TOOLTailor: PERSONALISED TOOL USE IN MULTI-TURN TRAVEL PLANNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Tool-augmented language models have become the dominant paradigm for task-oriented assistants, yet existing benchmarks rarely evaluate whether these models integrate individual user preferences into their tool calls. We introduce ToolTailor, a benchmark for personalised tool use in travel planning, together with a synthetic data generation pipeline that produces personalised tool-use trajectories at scale. ToolTailor is the first benchmark to combine long-horizon, multi-turn tool use with rich user profiles, requiring agents to satisfy individual preferences while orchestrating multiple tools across a task. Tasks are seeded from travel-planning scenarios collected from web forums, and the development and test splits are human-validated. Evaluation with state-of-the-art LLMs shows that personalised tool use remains challenging: models fail to identify and incorporate relevant user preferences into tool calling. Supervised fine-tuning on our synthetic trajectories improves preference integration by up to 35 F1 points when the relevant preferences are provided directly, while also improving schema compliance. When the model must instead identify them in the full user profile, all four fine-tuned open models still outperform three proprietary models (Gemini-3.6-Flash, GPT-4o-mini and GPT-5.6-Terra) prompted zero-shot.[1].

1 INTRODUCTION

Advances in large language models (Zhao et al., 2025) have shifted attention from general-purpose assistants towards systems that adapt to individual users (Zhang et al., 2025). Most existing work on LLM personalisation adapts the *generated response*: modifying style, tone, or complexity (Salemi et al., 2024; Mysore et al., 2024; He et al., 2025), or improving recommendations from user interaction history (Chen, 2023; Li et al., 2024; Wu et al., 2024). While these approaches make responses *appear* personalised, they rarely affect the underlying decision-making process. In task-oriented settings, however, preferences must shape not only what a model says but also how it acts.

Consider a travel-planning assistant. Completing a request may require coordinating accommodation, transport, restaurant, and calendar tools while respecting constraints such as budget, accessibility, dietary requirements, and preferred providers. Missing one relevant preference can alter the selected tool, its arguments, or the resulting plan. Personalised tool use, therefore, introduces challenges beyond personalised response generation: the model must identify which preferences are relevant and incorporate them throughout a sequence of interdependent decisions.

Modern benchmarks for tool-augmented LLMs increasingly evaluate more realistic agent capabilities including long-horizon planning, multi-turn interaction, orchestration of multiple tools, and execution in complex environments (Lu et al., 2024; Andrews et al., 2025; Ahn et al., 2026; He et al., 2026; Wang et al., 2026; Zeng et al., 2026). A separate line of work studies how user preferences affect tool use (Moghe et al., 2024; Wang et al., 2024; Hao et al., 2025; Cai et al., 2025; Wang et al., 2025). Yet, existing agent benchmarks generally overlook personalisation, while personalised tool use benchmarks often consider relatively simple tasks and represent preferences as explicit instructions or shallow profiles. As a result, no existing benchmark evaluates long-horizon agentic tool use and realistic user constraints together (see Table 1).

[1]Code and data are released at anonymous.4open.science/r/tool-tailor

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Table 1: Comparison of tool use benchmarks across five key features. **Tool Tailor** is the only benchmark to jointly support all five. $*$ indicates partial or limited support.

Benchmark	Personalised	Multi-turn	Long-horizon	Multi-tool	Rich User Profiles
API-Bank (Li et al., 2023a)	✗	✓	✓*	✓	✗
ToolSandbox (Lu et al., 2024)	✗	✓	✓	✓	✗
OrchestrationBench (Ahn et al., 2026)	✗	✓	✓	✓	✗
TRAJECT-Bench (He et al., 2026)	✗	✓	✓	✓	✗
NLSI (Moghe et al., 2024)	✓	✗	✗	✓	✗
ETAPP (Hao et al., 2025)	✓	✓	✓*	✓	✓*
FamilyTool (Wang et al., 2025)	✓	✗	✓*	✓*	✗
Tool Tailor	✓	✓	✓	✓	✓

To address this gap, we introduce **ToolTailor**, a benchmark for personalised tool use in travel planning which combines multi-tool orchestration, long-horizon planning, and reasoning over rich user constraints. Unlike existing personalised tool-use datasets, our benchmark models users through detailed profiles combining general lifestyle preferences and recurring tool-specific choices. The benchmark includes realistic multi-tool environments, hierarchical task decomposition, and complete interaction trajectories for evaluating high-level planning and fine-grained tool use. Starting from travel-planning scenarios collected from web forums and expanded across diverse task themes, our staged generation pipeline constructs user profiles, task constraints, subgoals, and complete tool-use trajectories. Training data are automatically filtered and repaired, while the development and test data undergo human validation. This pipeline also provides scalable synthetic trajectories for supervised training. Our contributions are as follows:

- We introduce ToolTailor, a benchmark that combines realistic travel-planning tasks, detailed user profiles, and multi-tool orchestration to evaluate agents' abilities to incorporate personal preferences into tool calling;
- We propose a scalable framework for generating supervised training trajectories;
- We develop a comprehensive evaluation methodology that separately measures general tool use accuracy and personalised argument prediction, while distinguishing incorrect tool calls from alternative tool calls that still satisfy the task requirements;
- We conduct experiments with state-of-the-art LLMs showing that supervised fine-tuning on synthetic trajectories improves personalised tool use in every evaluation setting, with F1 gains ranging from $+3$ when the model must find the relevant preferences in the full user profile to $+35$ when they are provided directly. Our analysis reveals that identifying which user preferences are relevant remains a key challenge.

2 Related Work

LLM Personalisation Research on LLM personalisation has mainly considered how outputs should adapt to individual users. This includes personalised text generation (Salemi et al., 2024; Lee et al., 2024; Mysore et al., 2024; He et al., 2025); recommendation based on interaction histories (Li et al., 2023b; Yang et al., 2023; Lyu et al., 2024; Lee & Kim, 2025), and persona-conditioned user simulation (Tseng et al., 2024). These approaches personalise response content or style. We instead study *behavioural personalisation* in task-oriented settings, where user information must guide tool selection, argument filling, and the sequence of actions used to complete a task.

