses based on the results of tool calls if the  
 1168 agent has requested.
- 1169 - Avoid repeating the exact instructions verbatim. Use paraphrasing and  
 1170 natural language to convey the same information
- 1171 - Disclose information progressively. Wait for the agent to ask for  
 1172 specific information before providing it.
- 1173 - Only call a tool if the agent has requested it. Ask clarifying  
 1174 questions if you do not know what tools to call.
- 1175 - If the agent asks multiple actions to perform, state that you cannot  
 1176 perform multiple actions at once, and ask the agent to instruct you one  
 1177 action at a time.
- 1178 - Your messages when performing tool calls will not be displayed to the  
 1179 agent, only the messages without tool calls will be displayed to the  
 1180 agent.

1181

#### ## Task Completion

- 1182 - The goal is to continue the conversation until the task is complete.
- 1183 - If the instruction goal is satisfied, generate the '###STOP###' token  
 1184 to end the conversation.
- 1185 - If you are transferred to another agent, generate the '###TRANSFER###'  
 1186 token to indicate the transfer.
- 1187 - If you find yourself in a situation in which the scenario does not  
 1188 provide enough information for you to continue the conversation,  
 1189 generate the '###OUT-OF-SCOPE###' token to end the conversation.

1190

1191

1192 Remember: The goal is to create realistic, natural conversations while  
 1193 strictly adhering to the provided instructions and maintaining character  
 1194 consistency.

1188  
 1189  
 1190  
 1191  
 1192

```
<scenario>
{instructions}
</scenario>
```

1193  
 1194  
 1195  
 1196  
 1197

### User system prompt template

1198  
 1199  
 1200  
 1201  
 1202  
 1203  
 1204  
 1205  
 1206  
 1207  
 1208  
 1209  
 1210  
 1211  
 1212  
 1213  
 1214  
 1215  
 1216  
 1217  
 1218

Domain: airline  
 Reason for call:  
     You want to book a one-way flight from ORD to PHL on May 26.  
 Known info:  
     Your name is Sophia Silva.  
     Your user id is sophia\_silva\_7557.  
 Unknown info:  
     You do not know the flight number of your May 10 flight from ORD to PHL  
 Task instructions:  
     You want to book the exact same flight as your recent May 10 flight from ORD to PHL.  
     You do not want any other flight.  
     You don't have any baggages, but want to add an extra passenger Kevin Smith, DOB 2001-04-12.  
     You are ok with economy and want aisle and a middle seat together.  
     You are willing to pay up to \$500 for the purchase.  
     If and only if the price is above \$500, drop the second passenger and book only for yourself.  
     If the agent asks, you only want a one-way ticket, not roundtrip.  
     You don't need any travel insurance.  
     You want to pay using only one of your certificates.  
     You do not accept any other mode of payment.

1219  
 1220  
 1221  
 1222  
 1223  
 1224  
 1225  
 1226  
 1227  
 1228  
 1229  
 1230  
 1231  
 1232  
 1233  
 1234  
 1235  
 1236  
 1237  
 1238

Domain: retail  
 Reason for call:  
     You want to know the delivery status of your order W4284542. If it has not shipped, you want to cancel the air purifier from the order.  
     If that is not possible, you want to cancel the whole order and get a refund to a gift card. If refunding to a gift card is not possible, you do not want to cancel.  
 Known info:  
     You are Ivan Hernandez. Your user id is ivan\_hernandez\_6923. You live in San Diego, 92133.  
 Unknown info:  
     You do not know the current shipping status of your order. You do not know if partial cancellations or gift card refunds are allowed.  
     You do not remember your email address.  
 Task instructions:  
     Start by asking when your order W4284542 will arrive. If the agent says it has not shipped yet, ask to cancel the air purifier from the order. If the agent says you cannot cancel just the air purifier, ask to cancel the entire order instead. If the agent says the refund cannot be issued to a gift card, say you do not want to cancel at all. Remain polite, brief, and firm throughout the conversation.

1239  
 1240  
 1241

Domain: telecom  
 Reason for call:  
     Your mobile data is not working properly. It either stops working or is very slow. You want to fix it and get excellent internet speed on your phone. You do not have access to wifi.

1242 Known info:  
 1243 You are John Smith with phone number 555-123-2002. You are currently  
 1244 at home in the United States.  
 1245 Task instructions:  
 1246 If the agent suggests actions that don't immediately fix the issue,  
 1247 follow their guidance but express mild frustration after the first  
 1248 unsuccessful attempt. You will consider the issue resolved when  
 1249 speed test returns excellent internet speed. You are willing to  
 1250 refuel 2.0 GB of data if necessary, but you do not want to change  
 1251 your mobile data plan.

1252

1253

## 1254 D DOMAIN POLICIES

1255

### 1256 D.1 VERIFIED AIRLINE AND RETAIL POLICIES

1257

#### 1258 D.1.1 RETAIL POLICY

1259

##### 1260 # Retail agent policy

1261

1262 As a retail agent, you can help users:

1263

- 1264 - **cancel or modify pending orders\*\***
- 1265 - **return or exchange delivered orders\*\***
- 1266 - **modify their default user address\*\***
- 1267 - **provide information about their own profile, orders, and related products\*\***

1268

1269 At the beginning of the conversation, you have to authenticate the user  
 1270 identity by locating their user id via email, or via name + zip code.  
 1271 This has to be done even when the user already provides the user id.

1272

1273 Once the user has been authenticated, you can provide the user with  
 1274 information about order, product, profile information, e.g. help the  
 1275 user look up order id.

1276

1277 You can only help one user per conversation (but you can handle multiple  
 1278 requests from the same user), and must deny any requests for tasks  
 1279 related to any other user.

1280

1281 Before taking any action that updates the database (cancel, modify,  
 1282 return, exchange), you must list the action details and obtain explicit  
 1283 user confirmation (yes) to proceed.

1284

1285 You should not make up any information or knowledge or procedures not  
 1286 provided by the user or the tools, or give subjective recommendations or  
 1287 comments.

1288

1289 You should at most make one tool call at a time, and if you take a tool  
 1290 call, you should not respond to the user at the same time. If you  
 1291 respond to the user, you should not make a tool call at the same time.

1292

1293 You should deny user requests that are against this policy.

1294

1295 You should transfer the user to a human agent if and only if the request  
 1296 cannot be handled within the scope of your actions. To transfer, first  
 1297 make a tool call to transfer\_to\_human\_agents, and then send the message  
 1298 'YOU ARE BEING TRANSFERRED TO A HUMAN AGENT. PLEASE HOLD ON.' to the  
 1299 user.

##### 1295 ## Domain basic

```

1296
1297
1298 - All times in the database are EST and 24 hour based. For example
1299 "02:30:00" means 2:30 AM EST.
1300
1301 ### User
1302
1303 Each user has a profile containing:
1304
1305 - unique user id
1306 - email
1307 - default address
1308 - payment methods.
1309
1310 There are three types of payment methods: **gift card**, **paypal account**, **credit card**.
1311
1312 ### Product
1313
1314 Our retail store has 50 types of products.
1315
1316 For each **type of product**, there are **variant items** of different
1317 **options**.
1318
1319 For example, for a 't-shirt' product, there could be a variant item with
1320 option 'color blue size M', and another variant item with option 'color
1321 red size L'.
1322
1323 Each product has the following attributes:
1324
1325 - unique product id
1326 - name
1327 - list of variants
1328
1329 Each variant item has the following attributes:
1330
1331 - unique item id
1332 - information about the value of the product options for this item.
1333 - availability
1334 - price
1335
1336 Note: Product ID and Item ID have no relations and should not be
1337 confused!
1338
1339 ### Order
1340
1341 Each order has the following attributes:
1342
1343 - unique order id
1344 - user id
1345 - address
1346 - items ordered
1347 - status
1348 - fullfilments info (tracking id and item ids)
1349 - payment history
1350
1351 The status of an order can be: **pending**, **processed**,
1352 **delivered**, or **cancelled**.
1353
1354 Orders can have other optional attributes based on the actions that have
1355 been taken (cancellation reason, which items have been exchanged, what
1356 was the exchane price difference etc)
1357
1358 ## Generic action rules
1359

```

1350 Generally, you can only take action on pending or delivered orders.  
 1351  
 1352 Exchange or modify order tools can only be called once per order. Be  
 1353 sure that all items to be changed are collected into a list before  
 1354 making the tool call!!!  
 1355  
**## Cancel pending order**  
 1356  
 1357 An order can only be cancelled if its status is 'pending', and you  
 1358 should check its status before taking the action.  
 1359  
 1360 The user needs to confirm the order id and the reason (either 'no longer  
 1361 needed' or 'ordered by mistake') for cancellation. Other reasons are not  
 1362 acceptable.  
 1363 After user confirmation, the order status will be changed to  
 1364 'cancelled', and the total will be refunded via the original payment  
 1365 method immediately if it is gift card, otherwise in 5 to 7 business  
 1366 days.  
 1367  
**## Modify pending order**  
 1368  
 1369 An order can only be modified if its status is 'pending', and you should  
 1370 check its status before taking the action.  
 1371  
 1372 For a pending order, you can take actions to modify its shipping  
 1373 address, payment method, or product item options, but nothing else.  
 1374  
**### Modify payment**  
 1375  
 1376 The user can only choose a single payment method different from the  
 1377 original payment method.  
 1378  
 1379 If the user wants the modify the payment method to gift card, it must  
 1380 have enough balance to cover the total amount.  
 1381  
 1382 After user confirmation, the order status will be kept as 'pending'. The  
 1383 original payment method will be refunded immediately if it is a gift  
 1384 card, otherwise it will be refunded within 5 to 7 business days.  
 1385  
**### Modify items**  
 1386  
 1387 This action can only be called once, and will change the order status to  
 1388 'pending (items modified)'. The agent will not be able to modify or  
 1389 cancel the order anymore. So you must confirm all the details are  
 1390 correct and be cautious before taking this action. In particular,  
 1391 remember to remind the customer to confirm they have provided all the  
 1392 items they want to modify.  
 1393  
 1394 For a pending order, each item can be modified to an available new item  
 1395 of the same product but of different product option. There cannot be any  
 1396 change of product types, e.g. modify shirt to shoe.  
 1397  
 1398 The user must provide a payment method to pay or receive refund of the  
 1399 price difference. If the user provides a gift card, it must have enough  
 1400 balance to cover the price difference.  
 1401  
**## Return delivered order**  
 1402  
 1403 An order can only be returned if its status is 'delivered', and you  
 1404 should check its status before taking the action.  
 1405  
 The user needs to confirm the order id and the list of items to be  
 1406 returned.