Tool-Augmented LLM Evaluation Tool use extends language models beyond text generation by allowing them to interact with external APIs and environments (Parisi et al., 2022; Schick et al., 2023). While early work focuses on how to use tools for question answering and mathematical reasoning (Patil et al., 2023; Hao et al., 2023; Zhuang et al., 2024; Yin et al., 2024), recent benchmarks evaluate more complex agent behaviours, including long-horizon planning, multi-tool orchestration, and tool execution in realistic environments (Mialon et al., 2023; Lu et al., 2024; Ahn et al., 2026; He et al., 2026; Wang et al., 2026; Yu et al., 2026; Zeng et al., 2026). They primarily evaluate general tool use capabilities without considering how user preferences influence agent decisions.

Personalised Tool Use A smaller body of work incorporates persistent user information into tool-use evaluation. NLSI (Moghe et al., 2024) tests whether models apply explicit natural language

instructions to tool calls. ETAPP (Hao et al., 2025) introduces detailed user profiles and proactive tool use in simulated environments, while FamilyTool (Wang et al., 2025) studies personalisation through reasoning over family knowledge graphs. These datasets typically focus on simpler tasks or represent preferences through explicit constraints, categorical attributes, or structured past interactions. In contrast, ToolTailor evaluates tool use in realistic travel-planning scenarios requiring long-horizon reasoning and multi-tool orchestration, while modelling users through rich profiles.

Synthetic Tool-Use Data Synthetic generation has increasingly been used to create tool-use instructions and interaction trajectories at scale. ToolLLM (Qin et al., 2024) automatically generates single- and multi-tool instructions and searches for valid chains of API calls, while AgentTuning (Zeng et al., 2024) collects and filters interaction trajectories across multiple environments. More recent pipelines place greater emphasis on data verification: APIGen (Liu et al., 2024) validates generated function calls through format checking, execution, and semantic verification, while ToolACE (Liu et al., 2025) combines multi-agent dialogue generation with rule- and model-based checks. APIGen-MT (Prabhakar et al., 2025) extends this direction to complete multi-turn trajectories generated through simulated user-agent interaction.

Rather than generating tool-use instructions or trajectories in isolation, our pipeline jointly constructs a user profile, general and tool-specific preferences, contextual constraints, hierarchical subgoals, and a complete interaction trajectory. This makes it possible to trace which user information should influence each tool call and to evaluate preference integration separately from general tool-use correctness. We further apply automatic consistency checks to the synthetic training data and human and expert validation to the evaluation data.

3 THE TOOLTailor BENCHMARK

3.1 DATASET FEATURES

We next describe the features underlying the design of our benchmark: realistic tasks such as travel-planning, rich user profiles, structured tool environments, and hierarchical task decomposition.

Travel Planning Travel planning provides a natural setting for personalised tool use because it requires agents to satisfy multiple constraints while coordinating tools across interdependent stages. In the example in Figure 1, the traveller has a 17-hour stopover at IAH and wants to leave the airport, but activities and restaurants must remain within a short travel radius and fit the available time. Decisions about location, transport, and times constrain the itinerary constructed at later stages.

Rich User Profiles Each scenario is paired with a detailed user profile containing demographic information, general lifestyle preferences, and tool-specific preferences. The profile in Figure 1 records an interest in green spaces and plant-based food, together with a preference for highly rated restaurants. Because preferences are natural-language descriptions rather than tool arguments, the model must infer that the last of these should set the restaurant tool's minimum-rating argument.

Structured Tool Environment The benchmark includes tools spanning six travel-related domains (Table 3, Appendix B). Each tool is modelled as an application with multiple functions, and several applications offer overlapping functionality. In Figure 1, for example, the agent uses a geographical-search tool to find nearby parks and a restaurant-search tool to retrieve dining options. Personalisation can affect both tool selection and argument filling, such as the chosen restaurant application, cuisine, search radius, or minimum rating.

Hierarchical Task Structure Each scenario is decomposed into an ordered sequence of subgoals corresponding to stages of the planning process. In Figure 1, the agent first researches activities and restaurants and then constructs a draft itinerary from the retrieved options. This decomposition guides trajectory generation and enables performance to be analysed at each planning stage.

Fine-grained Evaluation Beyond aggregate tool-call accuracy, ToolTailor distinguishes between failures at different levels. For example, a model may select the correct restaurant tool but omit the user's minimum-rating preference, choose the wrong function within the correct tool, or fail to request a restaurant at all. Reporting performance by personalised arguments, subgoal, and function type helps separate failures of preference integration from failures of general tool use.

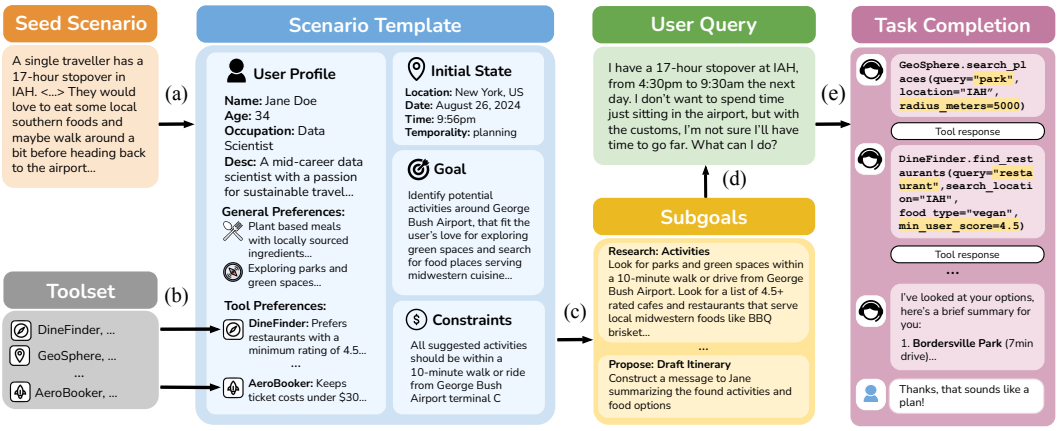


Figure 1: **Dataset generation pipeline.** Given a seed scenario, we **(a)** generate a scenario template defining the conversation context, characters, constraints, and goals; **(b)** enrich user profiles with general and tool-specific preferences; **(c)** decompose the task into subgoals; **(d)** simulate the initial user query; and **(e)** iteratively create task completion trajectories containing dialogue turns and tool calls for each subgoal.

3.2 Dataset Construction

We assemble ToolTailor in three stages. We first collect seed scenarios (Section 3.2.1), then expand them into a complete dataset through automated generation (Section 3.2.2), and finally, filter and validate the output (Sections 3.2.3–3.2.4).

3.2.1 Seed Scenario Collection

To ensure the scenarios in ToolTailor are realistic, we manually collect ten user queries from web forums (e.g., Reddit[2]). We refer to these examples as **seed scenarios**. To increase task diversity, we define 15 travel-planning themes (e.g., business travel, last minute plan cancellation, and multi-city itineraries) and use the ten manual examples as few-shot demonstrations, prompting the model to generate ten further seed scenarios per theme, repeating this process until the desired total is reached. The full list of themes is provided in Appendix C.2.2.

3.2.2 Dataset Generation

Starting from a seed scenario, we generate a personalised tool-use trajectory through the five-stage pipeline in Figure 1. In the running example, a short description of an airport stopover is expanded into: (a) a structured scenario template containing the user, setting, goal, and constraints; (b) a profile enriched with general and tool-specific preferences; (c) a scenario-specific sequence of subgoals; (d) an initial user query that expresses the task without explicitly revealing the stored preferences; and (e) a multi-turn trajectory containing dialogue, tool calls, and tool outputs. Each stage consumes the structured output of the preceding stage. Training data is generated with Qwen3-8B, while the human-validated development and test splits use the larger Qwen3-14B to reduce generation artifacts; full details are in Section C.2.

Scenario Templates (Figure 1a) Given a seed scenario, we generate the structured scenario template illustrated in Figure 1(a). It specifies the initial conversation state, the user profile, the task goal, and the constraints that must hold during completion. In the running example, the informal stopover request is converted into explicit temporal and geographical constraints, including the airport location, the duration of the stopover, and the maximum travel distance from the terminal. We distinguish between two types of constraints: *resource constraints*, such as financial or time limitations, and *contextual constraints*, including geospatial and accessibility requirements (see Table 6).

Tool Preferences (Figure 1b) To introduce tool-specific personalisation, we assign each user a personal toolset by randomly sampling one tool from each category. For each selected tool, we

[2]reddit.com

generate two or three preferences consistent with the broader user profile, such as a preferred airline, dietary requirements, a minimum restaurant rating, or a preference for using a particular application.

Subgoals (Figure 1c) To guide task completion, we manually curate a set of high-level subgoals representing stages of travel planning (detailed in Table 7). For each scenario, the model then selects the relevant subgoals from this set and adapts them to the user goal, producing a scenario-specific task-completion plan.

User Query (Figure 1d) Using the generated scenario and profile, the model produces the initial user message shown in Figure 1(d). The message communicates the immediate request and conversational context but deliberately omits stored personal preferences. Models must retrieve and apply relevant preferences from the profile rather than copying explicit constraints from the query.

Task Completion Trajectories (Figure 1e) Finally, we generate complete trajectories containing all dialogue turns and tool interactions required to complete the task. Rather than generating an entire trajectory at once, we iteratively generate one message at a time while maintaining the conversation state and active subgoal. We decompose the process into four components:

- **Action LLM**: generates assistant actions, including tool calls and responses;
- **User LLM**: simulates user responses, generating replies that reflect the user's goals, preferences, personality traits, and behavioural patterns inferred from the user profile;
- **Tool Simulator**: produces tool outputs according to tool schemas and example responses;
- **Task Completion Checker**: determines whether the active subgoal has been completed and provides guidance for subsequent actions.

In the example in Figure 1(e), the Action LLM invokes geographical and restaurant-search tools, the Tool Simulator returns candidate locations, and the Action LLM uses these outputs to construct the itinerary. The Task Completion Checker judges when the research subgoal has been satisfied against explicit stopping criteria (e.g., user approval confirmed if needed) before generation proceeds to itinerary construction. The components communicate through constrained message types representing clarification requests, tool calls, tool outputs, and dialogue responses. Generation continues until all subgoals are completed (see Appendix C.2.5); Figure 6 illustrates a full trajectory.

3.2.3 Filtering and Post-Processing

To ensure high-quality data, we combine automatic filtering with rule-based post-processing. Automatic checks verify structural validity: adherence to tool schemas, valid tool selection given the toolset, and consistent personalised preferences across tool calls. A subsequent repair stage then fixes common issues flagged by these checks, correcting value types to match tool schemas and propagating personalised preferences consistently across all relevant tool calls and assistant actions.

3.2.4 Human Validation

To ensure data quality, we rely on human validation. Annotators are recruited through Prolific[3] after a training phase with at least 80% accuracy. We include control tasks covering 20% of each annotator's workload and retain only annotations from participants who correctly complete all controls. Annotators are compensated at an average rate of £13.19 per hour, above the UK minimum wage.

Validation consists of three independent annotation tasks. First, annotators revise the **user query** to ensure that it clearly specifies the request without revealing personalised preferences. Second, they verify the logical consistency of the **scenario setting**, including location, date, time, and whether the task requires immediate assistance or future planning. Third, they identify the **subgoals** required to complete the task. Each scenario is annotated once for each task, after which an expert resolves any inconsistencies. Validation changes a substantial share of the generated data: annotators revise 62.6% of user queries, 63.7% of scenario settings and 86.3% of subgoal sets, and the expert subsequently corrects 24.4%, 24.4% and 62.7% of those outputs respectively (Table 11). These rates indicate both that human validation is necessary and that the synthetic training split, which receives only automatic filtering, is noisier than the evaluation splits. Crowd and expert correction rates for each project are reported in Appendix D, which also details recruitment and annotation procedures.