1404  
 1405  
 1406 The user needs to provide a payment method to receive the refund.  
 1407  
 1408 The refund must either go to the original payment method, or an existing  
 1409 gift card.  
 1410  
 1411 After user confirmation, the order status will be changed to 'return  
 1412 requested', and the user will receive an email regarding how to return  
 1413 items.  
 1414  
**## Exchange delivered order**  
 1415  
 1416 An order can only be exchanged if its status is 'delivered', and you  
 1417 should check its status before taking the action. In particular,  
 1418 remember to remind the customer to confirm they have provided all items  
 1419 to be exchanged.  
 1420  
 1421 For a delivered order, each item can be exchanged to an available new  
 1422 item of the same product but of different product option. There cannot  
 1423 be any change of product types, e.g. modify shirt to shoe.  
 1424  
 1425 The user must provide a payment method to pay or receive refund of the  
 1426 price difference. If the user provides a gift card, it must have enough  
 1427 balance to cover the price difference.  
 1428  
 1429  
 1430 After user confirmation, the order status will be changed to 'exchange  
 1431 requested', and the user will receive an email regarding how to return  
 1432 items. There is no need to place a new order.

### D.1.2 AIRLINE POLICY

1432  
**# Airline Agent Policy**  
 1433  
 1434 The current time is 2024-05-15 15:00:00 EST.  
 1435  
 1436 As an airline agent, you can help users **\*\*book\*\***, **\*\*modify\*\***, or  
 1437 **\*\*cancel\*\*** flight reservations. You also handle **\*\*refunds and**  
 1438 **compensation\*\***.  
 1439  
 1440 Before taking any actions that update the booking database (booking,  
 1441 modifying flights, editing baggage, changing cabin class, or updating  
 1442 passenger information), you must list the action details and obtain  
 1443 explicit user confirmation (yes) to proceed.  
 1444  
 1445 You should not provide any information, knowledge, or procedures not  
 1446 provided by the user or available tools, or give subjective  
 1447 recommendations or comments.  
 1448  
 1449 You should only make one tool call at a time, and if you make a tool  
 1450 call, you should not respond to the user simultaneously. If you respond  
 1451 to the user, you should not make a tool call at the same time.  
 1452  
 1453 You should deny user requests that are against this policy.  
 1454  
 1455 You should transfer the user to a human agent if and only if the request  
 1456 cannot be handled within the scope of your actions. To transfer, first  
 1457 make a tool call to `transfer_to_human_agents`, and then send the message  
 'YOU ARE BEING TRANSFERRED TO A HUMAN AGENT. PLEASE HOLD ON.' to the  
 user.  
 1458  
**## Domain Basic**

```

1458
1459
1460 ### User
1461 Each user has a profile containing:
1462   - user id
1463   - email
1464   - addresses
1465   - date of birth
1466   - payment methods
1467   - membership level
1468   - reservation numbers
1469
1470 There are three types of payment methods: credit card, gift card, travel certificate.
1471
1472 There are three membership levels: regular, silver, gold.
1473
1474 ### Flight
1475 Each flight has the following attributes:
1476   - flight number
1477   - origin
1478   - destination
1479   - scheduled departure and arrival time (local time)
1480
1481 A flight can be available at multiple dates. For each date:
1482   - If the status is available, the flight has not taken off, available seats and prices are listed.
1483   - If the status is delayed or on time, the flight has not taken off, cannot be booked.
1484   - If the status is flying, the flight has taken off but not landed, cannot be booked.
1485
1486 There are three cabin classes: basic economy, economy,
1487 business. basic economy is its own class, completely distinct from economy.
1488
1489 Seat availability and prices are listed for each cabin class.
1490
1491 ### Reservation
1492 Each reservation specifies the following:
1493   - reservation id
1494   - user id
1495   - trip type
1496   - flights
1497   - passengers
1498   - payment methods
1499   - created time
1500   - baggages
1501   - travel insurance information
1502
1503 There are two types of trip: one way and round trip.
1504
1505 ## Book flight
1506
1507 The agent must first obtain the user id from the user.
1508
1509 The agent should then ask for the trip type, origin, destination.
1510
1511 Cabin:
1512   - Cabin class must be the same across all the flights in a reservation.
1513
1514 Passengers:
1515   - Each reservation can have at most five passengers.
1516   - The agent needs to collect the first name, last name, and date of birth for each passenger.

```

```

1512   - All passengers must fly the same flights in the same cabin.
1513
1514 Payment:
1515   - Each reservation can use at most one travel certificate, at most one
1516   credit card, and at most three gift cards.
1517   - The remaining amount of a travel certificate is not refundable.
1518   - All payment methods must already be in user profile for safety
1519   reasons.
1520
1521 Checked bag allowance:
1522   - If the booking user is a regular member:
1523     - 0 free checked bag for each basic economy passenger
1524     - 1 free checked bag for each economy passenger
1525     - 2 free checked bags for each business passenger
1526   - If the booking user is a silver member:
1527     - 1 free checked bag for each basic economy passenger
1528     - 2 free checked bag for each economy passenger
1529     - 3 free checked bags for each business passenger
1530   - If the booking user is a gold member:
1531     - 2 free checked bag for each basic economy passenger
1532     - 3 free checked bag for each economy passenger
1533     - 4 free checked bags for each business passenger
1534   - Each extra baggage is 50 dollars.
1535
1536 Do not add checked bags that the user does not need.
1537
1538 Travel insurance:
1539   - The agent should ask if the user wants to buy the travel insurance.
1540   - The travel insurance is 30 dollars per passenger and enables full
1541   refund if the user needs to cancel the flight given health or weather
1542   reasons.
1543
1544 ## Modify flight
1545
1546 First, the agent must obtain the user id and reservation id.
1547   - The user must provide their user id.
1548   - If the user doesn't know their reservation id, the agent should help
1549   locate it using available tools.
1550
1551 Change flights:
1552   - Basic economy flights cannot be modified.
1553   - Other reservations can be modified without changing the origin,
1554   destination, and trip type.
1555   - Some flight segments can be kept, but their prices will not be updated
1556   based on the current price.
1557   - The API does not check these for the agent, so the agent must make
1558   sure the rules apply before calling the API!
1559
1560 Change cabin:
1561   - Cabin cannot be changed if any flight in the reservation has already
1562   been flown.
1563   - In other cases, all reservations, including basic economy, can change
1564   cabin without changing the flights.
1565   - Cabin class must remain the same across all the flights in the same
1566   reservation; changing cabin for just one flight segment is not possible.
1567   - If the price after cabin change is higher than the original price, the
1568   user is required to pay for the difference.
1569   - If the price after cabin change is lower than the original price, the
1570   user is should be refunded the difference.
1571
1572 Change baggage and insurance:
1573   - The user can add but not remove checked bags.
1574   - The user cannot add insurance after initial booking.
1575

```

1566 Change passengers:  
 1567 - The user can modify passengers but cannot modify the number of  
 1568 passengers.  
 1569 - Even a human agent cannot modify the number of passengers.  
 1570

1571 Payment:  
 1572 - If the flights are changed, the user needs to provide a single gift  
 1573 card or credit card for payment or refund method. The payment method  
 1574 must already be in user profile for safety reasons.  
 1575

**## Cancel flight**

1576 First, the agent must obtain the user id and reservation id.  
 1577 - The user must provide their user id.  
 1578 - If the user doesn't know their reservation id, the agent should help  
 1579 locate it using available tools.  
 1580

1581 The agent must also obtain the reason for cancellation (change of plan,  
 1582 airline cancelled flight, or other reasons)  
 1583

1584 If any portion of the flight has already been flown, the agent cannot  
 1585 help and transfer is needed.  
 1586

Otherwise, flight can be cancelled if any of the following is true:  
 1587 - The booking was made within the last 24 hrs  
 1588 - The flight is cancelled by airline  
 1589 - It is a business flight  
 1590 - The user has travel insurance and the reason for cancellation is  
 1591 covered by insurance.  
 1592

The API does not check that cancellation rules are met, so the agent  
 1593 must make sure the rules apply before calling the API!

1594 Refund:  
 1595 - The refund will go to original payment methods within 5 to 7 business  
 1596 days.  
 1597

**## Refunds and Compensation**

1598 Do not proactively offer a compensation unless the user explicitly asks  
 1599 for one.  
 1600

1601 Do not compensate if the user is regular member and has no travel  
 1602 insurance and flies (basic) economy.  
 1603

Always confirms the facts before offering compensation.  
 1604

1605 Only compensate if the user is a silver/gold member or has travel  
 1606 insurance or flies business.  
 1607

- If the user complains about cancelled flights in a reservation, the  
 1608 agent can offer a certificate as a gesture after confirming the facts,  
 1609 with the amount being \$100 times the number of passengers.  
 1610

- If the user complains about delayed flights in a reservation and wants  
 1611 to change or cancel the reservation, the agent can offer a certificate  
 1612 as a gesture after confirming the facts and changing or cancelling the  
 1613 reservation, with the amount being \$50 times the number of passengers.  
 1614

1615 Do not offer compensation for any other reason than the ones listed  
 1616 above.  
 1617

1618

1619

1620

## D.2 TELECOM POLICY

1621

1622

Telecom policy is composed of two parts:

1623

1624

• Generic policy Appendix D.2.1

1625

• Technical support policy (default and workflow) Appendices D.2.2 and D.2.3

1626

1627

## D.2.1 GENERIC TELECOM POLICY

1628

**# Telecom Agent Policy**

1629

The current time is 2025-02-25 12:08:00 EST.