[3] prolific.com

Table 2: Dataset statistics for **ToolTailor**. *Test Balanced* is the curated test set obtained via balanced sampling over individual tool-calling steps; *Test Full* (**ToolTailor**-Full) contains the complete, un-subsampled test trajectories for the same scenarios.

Statistic	Train	Dev	Test Balanced	Test Full
Scenarios	82	10	331	331
Instances	2313	67	1945	5771
Tool Calls / Scenario	28.21	6.70	5.88	17.44
Subgoals / Scenario	5.27	4.10	6.44	6.44

For test-set task-completion trajectories, we use a human-in-the-loop generation process. An LLM generates the trajectory incrementally, one subgoal at a time. After each segment, an expert reviews and corrects it with assistance from ClaudeCode[4] (Anthropic, 2026), which identifies potential inconsistencies and suggests edits. The assistant only proposes edits: it neither accepts them nor sees the reference labels, and every change is approved by the expert before use. The expert verifies all edits resolves any remaining errors manually before the corrected segment is used as context for generating the next one. This process continues until the full trajectory is finalised (see Appendix C.2.6 for more details).

3.3 Final Dataset

Following API-Bank (Li et al., 2023a), we decompose each scenario into a sequence of **evaluation instances**, one for each tool call. Each instance contains the full preceding dialogue, the user profile, and the available tools, with the next tool call as the prediction target. Evaluation in ToolTailor therefore measures step-wise decision quality within a long-horizon task, rather than end-to-end task completion, and assesses whether a model can select the correct next tool call given the interaction history and the user's preferences.

Table 2 summarises the size of ToolTailor's splits. The final benchmark comprises a synthetic training split and human-validated development and test splits. We construct the test split, *Test Balanced*, by sampling evaluation instances from the complete test trajectories, balancing coverage across active subgoals, tools, function types, and numbers of relevant user preferences. We additionally release *Test Full* (ToolTailor-Full), which contains the complete trajectories for all 331 test scenarios. Whereas the primary benchmark supports fine-grained evaluation of individual tool-use decisions, ToolTailor-Full enables analysis of long-horizon planning and end-to-end tool use. The corresponding per-tool distribution of test instances is provided in Appendix B.

4 Evaluation

Since ToolTailor does not include executable tools, we evaluate predicted tool calls against reference calls. Extending the methodology of Moghe et al. (2024), we represent each tool call as a 4-tuple $\langle$`tool`, `function`, `argument name`, `argument value`$\rangle$, and compare predicted and reference tuples using exact match. For open-text values, we instead use embedding-based semantic matching with cosine similarity, which allows us to treat semantically equivalent values as correct. The values are encoded with `all-MiniLM-L6-v2` (Reimers & Gurevych, 2019) and counted as matching at a cosine similarity of at least 0.8. All other values, including numbers, dates and categorical options, use exact match.

Complex tasks like travel planning often admit multiple valid orders of execution, so a single reference trajectory may not capture every correct solution. ToolTailor encodes both dependencies between subgoals and tool calls within each subgoal. We report both strict trajectory matching against the reference order and dependency-aware matching, which accepts alternative tool-call orders provided that all annotated dependencies are satisfied. Unless stated otherwise, reported numbers use strict matching, which is the more conservative of the two. Additional details are in Appendix E.

[4] `claude-sonnet-5`

4.1 Evaluation Metrics

We propose to evaluate personalised tool use along three complementary dimensions:

Tool Use Our tool-use metrics capture whether the model calls the right tools with the right arguments. We evaluate overall tool-calling performance using *Precision*, *Recall*, *F1*, and *Exact Match* (EM), computed over tool-call 4-tuples. We additionally report *Tool Match*, which measures whether the correct tool is selected, and *Function Match*, which measures whether the correct (`tool`,`function`) pair is predicted. Together, these metrics evaluate whether the model correctly identifies both user intent and user tool-selection preferences.

Personalisation Metrics Unlike other tool-use benchmarks, ToolTailor focuses on personalised, preference-aware tool use: each tool schema in the dataset identifies a subset of arguments whose values should be determined by user preferences (e.g. dietary requirements or accommodation preferences). We compute *Precision*, *Recall*, *F1*, and *EM* exclusively over these personalised arguments, providing a direct measure of how well the model reflects user preferences in its tool calls.

Schema Compliance These metrics assess whether generated tool calls are structurally well-formed, independent of whether they are correct. We measure the proportion of arguments (i) whose names are valid, (ii) whose values have the correct type, and (iii) whose categorical values belong to the predefined option set, and additionally report the proportion of calls missing a required argument.

To support fine-grained analysis, all metrics are reported under three aggregation axes. **Dataset-level** aggregation measures overall performance across the benchmark. **Subgoal-level** aggregation groups tool calls by planning stage, exposing which stages are hardest. **Function-type** aggregation separates tool calls by functional role (e.g., information retrieval vs. action execution), so retrieval and environment-modifying behaviours can be analysed separately.

4.2 Experimental Setting

We evaluate four open-weight, instruction-tuned models: Qwen3-4B, Qwen3-8B, and Qwen3-14B (Yang et al., 2025) and Gemma3-4B (Team et al., 2025). As proprietary reference points, we also evaluate Gemini-3.6-Flash (Google, 2026), GPT-4o-mini (OpenAI, 2024) and GPT-5.6-Terra (OpenAI, 2026) zero-shot. Full per-metric results for all models are reported in Appendix G.

As a baseline, we evaluate each model zero-shot, then fine-tune it with LoRA adapters (Hu et al., 2022) on the synthetic training split, selecting checkpoints on the development set and evaluating on the test set. Unless otherwise specified, all experiments use a LoRA rank of 16 and scaling parameter $\alpha = 32$; we ablate alternative ranks on Qwen3-4B in Appendix F.2. Training runs for 4 epochs with a learning rate of 2×10^{-4}, an effective batch size of 32, and a maximum sequence length of 17,408. Additional details are provided in Appendix F. For both zero-shot and LoRA models, we repeat inference across three random seeds and report mean and standard deviation; variation across seeds is small, with a mean standard deviation of 0.43 F1 and a maximum of 3.2.