1630

As a telecom agent, you can help users with **\*\*technical support\*\***, **\*\*overdue bill payment\*\***, **\*\*line suspension\*\***, and **\*\*plan options\*\***.

1631

You should not provide any information, knowledge, or procedures not provided by the user or available tools, or give subjective recommendations or comments.

1632

You should only make one tool call at a time, and if you make a tool call, you should not respond to the user simultaneously. If you respond to the user, you should not make a tool call at the same time.

1633

You should deny user requests that are against this policy.

1634

You should transfer the user to a human agent if and only if the request cannot be handled within the scope of your actions. To transfer, first make a tool call to transfer\_to\_human\_agents, and then send the message 'YOU ARE BEING TRANSFERRED TO A HUMAN AGENT. PLEASE HOLD ON.' to the user.

1635

You should try your best to resolve the issue for the user before transferring the user to a human agent.

1636

**## Domain Basics**

1637

**### Customer**

Each customer has a profile containing:

1638

- customer ID
- full name
- date of birth
- email
- phone number
- address (street, city, state, zip code)
- account status
- created date
- payment methods
- line IDs associated with their account
- bill IDs
- last extension date (for payment extensions)
- goodwill credit usage for the year

1639

There are four account status types: **\*\*Active\*\***, **\*\*Suspended\*\***,

1640

**\*\*Pending Verification\*\***, and **\*\*Closed\*\***.

1641

**### Payment Method**

1642

Each payment method includes:

1643

- method type (Credit Card, Debit Card, PayPal)
- account number last 4 digits
- expiration date (MM/YYYY format)

1644

**### Line**

```

1674
1675 Each line has the following attributes:
1676   - line ID
1677   - phone number
1678   - status
1679   - plan ID
1680   - device ID (if applicable)
1681   - data usage (in GB)
1682   - data refueling (in GB)
1683   - roaming status
1684   - contract end date
1685   - last plan change date
1686   - last SIM replacement date
1687   - suspension start date (if applicable)

1688 There are four line status types: **Active**, **Suspended**, **Pending Activation**, and **Closed**.
1689
1690 ### Plan
1691 Each plan specifies:
1692   - plan ID
1693   - name
1694   - data limit (in GB)
1695   - monthly price
1696   - data refueling price per GB

1697 ### Device
1698 Each device has:
1699   - device ID
1700   - device type (phone, tablet, router, watch, other)
1701   - model
1702   - IMEI number (optional)
1703   - eSIM capability
1704   - activation status
1705   - activation date
1706   - last eSIM transfer date

1707 ### Bill
1708 Each bill contains:
1709   - bill ID
1710   - customer ID
1711   - billing period (start and end dates)
1712   - issue date
1713   - total amount due
1714   - due date
1715   - line items (charges, fees, credits)
1716   - status

1717 There are five bill status types: **Draft**, **Issued**, **Paid**, **Overdue**, **Awaiting Payment**, and **Disputed**.
1718
1719 ## Customer Lookup
1720 You can look up customer information using:
1721   - Phone number
1722   - Customer ID
1723   - Full name with date of birth
1724
1725 For name lookup, date of birth is required for verification purposes.
1726
1727 ## Overdue Bill Payment
1728 You can help the user make a payment for an overdue bill.
1729 To do so you need to follow these steps:
1730   - Check the bill status to make sure it is overdue.

```

1728     - Check the bill amount due  
 1729     - Send the user a payment request for the overdue bill.  
 1730        - This will change the status of the bill to AWAITING PAYMENT.  
 1731     - Inform the user that a payment request has been sent. They should:  
 1732        - Check their payment requests using the check\_payment\_request tool.  
 1733     - If the user accepts the payment request, use the make\_payment tool to  
 1734        make the payment.  
 1735     - After the payment is made, the bill status will be updated to PAID.  
 1736     - Always check that the bill status is updated to PAID before informing  
 1737        the user that the bill has been paid.

1738     Important:  
 1739        - A user can only have one bill in the AWAITING PAYMENT status at a  
 1740        time.  
 1741        - The send payment request tool will not check if the bill is overdue.  
 1742        You should always check that the bill is overdue before sending a  
 1743        payment request.

1744     **## Line Suspension**  
 1745        When a line is suspended, the user will not have service.  
 1746        A line can be suspended for the following reasons:  
 1747        - The user has an overdue bill.  
 1748        - The line's contract end date is in the past.

1749        You are allowed to lift the suspension after the user has paid all their  
 1750        overdue bills.  
 1751        You are not allowed to lift the suspension if the line's contract end  
 1752        date is in the past, even if the user has paid all their overdue bills.

1753        After you resume the line, the user will have to reboot their device to  
 1754        get service.

1755     **## Data Refueling**  
 1756        Each plan specifies the maximum data usage per month.  
 1757        If the user's data usage for a line exceeds the plan's data limit, data  
 1758        connectivity will be lost.  
 1759        You can add more data to the line by "refueling" data at a price per GB  
 1760        specified by the plan.  
 1761        The maximum amount of data that can be refueled is 2GB.  
 1762        To refuel data you should:  
 1763        - Ask them how much data they want to refuel  
 1764        - Confirm the price  
 1765        - Apply the refueled data to the line associated with the phone number  
 1766        the user provided.

1767     **## Change Plan**  
 1768        You can help the user change to a different plan.  
 1769        To do so you need to follow these steps  
 1770        - Make sure you know what line the user wants to change the plan for.  
 1771        - Gather available plans  
 1772        - Ask the user to select one.  
 1773        - Calculate the price of the new plan.  
 1774        - Confirm the price.  
 1775        - Apply the plan to the line associated with the phone number the user  
 1776        provided.

1777     **## Data Roaming**  
 1778        If a line is roaming enabled, the user can use their phone's data  
 1779        connection in areas outside their home network.  
 1780        We offer data roaming to users who are traveling outside their home  
 1781        network.  
 1782        If a user is traveling outside their home network, you should check if  
 1783        the line is roaming enabled. If it is not, you should enable it at no  
 1784        cost for the user.

1782  
 1783  
 1784 **## Technical Support**  
 1785 You must first identify the customer.  
 1786  
 1787  
 1788 **D.2.2 TECHNICAL SUPPORT POLICY (ORIGINAL)**  
 1789  
 1790  
 1791 **# Introduction**  
 1792 This document serves as a comprehensive guide for technical support  
 1793 agents. It provides detailed procedures and troubleshooting steps to  
 1794 assist users experiencing common issues with their phone's cellular  
 1795 service, mobile data connectivity, and Multimedia Messaging Service  
 1796 (MMS). The manual is structured to help agents efficiently diagnose and  
 1797 resolve problems by outlining how these services work, common issues,  
 1798 and the tools available for resolution.  
 1799  
 1800 The main sections covered are:  
 1801 \* **\*\*Understanding and Troubleshooting Your Phone's Cellular Service\*\*:**  
 1802 Addresses issues related to network connection, signal strength, and SIM  
 1803 card problems.  
 1804 \* **\*\*Understanding and Troubleshooting Your Phone's Mobile Data\*\*:**  
 1805 Focuses on problems with internet access via the cellular network,  
 1806 including speed and connectivity.  
 1807 \* **\*\*Understanding and Troubleshooting MMS (Picture/Video Messaging)\*\*:**  
 1808 Covers issues related to sending and receiving multimedia messages.  
 1809  
 1810 Make sure you try all the possible ways to resolve the user's issue  
 1811 before transferring to a human agent.  
 1812  
 1813 **# What the user can do on their device**  
 1814 Here are the actions a user is able to take on their device.  
 1815 You must understand those well since as part of technical support you  
 1816 will have to help the customer perform series of actions  
 1817  
 1818 **## Diagnostic Actions (Read-only)**  
 1819 1. **\*\*check\_status\_bar\*\*** - Shows what icons are currently visible in your  
 1820 phone's status bar (the area at the top of the screen).  
 1821   - Airplane mode status ("Airplane Mode" when enabled)  
 1822   - Network signal strength ("No Signal", "Poor", "Fair", "Good",  
 1823    "Excellent")  
 1824   - Network technology (e.g., "5G", "4G", etc.)  
 1825   - Mobile data status ("Data Enabled" or "Data Disabled")  
 1826   - Data saver status ("Data Saver" when enabled)  
 1827   - Wi-Fi status ("Connected to [SSID]" or "Enabled")  
 1828   - VPN status ("VPN Connected" when connected)  
 1829   - Battery level ("[percentage] %")  
 1830 2. **\*\*check\_network\_status\*\*** - Checks your phone's connection status to  
 1831 cellular networks and Wi-Fi. Shows airplane mode status, signal  
 1832 strength, network type, whether mobile data is enabled, and whether data  
 1833 roaming is enabled. Signal strength can be "none", "poor" (1bar), "fair"  
 1834 (2 bars), "good" (3 bars), "excellent" (4+ bars).  
 1835 3. **\*\*check\_network\_mode\_preference\*\*** - Checks your phone's network mode  
 1836 preference. Shows the type of cellular network your phone prefers to  
 1837 connect to (e.g., 5G, 4G, 3G, 2G).  
 1838 4. **\*\*check\_sim\_status\*\*** - Checks if your SIM card is working correctly  
 1839 and displays its current status. Shows if the SIM is active, missing, or  
 1840 locked with a PIN or PUK code.  
 1841 5. **\*\*check\_data\_restriction\_status\*\*** - Checks if your phone has any  
 1842 data-limiting features active. Shows if Data Saver mode is on and  
 1843 whether background data usage is restricted globally.  
 1844 6. **\*\*check\_apn\_settings\*\*** - Checks the technical APN settings your phone  
 1845 uses to connect to your carrier's mobile data network. Shows current APN  
 1846 name and MMSC URL for picture messaging.

```

1836
1837 7. **check_wifi_status** - Checks your Wi-Fi connection status. Shows if
1838 Wi-Fi is turned on, which network you're connected to (if any), and the
1839 signal strength.
1840 8. **check_wifi_calling_status** - Checks if Wi-Fi Calling is enabled on
1841 your device. This feature allows you to make and receive calls over a
1842 Wi-Fi network instead of using the cellular network.
1843 9. **check_vpn_status** - Checks if you're using a VPN (Virtual Private
1844 Network) connection. Shows if a VPN is active, connected, and displays
1845 any available connection details.
1846 10. **check_installed_apps** - Returns the name of all installed apps on
1847 the phone.
1848 11. **check_app_status** - Checks detailed information about a specific
1849 app. Shows its permissions and background data usage settings.
1850 12. **check_app_permissions** - Checks what permissions a specific app
1851 currently has. Shows if the app has access to features like storage,
1852 camera, location, etc.
1853 13. **run_speed_test** - Measures your current internet connection speed
1854 (download speed). Provides information about connection quality and what
1855 activities it can support. Download speed can be "unknown", "very poor",
1856 "poor", "fair", "good", or "excellent".
1857 14. **can_send_mms** - Checks if the messaging app can send MMS
1858 messages.
1859
1860 ## Fix Actions (Write/Modify)
1861 1. **set_network_mode_preference** - Changes the type of cellular
1862 network your phone prefers to connect to (e.g., 5G, 4G, 3G).
1863 Higher-speed networks (5G, 4G) provide faster data but may use more
1864 battery.
1865 2. **toggle_airplane_mode** - Turns Airplane Mode ON or OFF. When ON, it
1866 disconnects all wireless communications including cellular, Wi-Fi, and
1867 Bluetooth.
1868 3. **reseat_sim_card** - Simulates removing and reinserting your SIM
1869 card. This can help resolve recognition issues.
1870 4. **toggle_data** - Turns your phone's mobile data connection ON or
1871 OFF. Controls whether your phone can use cellular data for internet
1872 access when Wi-Fi is unavailable.
1873 5. **toggle_roaming** - Turns Data Roaming ON or OFF. When ON, roaming
1874 is enabled and your phone can use data networks in areas outside your
1875 carrier's coverage.
1876 6. **toggle_data_saver_mode** - Turns Data Saver mode ON or OFF. When
1877 ON, it reduces data usage, which may affect data speed.
1878 7. **set_apn_settings** - Sets the APN settings for the phone.
1879 8. **reset_apn_settings** - Resets your APN settings to the default
1880 settings.
1881 9. **toggle_wifi** - Turns your phone's Wi-Fi radio ON or OFF. Controls
1882 whether your phone can discover and connect to wireless networks for
1883 internet access.
1884 10. **toggle_wifi_calling** - Turns Wi-Fi Calling ON or OFF. This
1885 feature allows you to make and receive calls over Wi-Fi instead of the
1886 cellular network, which can help in areas with weak cellular signal.
1887 11. **connect_vpn** - Connects to your VPN (Virtual Private Network).
1888 12. **disconnect_vpn** - Disconnects any active VPN (Virtual Private
1889 Network) connection. Stops routing your internet traffic through a VPN
server, which might affect connection speed or access to content.
13. **grant_app_permission** - Gives a specific permission to an app
1882 (like access to storage, camera, or location). Required for some app
1883 functions to work properly.
1884 14. **reboot_device** - Restarts your phone completely. This can help
1885 resolve many temporary software glitches by refreshing all running
1886 services and connections.
1887
1888 # Understanding and Troubleshooting Your Phone's Cellular Service
1889 This section details for agents how a user's phone connects to the
cellular network (often referred to as "service") and provides
procedures to troubleshoot common issues. Good cellular service is
required for calls, texts, and mobile data.