We evaluate models under settings designed to isolate their ability to identify relevant user preferences, select appropriate tools from a large pool, and generate correct tool calls: (1) **All Preferences & Tools**: the model receives the complete user profile together with the schemas of all available tools; (2) **Relevant Preferences**: only preferences relevant to the current request are provided, while all tool schemas remain available; and (3) **Relevant Preferences & Tools**: preferences and tools are restricted to those relevant for the task. Relevant subsets are selected deterministically from dataset metadata and the user profile: relevant preferences are the tool-specific preferences for the reference target tool plus the matching general preferences, and relevant tools are those in the user's profile whose categories are permitted by the active subgoal. Further details are in Appendix E.

5 Results

Figures 2–4 present the main evaluation results on ToolTailor. Additional results are provided in Appendix G. We summarise key findings below.

Fine-tuning substantially improves personalised tool use. Across all model families and all three evaluation configurations, supervised fine-tuning on our synthetic trajectories consistently improves

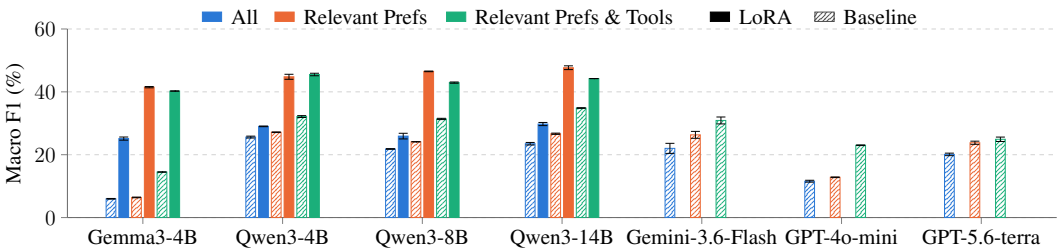


Figure 2: Overall F1 across the three evaluation configurations. Fine-tuning on synthetic trajectories improves personalised tool use over the zero-shot baseline for every model, with the largest gains when only relevant preferences are provided.

both overall tool use (Figure 2) and personalised argument prediction (Tables 17–18) over the zero-shot baseline. Under the most challenging *All Preferences & Tools* setting, F1 gains range from +3.41 (Qwen3-4B), +4.09 (Qwen3-8B), and +6.26 (Qwen3-14B) to +19.15 (Gemma3-4B). When only relevant preferences are provided, gains grow to 17–35 F1 in the *Relevant Preferences* setting and 13–30 F1 in *Relevant Preferences & Tools*. Exact match follows the same trend, rising by up to 9.42 points in *Relevant Preferences* and over 6 points in *Relevant Preferences & Tools*, again largest for Gemma3-4B. Tool and function selection improve markedly: tool-match increases by over 69 points for Gemma3-4B and exceeds 80% for both Gemma3-4B and all Qwen3 models under *Relevant Preferences*. Together, these results indicate that the synthetic trajectories teach models to select appropriate tools and functions while integrating user preferences into their calls.

Notably, all four LoRA-tuned models surpass Gemini-3.6-Flash, GPT-4o-mini, and GPT-5.6-Terra used zero-shot under the *All Preferences & Tools* setting (22.02/11.57/20.10 F1, respectively). The effect is clearest for Gemma3-4B, which rises from 5.98 F1 — below all three baselines — to 25.13. Full results are in Appendix G, Table 17.

Fine-tuning generally improves schema compliance. We also find that fine-tuning improves overall schema compliance for every model except Qwen3-14B, where it is essentially unchanged (82.42% vs. 82.21%), primarily through near-perfect argument-name precision and almost eliminating missing required arguments. Argument-type precision improves sharply for Gemma3-4B but is roughly flat for the Qwen models. We interpret this as a reflection of the supervision signal. Supervised training strongly reinforces *which* tool, function, and personalised arguments to select, but argument *type* correctness receives a weaker signal and can be overwritten by patterns in the synthetic data. Moreover, because our automatic repair step is heuristic-based, it may leave some type inconsistencies across ToolTailor's diverse tool schemas, allowing these errors to persist during training. Improving schema conformity is therefore an important direction we leave to future work.

Strict trajectory matching underestimates model capability. Before turning to specific capabilities, we note that our strict reference matching is conservative. Dependency-aware matching, which credits any tool call permitted by the task's dependency structure, consistently scores higher than strict matching (Table 17), with the largest gap on LoRA models under *All Preferences & Tools* (e.g., Qwen3-4B: 29.02 $\rightarrow$ 37.23 F1; Qwen3-8B: 25.92 $\rightarrow$ 34.68).

Models frequently produce valid tool calls that differ from the reference, and evaluating against a single fixed trajectory understates performance when multiple completion orders are valid. Dependency-aware matching recovers these alternative paths, but it scores each call independently and cannot verify that a model produces the *complete* set of calls a task requires; end-to-end trajectory evaluation remains necessary to assess genuine task completion.

Identifying relevant preferences is the primary bottleneck. Restricting the *preferences* shown to the model yields far larger gains than restricting the *tools*. Providing only task-relevant preferences produces the largest improvement for every model; Qwen3-4B, for instance, rises from 29.02 to 44.79 F1 (Figure 2), with similar jumps for Qwen3-8B and Gemma3-4B; restricting the toolset to relevant tools yields only modest gains. Identifying which user preferences apply is therefore harder than selecting the appropriate tool.

Figure 3 refines this picture by the number of relevant preferences. Fine-tuned models are stable across preference-count buckets, varying by no more than 7.0 F1 in either configuration, and improve slightly as more preferences become relevant. Zero-shot models are considerably less stable, varying by up to 11.4 points (Qwen3-8B, *Relevant Preferences & Tools*), and reach their highest scores

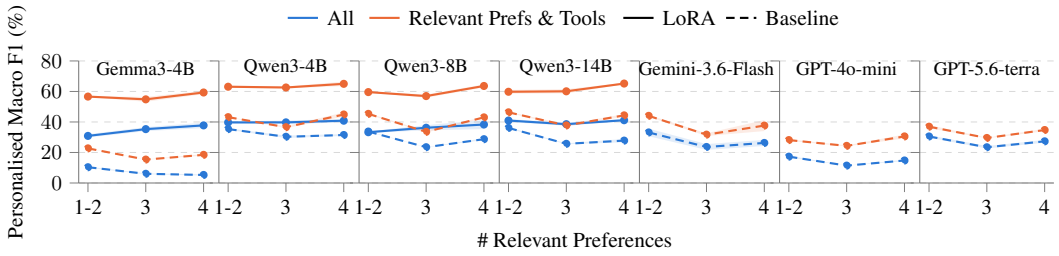


Figure 3: Personalised F1 versus number of relevant user preferences, per model (*All Preferences & Tools* and *Relevant Preferences & Tools* configurations). Fine-tuned models stay stable or improve as more preferences become relevant, whereas zero-shot baselines degrade.

when only 1–2 preferences are relevant. For the fine-tuned models, exact match is more sensitive still, rising by up to 10.9 points from the smallest (1–2) to the largest (4) bucket under *Relevant Preferences & Tools*. Together, this indicates that supervision helps models incorporate *multiple* preferences rather than relying on a few salient ones, and that reliably identifying and applying relevant preferences remains the central challenge for personalised tool use.

Performance varies substantially across subgoals. Fine-grained evaluation reveals a difference between retrieval-oriented and execution-oriented subgoals (Figure 4). Models achieve higher performance on retrieval tasks (Transport, Accommodation, Activities, and Restaurants) than on execution tasks (Booking and Calendar).

Although fine-tuning consistently improves retrieval performance, it provides only limited gains for execution tasks and occasionally reduces performance relative to the zero-shot baseline. LoRA lifts retrieval F1 up to 47.22 for Gemma3-4B, 47.21 for Qwen3-4B, 41.56 for Qwen3-8B, and 48.61 for Qwen3-14B, compared with zero-shot scores below 30 F1 across most retrieval categories.

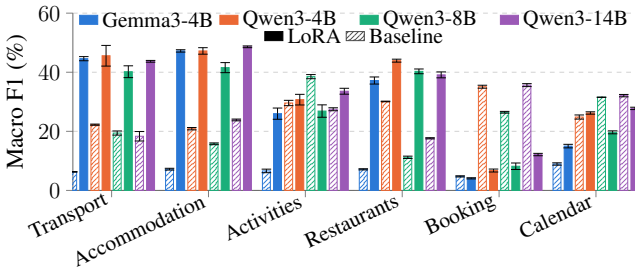


Figure 4: Subgoal-level evaluation on **Tool Tailor**. F1 for each planning subgoal under the *All Preferences & Tools* configuration; retrieval subgoals are consistently easier than execution subgoals (Booking, Calendar).

In contrast, execution tasks remain substantially more challenging: Booking F1 decreases considerably for most models, while Calendar gains remain limited. We attribute this to the synthetic data generation pipeline, in which retrieval tool calls occur far more frequently than actionable ones, biasing the training distribution towards information-seeking calls. This imbalance may also induce a form of catastrophic forgetting, in which fine-tuning shifts behaviour towards retrieval at the expense of pre-existing execution capabilities. Increasing the diversity and frequency of execution trajectories is therefore an important direction for future versions of ToolTailor.

6 Conclusion

We introduce **ToolTailor**, a benchmark for evaluating personalised tool use in task-oriented dialogues that combines realistic travel-planning scenarios with detailed user profiles and multi-tool orchestration. We develop a synthetic data generation pipeline that produces personalised tool-use trajectories at scale for supervised fine-tuning. Training on these trajectories substantially improves personalised tool use, but identifying which user preferences apply remains a key challenge. We further show that models handle retrieval subgoals well but struggle with execution subgoals such as booking, a gap fine-tuning widens rather than closes. Finally, dependency-aware evaluation shows that strict reference matching understates model capability by ignoring valid alternative completion orders. We discuss these and other limitations of ToolTailor in Appendix A. We hope ToolTailor motivates further work on preference identification in agentic settings and the development of more capable task-oriented agents.

Ethical Considerations

Privacy remains one of the main concerns for any personalised system, which has access to user-specific information such as preferences, profiles, and dialogue history (Horvitz & Mulligan, 2015; Yao et al., 2024; Miranda et al., 2025). Such information can contain sensitive details about users and requires careful handling to prevent unintended exposure and misuse. To avoid these risks, ToolTailor does not collect or use real user data. Instead, all user profiles, preferences, and interactions are generated synthetically. This enables controlled evaluation of personalised tool use while avoiding the privacy concerns associated with collecting real user data.

Our experiments investigate trainable approaches, specifically supervised fine-tuning, which in real-world settings would require access to user data. Since our data is synthetic, we do not employ additional privacy-preserving techniques, as protecting user data during training is beyond the scope of this work. In deployed systems, privacy risks may be addressed through training user-specific models (Tan et al., 2024) or inference-time personalisation (Kuang et al., 2024). Exploring how these approaches affect personalised tool use remains an important direction for future work.

Finally, our human validation study was conducted on Prolific with informed consent and compensation above the local minimum wage (Appendix D). Seed scenarios are drawn from public web forums in accordance with their terms of use, and are used only as high-level prompts rather than reproduced in the benchmark.

Reproducibility statement

We made efforts to ensure the reproducibility of our work. We describe the dataset collection procedure in Section 3.2.2 and provide extra information in Appendices B-D. Details of our experimental setup, including the training and inference configurations, hyperparameters, and computational resources used, are discussed in Section 4 and further elaborated in Appendices E-F. We publically release the data and code used in the paper at https://anonymous.4open.science/r/tool-tailor/.

AI use statement

In this work, we used Claude (Anthropic, 2026) for assistance in data annotation and code refinement, as well as text editing for grammar and clarity. We have not used generative AI tools for research ideation, experiment design, data analysis, result interpretation or other purposes. We detail the use of AI for data annotation in Section 3.2.4 and Appendix C.2.6. We have reviewed all AI-assisted work and take full responsibility for the final content of this work, including the text, claims made or artifacts produced.

References

Aelim Ahn, Sooyeon Lee, Hyosun Wang, Chiwan Park, Daeryong Kim, Jihyeon Roh, Kichang Yang, Wonjun Jang, Hwang Woosung, Min Seok Kim, and Jihoon kang. Orchestrationbench: LLM-driven agentic planning and tool use in multi-domain scenarios. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=Oljnxmf4pc.