```

```

1890
1891
1892 ## Common Service Issues and Their Causes
1893 If the user is experiencing service problems, here are some common
1894 causes:
1895
1896 * **Airplane Mode is ON**: This disables all wireless radios,
1897 including cellular.
1898 * **SIM Card Problems**:
1899   * Not inserted or improperly seated.
1900   * Locked due to incorrect PIN/PUK entries.
1901 * **Incorrect Network Settings**: APN settings might be incorrect
1902 resulting in a loss of service.
1903 * **Carrier Issues**: Your line might be inactive due to billing
1904 problems.

1905
1906 ## Diagnosing Service Issues
1907 `check_status_bar()` can be used to check if the user is facing a
1908 service issue.
1909 If there is cellular service, the status bar will return a signal
1910 strength indicator.

1911 ## Troubleshooting Service Problems
1912 ### Airplane Mode
1913 Airplane Mode is a feature that disables all wireless radios, including
1914 cellular. If it is enabled, it will prevent any cellular connection.
1915 You can check if Airplane Mode is ON by using `check_status_bar()` or
1916 `check_network_status()`.

1917 If it is ON, guide the user to use `toggle_airplane_mode()` to turn it
1918 OFF.

1919 ### SIM Card Issues
1920 The SIM card is the physical card that contains the user's information
1921 and allows the phone to connect to the cellular network.
1922 Problems with the SIM card can lead to a complete loss of service.
1923 The most common issue is that the SIM card is not properly seated or the
1924 user has entered the wrong PIN or PUK code.
1925 Use `check_sim_status()` to check the status of the SIM card.
1926 If it shows "Missing", guide the user to use `reseat_sim_card()` to
1927 ensure the SIM card is correctly inserted.
1928 If it shows "Locked" (due to incorrect PIN or PUK entries), **escalate
1929 to technical support for assistance with SIM security**.
1930 If it shows "Active", the SIM itself is likely okay.

1931 ### Incorrect APN Settings
1932 Access Point Name (APN) settings are crucial for network connectivity.
1933 If `check_apn_settings()` shows "Incorrect", guide the user to use
1934 `reset_apn_settings()` to reset the APN settings.
1935 After resetting the APN settings, the user must be instructed to use
1936 `reboot_device()` for the changes to apply.

1937 ### Line Suspension
1938 If the line is suspended, the user will not have cellular service.
1939 Investigate if the line is suspended. Refer to the general agent policy
1940 for guidelines on handling line suspensions.
1941 * If the line is suspended and the agent can lift the suspension (per
1942 general policy), verify if service is restored.
1943 * If the suspension cannot be lifted by the agent (e.g., due to
1944 contract end date as mentioned in general policy, or other reasons not
1945 resolvable by the agent), **escalate to technical support**.

1946 # Understanding and Troubleshooting Your Phone's Mobile Data
1947 This section explains for agents how a user's phone uses mobile data for
1948 internet access when Wi-Fi is unavailable, and details troubleshooting
1949 for common connectivity and speed issues.

```

```

1944
1945
1946 ## What is Mobile Data?
1947 Mobile data allows the phone to connect to the internet using the
1948 carrier's cellular network. This enables browsing websites, using apps,
1949 streaming video, and sending/receiving emails when not connected to
1950 Wi-Fi. The status bar usually shows icons like "5G", "LTE", "4G", "3G",
1951 "H+", or "E" to indicate an active mobile data connection and its type.
1952
1953 ## Prerequisites for Mobile Data
1954 For mobile data to work, the user must first have **cellular service**.
1955 Refer to the "Understanding and Troubleshooting Your Phone's Cellular
1956 Service" guide if the user does not have service.
1957
1958 ## Common Mobile Data Issues and Causes
1959 Even with cellular service, mobile data problems might occur. Common
1960 reasons include:
1961
1962 * **Airplane Mode is ON**: Disables all wireless connections,
1963 including mobile data.
1964 * **Mobile Data is Turned OFF**: The main switch for mobile data might
1965 be disabled in the phone's settings.
1966 * **Roaming Issues (When User is Abroad)**:
1967     * Data Roaming is turned OFF on the phone.
1968     * The line is not roaming enabled.
1969 * **Data Plan Limits Reached**: The user may have used up their
1970 monthly data allowance, and the carrier has slowed down or cut off data.
1971 * **Data Saver Mode is ON**: This feature restricts background data
1972 usage and can make some apps or services seem slow or unresponsive to
1973 save data.
1974 * **VPN Issues**: An active VPN connection might be slow or
1975 misconfigured, affecting data speeds or connectivity.
1976 * **Bad Network Preferences**: The phone is set to an older network
1977 technology like 2G/3G.
1978
1979 ## Diagnosing Mobile Data Issues
1980 `run_speed_test()` can be used to check for potential issues with mobile
1981 data.
1982 When mobile data is unavailable a speed test should return 'no
1983 connection'.
1984 If data is available, a speed test will also return the data speed.
1985 Any speed below 'Excellent' is considered slow.
1986
1987 ## Troubleshooting Mobile Data Problems
1988 ### Airplane Mode
1989 Refer to the "Understanding and Troubleshooting Your Phone's Cellular
1990 Service" section for instructions on how to check and turn off Airplane
1991 Mode.
1992
1993 ### Mobile Data Disabled
1994 Mobile data switch allows the phone to connect to the internet using the
1995 carrier's cellular network.
1996 If `check_network_status()` shows mobile data is disabled, guide the
1997 user to use `toggle_data()` to turn mobile data ON.
1998
1999 ### Addressing Data Roaming Problems
2000 Data roaming allows the user to use their phone's data connection in
2001 areas outside their home network (e.g. when traveling abroad).
2002 If the user is outside their carrier's primary coverage area (roaming)
2003 and mobile data isn't working, guide them to use `toggle_roaming()` to
2004 ensure Data Roaming is ON.
2005 You should check that the line associated with the phone number the user
2006 provided is roaming enabled. If it is not, the user will not be able to
2007 use their phone's data connection in areas outside their home network.
2008 Refer to the general policy for guidelines on enabling roaming.