Pierre Andrews, Amine Benhalloum, Gerard Moreno-Torres Bertran, Matteo Bettini, Amar Budhiraja, Ricardo Silveira Cabral, Virginie Do, Romain Froger, Emilien Garreau, Jean-Baptiste Gaya, Hugo Laurençon, Maxime Lecanu, Kunal Malkan, Dheeraj Mekala, Pierre Ménard, Grégoire Mialon, Ulyana Piterbarg, Mikhail Plekhanov, Mathieu Rita, Andrey Rusakov, Thomas Scialom, Vladislav Vorotilov, Mengjue Wang, and Ian Yu. Are: Scaling up agent environments and evaluations, 2025. URL https://arxiv.org/abs/2509.17158.

Anthropic. Claude code overview, 2026. URL https://code.claude.com/docs/en/overview. Official Claude Code documentation.

Hongru Cai, Yongqi Li, Wenjie Wang, Fengbin Zhu, Xiaoyu Shen, Wenjie Li, and Tat-Seng Chua. Large language models empowered personalized web agents, 2025. URL https://arxiv.org/abs/2410.17236.

Junyi Chen. A survey on large language models for personalized and explainable recommendations, 2023. URL https://arxiv.org/abs/2311.12338.

Google. Gemini 3.6 flash. https://ai.google.dev/gemini-api/docs/models/gemini-3.6-flash, 2026. Accessed: 2026-09-20.

Shibo Hao, Tianyang Liu, Zhen Wang, and Zhiting Hu. Toolkengpt: Augmenting frozen language models with massive tools via tool embeddings. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 45870–45894. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/8fd1a81c882cd45f64958da6284f4a3f-Paper-Conference.pdf.

Yupu Hao, Pengfei Cao, Zhuoran Jin, Huanxuan Liao, Yubo Chen, Kang Liu, and Jun Zhao. Evaluating personalized tool-augmented LLMs from the perspectives of personalization and proactivity. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 21897–21935, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.1064. URL https://aclanthology.org/2025.acl-long.1064/.

Jerry Zhi-Yang He, Sashrika Pandey, Mariah L. Schrum, and Anca Dragan. Context steering: Controllable personalization at inference time, 2025. URL https://arxiv.org/abs/2405.01768.

Pengfei He, Zhenwei Dai, Bing He, Hui Liu, Xianfeng Tang, Hanqing Lu, Juanhui Li, Jiayuan Ding, Subhabrata Mukherjee, Suhang Wang, Yue Xing, Jiliang Tang, and Benoit Dumoulin. TRAJECT-bench:a trajectory-aware benchmark for evaluating agentic tool use. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=TZWnWvsQ0X.

Eric Horvitz and Deirdre Mulligan. Data, privacy, and the greater good. *Science*, 349(6245): 253–255, 2015. doi: 10.1126/science.aac4520. URL https://www.science.org/doi/abs/10.1126/science.aac4520.

Pin-Lun Hsu, Yun Dai, Vignesh Kothapalli, Qingquan Song, Shao Tang, Siyu Zhu, Steven Shimizu, Shivam Sahni, Haowen Ning, Yanning Chen, and Zhipeng Wang. Liger-kernel: Efficient triton kernels for LLM training. In *Championing Open-source DEvelopment in ML Workshop @ ICML25*, 2025. URL https://openreview.net/forum?id=36SjAIT42G.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Nikki Lijing Kuang, Wei Sun, Scott McFaddin, Yian Ma, and Markus Ettl. Towards personalized language models via inference-time human preference optimization. In *Adaptive Foundation Models: Evolving AI for Personalized and Efficient Learning*, 2024. URL https://openreview.net/forum?id=zXz9BuSD0e.

Namjun Lee and Jaekwang Kim. Sealr: Sequential emotion-aware llm-based personalized recommendation system. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '25, pp. 2906–2910, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 9798400715921. doi: 10.1145/3726302.3730249. URL https://doi.org/10.1145/3726302.3730249.

Seongyun Lee, Sue Hyun Park, Seungone Kim, and Minjoon Seo. Aligning to thousands of preferences via system message generalization. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 73783–73829. Curran Associates, Inc., 2024. doi: 10.52202/079017-2347. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/86c9df30129f7663ad4d429b6f80d461-Paper-Conference.pdf.

Lei Li, Yongfeng Zhang, Dugang Liu, and Li Chen. Large language models for generative recommendation: A survey and visionary discussions. In Nicoletta Calzolari, Min-Yen Kan, Veronique Hoste, Alessandro Lenci, Sakriani Sakti, and Nianwen Xue (eds.), *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pp. 10146–10159, Torino, Italia, May 2024. ELRA and ICCL. URL https://aclanthology.org/2024.lrec-main.886/.

Minghao Li, Yingxiu Zhao, Bowen Yu, Feifan Song, Hangyu Li, Haiyang Yu, Zhoujun Li, Fei Huang, and Yongbin Li. API-bank: A comprehensive benchmark for tool-augmented LLMs. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 3102–3116, Singapore, December 2023a. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.187. URL https://aclanthology.org/2023.emnlp-main.187/.

Zhiyu Li, Yanfang Chen, Xuan Zhang, and Xun Liang. Bookgpt: A general framework for book recommendation empowered by large language model. *Electronics*, 12(22), 2023b. ISSN 2079-9292. doi: 10.3390/electronics12224654. URL https://www.mdpi.com/2079-9292/12/22/4654.

Weiwen Liu, Xu Huang, Xingshan Zeng, Xinlong Hao, Shuai Yu, Dexun Li, Shuai Wang, Weinan Gan, Zhengying Liu, Yuanqing Yu, Zezhong Wang, Yuxian Wang, Wu Ning, Yutai Hou, Bin Wang, Chuhan Wu, Xinzhi Wang, Yong Liu, Yasheng Wang, Duyu Tang, Dandan Tu, Lifeng Shang, Xin Jiang, Ruiming Tang, Defu Lian, Qun Liu, and Enhong Chen. ToolACE: Winning the points of LLM function calling. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=8EB8k6DdCU.