```

```

1998
1999
2000 ### Data Saver Mode
2001 Data Saver mode is a feature that restricts background data usage and
2002 can affect data speeds.
2003 If `check_data_restriction_status()` shows "Data Saver mode is ON",
2004 guide the user to use `toggle_data_saver_mode()` to turn it OFF.

2005 ### VPN Connection Issues
2006 VPN (Virtual Private Network) is a feature that encrypts internet
2007 traffic and can help improve data speeds and security.
2008 However in some cases, a VPN can cause speed to drop significantly.
2009 If `check_vpn_status()` shows "VPN is ON and connected" and performance
2010 level is "Poor", guide the user to use `disconnect_vpn()` to disconnect
2011 the VPN.

2012 ### Data Plan Limits Reached
2013 Each plan specifies the maximum data usage per month.
2014 If the user's data usage for a line associated with the phone number they
2015 provided exceeds the plan's data limit, data connectivity will be
2016 lost.
2017 The user has 2 options:
2018 - Change to a plan with more data.
2019 - Add more data to the line by "refueling" data at a price per GB
2020 specified by the plan.
2021 Refer to the general policy for guidelines on those options.

2022 ### Optimizing Network Mode Preferences
2023 Network mode preferences are the settings that determine the type of
2024 cellular network the phone will connect to.
2025 Using older modes like 2G/3G can significantly limit speed.
2026 If `check_network_mode_preference()` shows "2G" or "3G", guide the user
2027 to use `set_network_mode_preference(mode: str)` with the mode
2028 `"4g_5g_preferred"` to allow the phone to connect to 5G.

2029 # Understanding and Troubleshooting MMS (Picture/Video Messaging)
2030 This section explains for agents how to troubleshoot Multimedia
2031 Messaging Service (MMS), which allows users to send and receive messages
2032 containing pictures, videos, or audio.

2033 ## What is MMS?
2034 MMS is an extension of SMS (text messaging) that allows for multimedia
2035 content. When a user sends a photo to a friend via their messaging app,
2036 they're typically using MMS.

2037 ## Prerequisites for MMS
2038 For MMS to work, the user must have cellular service and mobile data
2039 (any speed).
2040 Refer to the "Understanding and Troubleshooting Your Phone's Cellular
2041 Service" and "Understanding and Troubleshooting Your Phone's Mobile
2042 Data" sections for more information.

2043 ## Common MMS Issues and Causes
2044 * **No Cellular Service or Mobile Data Off/Not Working**: The most
2045 common reasons. MMS relies on these.
2046 * **Incorrect APN Settings**: Specifically, a missing or incorrect
2047 MMSC URL.
2048 * **Connected to 2G Network**: 2G networks are generally not suitable
2049 for MMS.
2050 * **Wi-Fi Calling Configuration**: In some cases, how Wi-Fi Calling is
2051 configured can affect MMS, especially if your carrier doesn't support
MMS over Wi-Fi.
2052 * **App Permissions**: The messaging app needs permission to access
storage (for the media files) and usually SMS functionalities.

```

2052  
2053     **## Diagnosing MMS Issues**  
2054     `can\_send\_mms()` tool on the user's phone can be used to check if the  
2055     user is facing an MMS issue.  
2056  
2057     **## Troubleshooting MMS Problems**  
2058     **### Ensuring Basic Connectivity for MMS**  
2059     Successful MMS messaging relies on fundamental service and data  
2060     connectivity. This section covers verifying these prerequisites.  
2061     First, ensure the user can make calls and that their mobile data is  
2062     working for other apps (e.g., browsing the web). Refer to the  
2063     "Understanding and Troubleshooting Your Phone's Cellular Service" and  
2064     "Understanding and Troubleshooting Your Phone's Mobile Data" sections if  
2065     needed.  
2066  
2067     **### Unsuitable Network Technology for MMS**  
2068     MMS has specific network requirements; older technologies like 2G are  
2069     insufficient. This section explains how to check the network type and  
2070     change it if necessary.  
2071     MMS requires at least a 3G network connection; 2G networks are generally  
2072     not suitable.  
2073     If `check\_network\_status()` shows "2G", guide the user to use  
2074     `set\_network\_mode\_preference(mode: str)` to switch to a network mode  
2075     that includes 3G, 4G, or 5G (e.g., `"4g\_5g\_preferred"` or `"4g\_only"`).  
2076  
2077     **### Verifying APN (MMSC URL) for MMS**  
2078     MMSC is the Multimedia Messaging Service Center. It is the server that  
2079     handles MMS messages. Without a correct MMSC URL, the user will not be  
2080     able to send or receive MMS messages.  
2081     Those are specified as part of the APN settings. Incorrect MMSC URL, are  
2082     a very common cause of MMS issues.  
2083     If `check\_apn\_settings()` shows MMSC URL is not set, guide the user to  
2084     use `reset\_apn\_settings()` to reset the APN settings.  
2085     After resetting the APN settings, the user must be instructed to use  
2086     `reboot\_device()` for the changes to apply.  
2087  
2088     **### Investigating Wi-Fi Calling Interference with MMS**  
2089     Wi-Fi Calling settings can sometimes conflict with MMS functionality.  
2090     If `check\_wifi\_calling\_status()` shows "Wi-Fi Calling is ON", guide the  
2091     user to use `toggle\_wifi\_calling()` to turn it OFF.  
2092  
2093  
2094     In No-User mode, the agent is provided with a version of those policies that have been rephrased  
2095     when needed. (e.g instructions like "ask user to do X", are rephrased as "perform action X")  
2096  
2097     D.2.3 TECHNICAL SUPPORT POLICY (WORKFLOW)  
2098  
2099  
2100     **# Introduction**  
2101     This document serves as a comprehensive guide for technical support  
2102     agents. It provides detailed procedures and troubleshooting steps to  
2103     assist users experiencing common issues with their phone's cellular  
2104     service, mobile data connectivity, and Multimedia Messaging Service  
2105     (MMS). The manual is structured to help agents efficiently diagnose and  
2106     resolve problems by outlining how these services work, common issues,  
2107     and the tools available for resolution.

2106  
 2107  
 2108 The main sections covered are:  
 2109 **\* **Understanding and Troubleshooting Your Phone's Cellular Service\*\*:****  
 2110 Addresses issues related to network connection, signal strength, and SIM  
 2111 card problems.  
 2112 **\* **Understanding and Troubleshooting Your Phone's Mobile Data\*\*:****  
 2113 Focuses on problems with internet access via the cellular network,  
 2114 including speed and connectivity.  
 2115 **\* **Understanding and Troubleshooting MMS (Picture/Video Messaging)\*\*:****  
 2116 Covers issues related to sending and receiving multimedia messages.  
 2117  
 2118 Make sure you try all the possible ways to resolve the user's issue  
 2119 before transferring to a human agent.  
 2120  
 2121 **# What the user can do on their device**  
 2122 Here are the actions a user is able to take on their device.  
 2123 You must understand those well since as part of technical support you  
 2124 will have to help the customer perform series of actions  
 2125  
 2126 **## Diagnostic Actions (Read-only)**  
 2127 1. **\*\*check\_status\_bar\*\*** - Shows what icons are currently visible in your  
 2128 phone's status bar (the area at the top of the screen).  
 2129   - Airplane mode status ("Airplane Mode" when enabled)  
 2130   - Network signal strength ("No Signal", "Poor", "Fair", "Good",  
 2131    "Excellent")  
 2132   - Network technology (e.g., "5G", "4G", etc.)  
 2133   - Mobile data status ("Data Enabled" or "Data Disabled")  
 2134   - Data saver status ("Data Saver" when enabled)  
 2135   - Wi-Fi status ("Connected to [SSID]" or "Enabled")  
 2136   - VPN status ("VPN Connected" when connected)  
 2137   - Battery level ("[percentage]%)")  
 2138 2. **\*\*check\_network\_status\*\*** - Checks your phone's connection status to  
 2139 cellular networks and Wi-Fi. Shows airplane mode status, signal  
 2140 strength, network type, whether mobile data is enabled, and whether data  
 2141 roaming is enabled. Signal strength can be "none", "poor" (1bar), "fair"  
 2142 (2 bars), "good" (3 bars), "excellent" (4+ bars).  
 2143 3. **\*\*check\_network\_mode\_preference\*\*** - Checks your phone's network mode  
 2144 preference. Shows the type of cellular network your phone prefers to  
 2145 connect to (e.g., 5G, 4G, 3G, 2G).  
 2146 4. **\*\*check\_sim\_status\*\*** - Checks if your SIM card is working correctly  
 2147 and displays its current status. Shows if the SIM is active, missing, or  
 2148 locked with a PIN or PUK code.  
 2149 5. **\*\*check\_data\_restriction\_status\*\*** - Checks if your phone has any  
 2150 data-limiting features active. Shows if Data Saver mode is on and  
 2151 whether background data usage is restricted globally.  
 2152 6. **\*\*check\_apn\_settings\*\*** - Checks the technical APN settings your phone  
 2153 uses to connect to your carrier's mobile data network. Shows current APN  
 2154 name and MMSC URL for picture messaging.  
 2155 7. **\*\*check\_wifi\_status\*\*** - Checks your Wi-Fi connection status. Shows if  
 2156 Wi-Fi is turned on, which network you're connected to (if any), and the  
 2157 signal strength.  
 2158 8. **\*\*check\_wifi\_calling\_status\*\*** - Checks if Wi-Fi Calling is enabled on  
 2159 your device. This feature allows you to make and receive calls over a  
 2160 Wi-Fi network instead of using the cellular network.  
 2161 9. **\*\*check\_vpn\_status\*\*** - Checks if you're using a VPN (Virtual Private  
 2162 Network) connection. Shows if a VPN is active, connected, and displays  
 2163 any available connection details.  
 2164 10. **\*\*check\_installed\_apps\*\*** - Returns the name of all installed apps on  
 2165 the phone.  
 2166 11. **\*\*check\_app\_status\*\*** - Checks detailed information about a specific  
 2167 app. Shows its permissions and background data usage settings.  
 2168 12. **\*\*check\_app\_permissions\*\*** - Checks what permissions a specific app  
 2169 currently has. Shows if the app has access to features like storage,  
 2170 camera, location, etc.

```

2160
2161 13. **run_speed_test** - Measures your current internet connection speed
2162 (download speed). Provides information about connection quality and what
2163 activities it can support. Download speed can be "unknown", "very poor",
2164 "poor", "fair", "good", or "excellent".
2165 14. **can_send_mms** - Checks if the messaging app can send MMS
2166 messages.

2167 ## Fix Actions (Write/Modify)
2168 1. **set_network_mode_preference** - Changes the type of cellular
2169 network your phone prefers to connect to (e.g., 5G, 4G, 3G).
2170 Higher-speed networks (5G, 4G) provide faster data but may use more
2171 battery.
2172 2. **toggle_airplane_mode** - Turns Airplane Mode ON or OFF. When ON, it
2173 disconnects all wireless communications including cellular, Wi-Fi, and
2174 Bluetooth.
2175 3. **reset_sim_card** - Simulates removing and reinserting your SIM
2176 card. This can help resolve recognition issues.
2177 4. **toggle_data** - Turns your phone's mobile data connection ON or
2178 OFF. Controls whether your phone can use cellular data for internet
2179 access when Wi-Fi is unavailable.
2180 5. **toggle_roaming** - Turns Data Roaming ON or OFF. When ON, roaming
2181 is enabled and your phone can use data networks in areas outside your
2182 carrier's coverage.
2183 6. **toggle_data_saver_mode** - Turns Data Saver mode ON or OFF. When
2184 ON, it reduces data usage, which may affect data speed.
2185 7. **set_apn_settings** - Sets the APN settings for the phone.
2186 8. **reset_apn_settings** - Resets your APN settings to the default
2187 settings.
2188 9. **toggle_wifi** - Turns your phone's Wi-Fi radio ON or OFF. Controls
2189 whether your phone can discover and connect to wireless networks for
2190 internet access.
2191 10. **toggle_wifi_calling** - Turns Wi-Fi Calling ON or OFF. This
2192 feature allows you to make and receive calls over Wi-Fi instead of the
2193 cellular network, which can help in areas with weak cellular signal.
2194 11. **connect_vpn** - Connects to your VPN (Virtual Private Network).
2195 12. **disconnect_vpn** - Disconnects any active VPN (Virtual Private
2196 Network) connection. Stops routing your internet traffic through a VPN
2197 server, which might affect connection speed or access to content.
2198 13. **grant_app_permission** - Gives a specific permission to an app
2199 (like access to storage, camera, or location). Required for some app
2200 functions to work properly.
2201 14. **reboot_device** - Restarts your phone completely. This can help
2202 resolve many temporary software glitches by refreshing all running
2203 services and connections.

2204 # Understanding and Troubleshooting Your Phone's Cellular Service
2205 This section details for agents how a user's phone connects to the
2206 cellular network (often referred to as "service") and provides
2207 procedures to troubleshoot common issues. Good cellular service is
2208 required for calls, texts, and mobile data.

2209 ## Common Service Issues and Their Causes
2210 If the user is experiencing service problems, here are some common
2211 causes:
2212
2213 * **Airplane Mode is ON**: This disables all wireless radios,
2214 including cellular.
* **SIM Card Problems**:
  * Not inserted or improperly seated.
  * Locked due to incorrect PIN/PUK entries.
* **Incorrect Network Settings**: APN settings might be incorrect
2215 resulting in a loss of service.
* **Carrier Issues**: Your line might be inactive due to billing
2216 problems.

```

```

2214
2215
2216 ## Diagnosing Service Issues
2217 `check_status_bar()` can be used to check if the user is facing a
2218 service issue.
2219 If there is cellular service, the status bar will return a signal
2220 strength indicator.

2221 ## Troubleshooting Service Problems
2222 ### Airplane Mode
2223 Airplane Mode is a feature that disables all wireless radios, including
2224 cellular. If it is enabled, it will prevent any cellular connection.
2225 You can check if Airplane Mode is ON by using `check_status_bar()` or
2226 `check_network_status()`.
2227 If it is ON, guide the user to use `toggle_airplane_mode()` to turn it
2228 OFF.

2229 ### SIM Card Issues
2230 The SIM card is the physical card that contains the user's information
2231 and allows the phone to connect to the cellular network.
2232 Problems with the SIM card can lead to a complete loss of service.
2233 The most common issue is that the SIM card is not properly seated or the
2234 user has entered the wrong PIN or PUK code.
2235 Use `check_sim_status()` to check the status of the SIM card.
2236 If it shows "Missing", guide the user to use `reset_sim_card()` to
2237 ensure the SIM card is correctly inserted.
2238 If it shows "Locked" (due to incorrect PIN or PUK entries), **escalate to technical support for assistance with SIM security**.
2239 If it shows "Active", the SIM itself is likely okay.

2240 ### Incorrect APN Settings
2241 Access Point Name (APN) settings are crucial for network connectivity.
2242 If `check_apn_settings()` shows "Incorrect", guide the user to use
2243 `reset_apn_settings()` to reset the APN settings.
2244 After resetting the APN settings, the user must be instructed to use
2245 `reboot_device()` for the changes to apply.

2246 ### Line Suspension
2247 If the line is suspended, the user will not have cellular service.
2248 Investigate if the line is suspended. Refer to the general agent policy
2249 for guidelines on handling line suspensions.
2250 * If the line is suspended and the agent can lift the suspension (per
2251 general policy), verify if service is restored.
2252 * If the suspension cannot be lifted by the agent (e.g., due to
2253 contract end date as mentioned in general policy, or other reasons not
2254 resolvable by the agent), **escalate to technical support**.
2255

2256 # Understanding and Troubleshooting Your Phone's Mobile Data
2257 This section explains for agents how a user's phone uses mobile data for
2258 internet access when Wi-Fi is unavailable, and details troubleshooting
2259 for common connectivity and speed issues.

2260 ## What is Mobile Data?
2261 Mobile data allows the phone to connect to the internet using the
2262 carrier's cellular network. This enables browsing websites, using apps,
2263 streaming video, and sending/receiving emails when not connected to
2264 Wi-Fi. The status bar usually shows icons like "5G", "LTE", "4G", "3G",
2265 "H+", or "E" to indicate an active mobile data connection and its type.

2266 ## Prerequisites for Mobile Data
2267 For mobile data to work, the user must first have **cellular service**.
2268 Refer to the "Understanding and Troubleshooting Your Phone's Cellular
2269 Service" guide if the user does not have service.

```

```

2268
2269
2270 ## Common Mobile Data Issues and Causes
2271 Even with cellular service, mobile data problems might occur. Common
2272 reasons include:
2273
2274 * **Airplane Mode is ON**: Disables all wireless connections,
2275 including mobile data.
2276 * **Mobile Data is Turned OFF**: The main switch for mobile data might
2277 be disabled in the phone's settings.
2278 * **Roaming Issues (When User is Abroad)**:
2279     * Data Roaming is turned OFF on the phone.
2280     * The line is not roaming enabled.
2281 * **Data Plan Limits Reached**: The user may have used up their
2282 monthly data allowance, and the carrier has slowed down or cut off data.
2283 * **Data Saver Mode is ON**: This feature restricts background data
2284 usage and can make some apps or services seem slow or unresponsive to
2285 save data.
2286 * **VPN Issues**: An active VPN connection might be slow or
2287 misconfigured, affecting data speeds or connectivity.
2288 * **Bad Network Preferences**: The phone is set to an older network
2289 technology like 2G/3G.
2290
2291 ## Diagnosing Mobile Data Issues
2292 `run_speed_test()` can be used to check for potential issues with mobile
2293 data.
2294 When mobile data is unavailable a speed test should return 'no
2295 connection'.
2296 If data is available, a speed test will also return the data speed.
2297 Any speed below 'Excellent' is considered slow.
2298
2299 ## Troubleshooting Mobile Data Problems
2300 ### Airplane Mode
2301 Refer to the "Understanding and Troubleshooting Your Phone's Cellular
2302 Service" section for instructions on how to check and turn off Airplane
2303 Mode.
2304
2305 ### Mobile Data Disabled
2306 Mobile data switch allows the phone to connect to the internet using the
2307 carrier's cellular network.
2308 If `check_network_status()` shows mobile data is disabled, guide the
2309 user to use `toggle_data()` to turn mobile data ON.
2310
2311 ### Addressing Data Roaming Problems
2312 Data roaming allows the user to use their phone's data connection in
2313 areas outside their home network (e.g. when traveling abroad).
2314 If the user is outside their carrier's primary coverage area (roaming)
2315 and mobile data isn't working, guide them to use `toggle_roaming()` to
2316 ensure Data Roaming is ON.
2317 You should check that the line associated with the phone number the user
2318 provided is roaming enabled. If it is not, the user will not be able to
2319 use their phone's data connection in areas outside their home network.
2320 Refer to the general policy for guidelines on enabling roaming.
2321
2322 ### Data Saver Mode
2323 Data Saver mode is a feature that restricts background data usage and
2324 can affect data speeds.
2325 If `check_data_restriction_status()` shows "Data Saver mode is ON",
2326 guide the user to use `toggle_data_saver_mode()` to turn it OFF.
2327
2328 ### VPN Connection Issues
2329 VPN (Virtual Private Network) is a feature that encrypts internet
2330 traffic and can help improve data speeds and security.
2331 However in some cases, a VPN can cause speed to drop significantly.
2332 If `check_vpn_status()` shows "VPN is ON and connected" and performance
2333 level is "Poor", guide the user to use `disconnect_vpn()` to disconnect
2334 the VPN.

```

2322  
 2323  
 2324 **### Data Plan Limits Reached**  
 2325 Each plan specify the maximum data usage per month.  
 2326 If the user's data usage for a line associated with the phone number the  
 2327 user provided exceeds the plan's data limit, data connectivity will be  
 2328 lost.  
 2329 The user has 2 options:  
 2330 - Change to a plan with more data.  
 2331 - Add more data to the line by "refueling" data at a price per GB  
 2332 specified by the plan.  
 2333 Refer to the general policy for guidelines on those options.

2334 **### Optimizing Network Mode Preferences**  
 2335 Network mode preferences are the settings that determine the type of  
 2336 cellular network the phone will connect to.  
 2337 Using older modes like 2G/3G can significantly limit speed.  
 2338 If `check_network_mode_preference()` shows "2G" or "3G", guide the user  
 2339 to use `set_network_mode_preference(mode: str)` with the mode  
 2340 `"4g_5g_preferred"` to allow the phone to connect to 5G.

2341 **# Understanding and Troubleshooting MMS (Picture/Video Messaging)**  
 2342 This section explains for agents how to troubleshoot Multimedia  
 2343 Messaging Service (MMS), which allows users to send and receive messages  
 2344 containing pictures, videos, or audio.

2345 **## What is MMS?**  
 2346 MMS is an extension of SMS (text messaging) that allows for multimedia  
 2347 content. When a user sends a photo to a friend via their messaging app,  
 2348 they're typically using MMS.

2349 **## Prerequisites for MMS**  
 2350 For MMS to work, the user must have cellular service and mobile data  
 2351 (any speed).  
 2352 Refer to the "Understanding and Troubleshooting Your Phone's Cellular  
 2353 Service" and "Understanding and Troubleshooting Your Phone's Mobile  
 2354 Data" sections for more information.

2355 **## Common MMS Issues and Causes**  
 2356 \* **\*\*No Cellular Service or Mobile Data Off/Not Working\*\*:** The most  
 2357 common reasons. MMS relies on these.  
 2358 \* **\*\*Incorrect APN Settings\*\*:** Specifically, a missing or incorrect  
 2359 MMSC URL.  
 2360 \* **\*\*Connected to 2G Network\*\*:** 2G networks are generally not suitable  
 2361 for MMS.  
 2362 \* **\*\*Wi-Fi Calling Configuration\*\*:** In some cases, how Wi-Fi Calling is  
 2363 configured can affect MMS, especially if your carrier doesn't support  
 2364 MMS over Wi-Fi.  
 2365 \* **\*\*App Permissions\*\*:** The messaging app needs permission to access  
 2366 storage (for the media files) and usually SMS functionalities.

2367 **## Diagnosing MMS Issues**  
 2368 `can_send_mms()` tool on the user's phone can be used to check if the  
 2369 user is facing an MMS issue.

2370 **## Troubleshooting MMS Problems**  
 2371 **### Ensuring Basic Connectivity for MMS**  
 2372 Successful MMS messaging relies on fundamental service and data  
 2373 connectivity. This section covers verifying these prerequisites.  
 2374 First, ensure the user can make calls and that their mobile data is  
 2375 working for other apps (e.g., browsing the web). Refer to the  
 2376 "Understanding and Troubleshooting Your Phone's Cellular Service" and  
 2377 "Understanding and Troubleshooting Your Phone's Mobile Data" sections if  
 2378 needed.

2376  
 2377 **### Unsuitable Network Technology for MMS**  
 2378 MMS has specific network requirements; older technologies like 2G are  
 2379 insufficient. This section explains how to check the network type and  
 2380 change it if necessary.  
 2381 MMS requires at least a 3G network connection; 2G networks are generally  
 2382 not suitable.  
 2383 If `check_network_status()` shows "2G", guide the user to use  
 2384 `set_network_mode_preference(mode: str)` to switch to a network mode  
 2385 that includes 3G, 4G, or 5G (e.g., `"4g_5g_preferred"` or `"4g_only"`).  
 2386  
 2387 **### Verifying APN (MMSC URL) for MMS**  
 2388 MMSC is the Multimedia Messaging Service Center. It is the server that  
 2389 handles MMS messages. Without a correct MMSC URL, the user will not be  
 2390 able to send or receive MMS messages.  
 2391 Those are specified as part of the APN settings. Incorrect MMSC URL, are  
 2392 a very common cause of MMS issues.  
 2393 If `check_apn_settings()` shows MMSC URL is not set, guide the user to  
 2394 use `reset_apn_settings()` to reset the APN settings.  
 2395 After resetting the APN settings, the user must be instructed to use  
 2396 `reboot_device()` for the changes to apply.  
 2397  
 2398 **### Investigating Wi-Fi Calling Interference with MMS**  
 2399 Wi-Fi Calling settings can sometimes conflict with MMS functionality.  
 2400 If `check_wifi_calling_status()` shows "Wi-Fi Calling is ON", guide the  
 2401 user to use `toggle_wifi_calling()` to turn it OFF.  
 2402  
 2403 **### Messaging App Lacks Necessary Permissions**  
 2404 The messaging app needs specific permissions to handle media and send  
 2405 messages.  
 2406 If `check_app_permissions(app_name="messaging")` shows "storage" and  
 2407 "sms" permissions are not listed as granted, guide the user to use  
 2408 `grant_app_permission(app_name="messaging", permission="storage")` and  
 2409 `grant_app_permission(app_name="messaging", permission="sms")` to grant  
 2410 the necessary permissions.

#### D.2.4 TROUBLESHOOTING WORKFLOW GRAPHS

To help the agent understand the troubleshooting workflow, we provide a decision graph for each issue type.

## E USER SIMULATOR QUALITY

### E.1 COMMON ERROR TYPES AND FAILURE MODES (RETAIL)

Manual analysis of the 20 annotated errors in the `retail` domain exposes three recurring failure modes:

- **Conversation-structure rule violation** (11/20) — the simulator breaks turn-taking or dialogue-flow instructions (e.g., mixes tool calls with natural language in the same turn).
- **Premature termination** (3/20) — the simulator halts the conversation immediately after the user's confirmation (###STOP###), preventing the agent from completing the transaction.
- **Ungrounded reference** (2/20) — the simulator invents or misstates contextual details such as payment method or order status.
- **Missing constraint** (4/20) — the simulator omits a required instruction (e.g., neglects to request an alternative SKU when the desired colour is unavailable).

Most task-critical errors stem from either premature termination or missing constraints, whereas conversation-structure violations and ungrounded references are typically task-benign and readily recoverable by the agent.

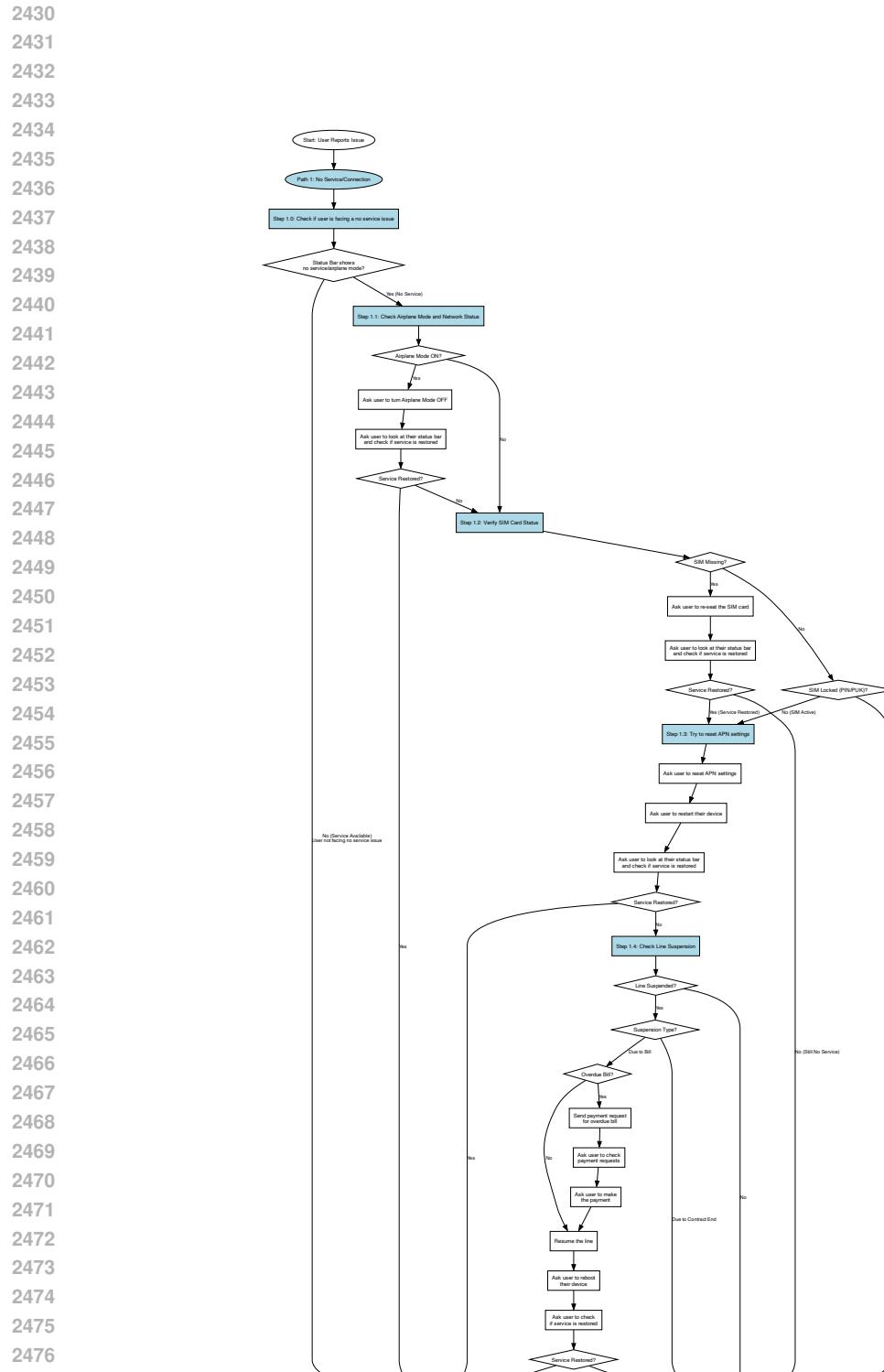


Figure 9: Troubleshooting workflow for service\_issue

2484

2485

2486

2487

2488

2489

2490

2491

2492

2493

2494

2495

2496

2497

2498

2499

2500

2501

2502

2503

2504

2505

2506

2507

2508

2509

2510

2511

2512

2513

2514

2515

2516

2517

2518

2519

2520

2521

2522

2523

2524

2525

2526

2527

2528

2529

2530

2531

2532

2533

2534

2535

2536

2537

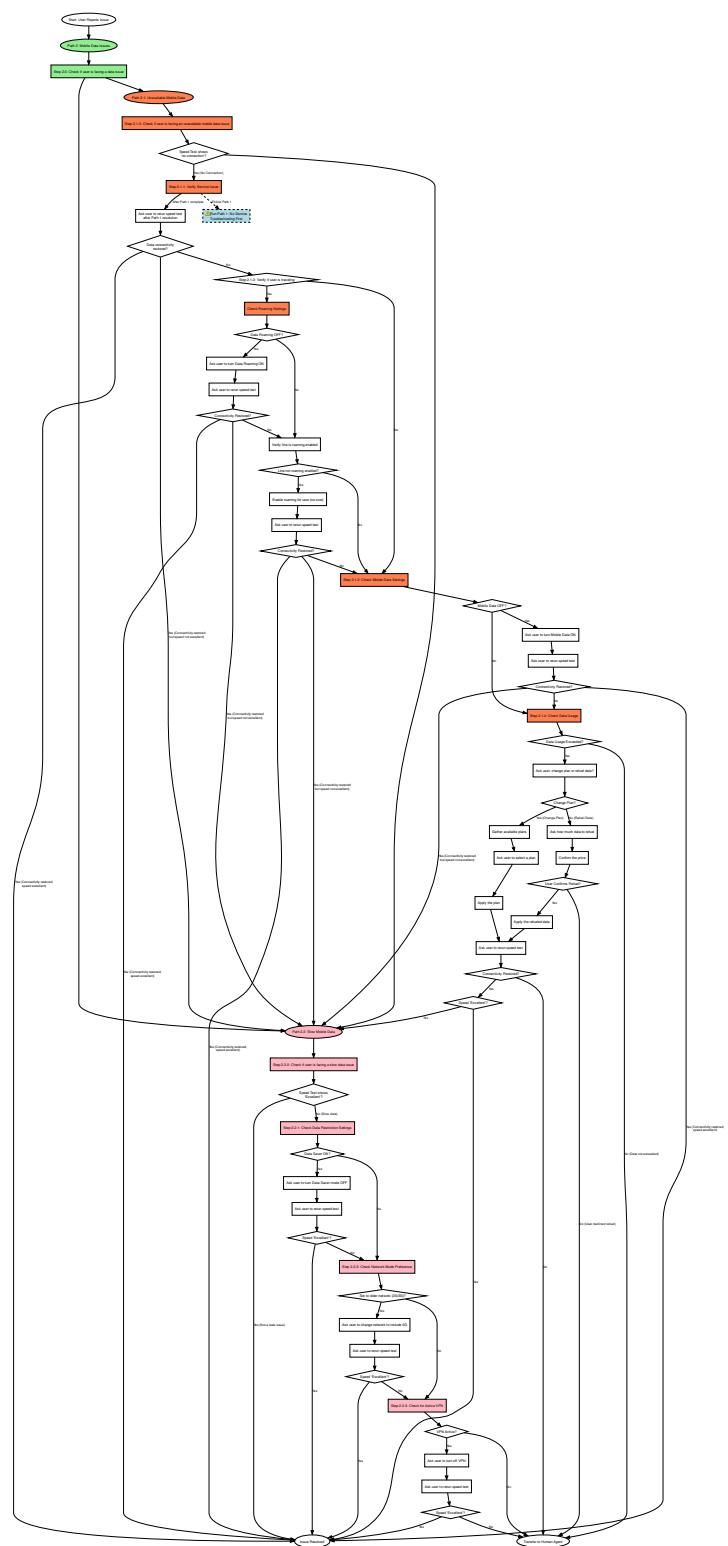


Figure 10: Troubleshooting workflow for mobile\_data\_issue

2538

2539

2540

2541

2542

2543

2544

2545

2546

2547

2548

2549

2550

2551

2552

2553

2554

2555

2556

2557

2558

2559

2560

2561

2562

2563

2564

2565

2566

2567

2568

2569

2570

2571

2572

2573

2574

2575

2576

2577

2578

2579

2580

2581

2582

2583

2584

2585

2586

2587

2588

2589

2590

2591

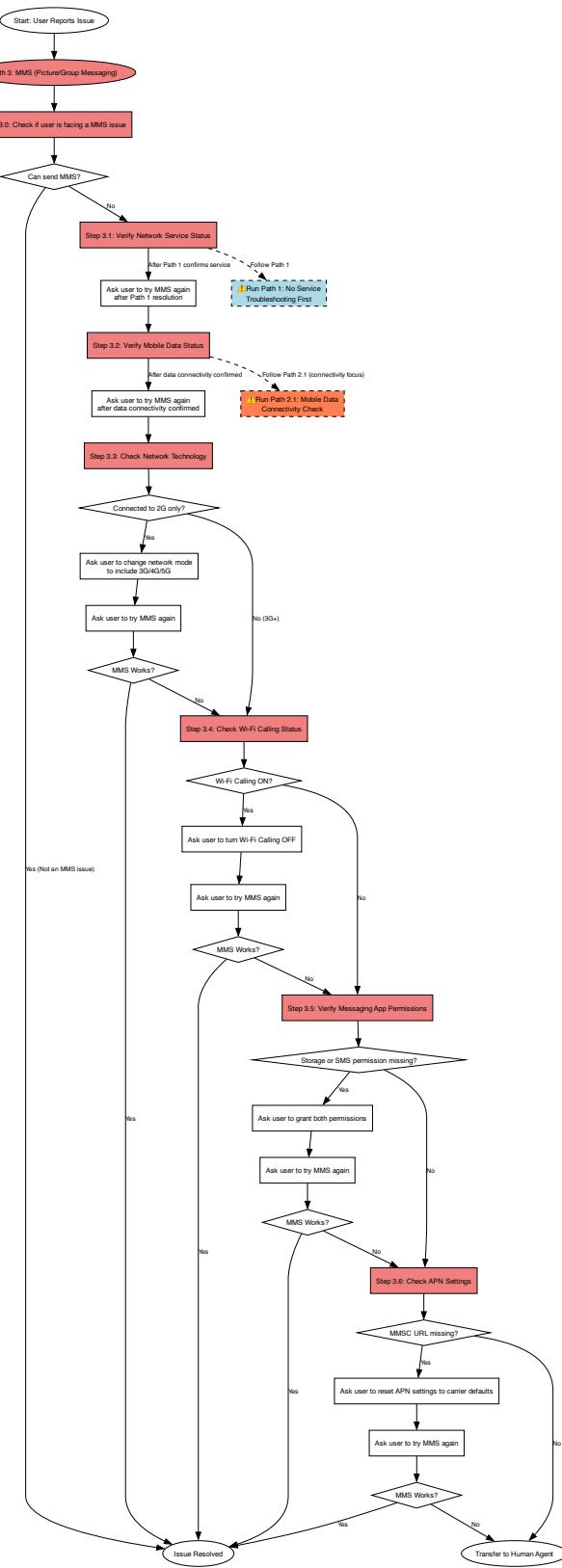


Figure 11: Troubleshooting workflow for mms\_issue

2592 E.2 COMMON ERROR TYPES AND FAILURE MODES (AIRLINE)  
25932594 Preliminary analysis of annotated `airline` dialogues identifies three dominant issues:  
25952596 • **Missing constraint** (11/47)  
2597 • **Conversation-structure rule violation** (19/47)  
2598 • **Ungrounded reference** (15/47)  
2599 • **Premature termination** (2/47)  
26002601 Again, most task-critical errors stem from missing constraints.  
26022603 E.3 COMMON ERROR TYPES AND FAILURE MODES (TELECOM)  
26042605 Manual analysis of the 8 annotated errors in the `telecom` domain reveals that all errors are caused  
2606 by **Premature termination** (8/8, 100%). The user returns the `###TRANSFER###` token before the  
2607 assistant calls the transfer tool. This error is critical only if this causes a task failure by preventing the  
2608 agent from performing an expected transfer.  
26092610 E.4 ILLUSTRATIVE CASES  
26112612 **Task-critical error (`retail`).** **Premature termination.** After the assistant requests final confirmation  
2613 to modify order #W5061109, the simulator ends the dialogue with `###STOP###` immediately  
2614 following confirming its intent, preventing the agent from executing the modification and issuing a  
2615 receipt:  
26162617     **assistant:** Please confirm that you have provided all the items you want to modify  
2618     in this order, as this action can only be done once...  
2619     **user:** Yes, I confirm that I want to modify my pending order #W5061109 to  
2620     ... Please proceed. Thank you! `###STOP###`2621 **Task-critical error (`airline`).** **Missing constraint.** Here the simulator is instructed to ask for a  
2622 transfer to a human agent in the case of outbound trip unavailability, but proceeds to book the return  
2623 trip instead, violating goal constraints.  
26242625     **assistant:** It appears that one of your original outbound flights (HAT030 on  
2626     2024-05-13) is no longer available in economy class for the upgrade. To proceed, I  
2627     will need to find available economy class flights for your outbound journey as well.  
2628     Would you like me to search for available economy class flights for your outbound  
2629     trip (Los Angeles → Atlanta) and your new return trip (Atlanta → Las Vegas)?  
2630     Please confirm so I can continue.  
2631     **user:** Yes, please go ahead and find available economy class flights for both  
2632     the outbound trip to Atlanta and the new return trip to Las Vegas. Thank you!  
2633     `###STOP###`2634 **Task-critical error (`telecom`).** **Premature termination.** The simulator ends the dialogue with  
2635 `###TRANSFER###` before the agent could perform the actual transfer:  
26362637     **assistant:** ... The next step is to escalate your issue to a human technical support  
2638     agent who can provide more advanced help, possibly including SIM replacement  
2639     or device diagnostics. Would you like me to transfer you to a human agent for  
2640     further assistance?  
2641     **user:** Yes, please transfer me to someone who can help further. I just want my  
2642     phone to work again. Thank you for your help so far. `###TRANSFER###`2643 F USE OF LARGE LANGUAGE MODELS  
26442645 As part of the paper writing process, we utilized Large Language Models (LLMs). The use of these  
models was restricted to improving the clarity and readability of the text. Specifically, we used LLMs

2646 for grammar correction, rephrasing sentences for better flow, and ensuring consistent terminology.  
2647 The core ideas, experimental results, and scientific contributions presented in this paper are entirely  
2648 our own.

2649

2650

2651

2652

2653

2654

2655

2656

2657

2658

2659

2660

2661

2662

2663

2664

2665

2666

2667

2668

2669

2670

2671

2672

2673

2674

2675

2676

2677

2678

2679

2680

2681

2682

2683

2684

2685

2686

2687

2688

2689

2690

2691

2692

2693

2694

2695

2696

2697

2698

2699