Zuxin Liu, Thai Hoang, Jianguo Zhang, Ming Zhu, Tian Lan, Shirley Kokane, Juntao Tan, Weiran Yao, Zhiwei Liu, Yihao Feng, Rithesh Murthy, Liangwei Yang, Silvio Savarese, Juan Carlos Niebles, Huan Wang, Shelby Heinecke, and Caiming Xiong. APIGen: Automated pipeline for generating verifiable and diverse function-calling datasets. In *Advances in Neural Information Processing Systems*, volume 37, pp. 54463–54482, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/61cce86d180b1184949e58939c4f983d-Abstract-Datasets_and_Benchmarks_Track.html. Datasets and Benchmarks Track.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Jiarui Lu, Thomas Holleis, Yizhe Zhang, Bernhard Aumayer, Feng Nan, Felix Bai, Shuang Ma, Shen Ma, Mengyu Li, Guoli Yin, Zirui Wang, and Ruoming Pang. Toolsandbox: A stateful, conversational, interactive evaluation benchmark for llm tool use capabilities, 2024. URL https://arxiv.org/abs/2408.04682.

Hanjia Lyu, Song Jiang, Hanqing Zeng, Yinglong Xia, Qifan Wang, Si Zhang, Ren Chen, Chris Leung, Jiajie Tang, and Jiebo Luo. LLM-rec: Personalized recommendation via prompting large language models. In Kevin Duh, Helena Gomez, and Steven Bethard (eds.), *Findings of the Association for Computational Linguistics: NAACL 2024*, pp. 583–612, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-naacl.39. URL https://aclanthology.org/2024.findings-naacl.39/.

Grégoire Mialon, Clémentine Fourrier, Craig Swift, Thomas Wolf, Yann LeCun, and Thomas Scialom. Gaia: a benchmark for general ai assistants, 2023. URL https://arxiv.org/abs/2311.12983.

Michele Miranda, Elena Sofia Ruzzetti, Andrea Santilli, Fabio Massimo Zanzotto, Sébastien Bratières, and Emanuele Rodolà. Preserving privacy in large language models: A survey on current threats and solutions. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=Ss9MTTN7OL.

Nikita Moghe, Patrick Xia, Jacob Andreas, Jason Eisner, Benjamin Van Durme, and Harsh Jhamtani. Interpreting user requests in the context of natural language standing instructions. In

Kevin Duh, Helena Gomez, and Steven Bethard (eds.), *Findings of the Association for Computational Linguistics: NAACL 2024*, pp. 4043–4060, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-naacl.255. URL https://aclanthology.org/2024.findings-naacl.255/.

Sheshera Mysore, Zhuoran Lu, Mengting Wan, Longqi Yang, Bahareh Sarrafzadeh, Steve Menezes, Tina Baghaee, Emmanuel Barajas Gonzalez, Jennifer Neville, and Tara Safavi. Pearl: Personalizing large language model writing assistants with generation-calibrated retrievers. In Sachin Kumar, Vidhisha Balachandran, Chan Young Park, Weijia Shi, Shirley Anugrah Hayati, Yulia Tsvetkov, Noah Smith, Hannaneh Hajishirzi, Dongyeop Kang, and David Jurgens (eds.), *Proceedings of the 1st Workshop on Customizable NLP: Progress and Challenges in Customizing NLP for a Domain, Application, Group, or Individual (CustomNLP4U)*, pp. 198–219, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.customnlp4u-1.16. URL https://aclanthology.org/2024.customnlp4u-1.16/.

OpenAI. Gpt-4o system card, 2024. URL https://arxiv.org/abs/2410.21276.

OpenAI. Gpt-5.6 system card. https://deploymentsafety.openai.com/gpt-5-6/gpt-5-6.pdf, 2026. Accessed: 2026-09-20.

Aaron Parisi, Yao Zhao, and Noah Fiedel. Talm: Tool augmented language models, 2022. URL https://arxiv.org/abs/2205.12255.

Shishir G. Patil, Tianjun Zhang, Xin Wang, and Joseph E. Gonzalez. Gorilla: Large language model connected with massive apis, 2023. URL https://arxiv.org/abs/2305.15334.

Akshara Prabhakar, Zuxin Liu, Ming Zhu, Jianguo Zhang, Tulika Manoj Awalgaonkar, Shiyu Wang, Zhiwei Liu, Haolin Chen, Thai Hoang, Juan Carlos Niebles, Shelby Heinecke, Weiran Yao, Huan Wang, Silvio Savarese, and Caiming Xiong. APIGen-MT: Agentic pipeline for multi-turn data generation via simulated agent-human interplay. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/5e3661f7fe4c8ac5652d62eb3d3c96ea-Abstract-Datasets_and_Benchmarks_Track.html. Datasets and Benchmarks Track.

Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, Sihan Zhao, Lauren Hong, Runchu Tian, Ruobing Xie, Jie Zhou, Mark Gerstein, Dahai Li, Zhiyuan Liu, and Maosong Sun. ToolLLM: Facilitating large language models to master 16000+ real-world APIs. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=dHng2O0Jjr.

Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 3982–3992, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1410. URL https://aclanthology.org/D19-1410/.

Matthew Renze. The effect of sampling temperature on problem solving in large language models. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 7346–7356, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.432. URL https://aclanthology.org/2024.findings-emnlp.432/.

Alireza Salemi, Sheshera Mysore, Michael Bendersky, and Hamed Zamani. LaMP: When large language models meet personalization. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7370–7392, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.399. URL https://aclanthology.org/2024.acl-long.399/.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools, 2023. URL https://arxiv.org/abs/2302.04761.

Hope Schroeder, Deb Roy, and Jad Kabbara. Just put a human in the loop? investigating LLM-assisted annotation for subjective tasks. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 25771–25795, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.1323. URL https://aclanthology.org/2025.findings-acl.1323/.

Zhaoxuan Tan, Qingkai Zeng, Yijun Tian, Zheyuan Liu, Bing Yin, and Meng Jiang. Democratizing large language models via personalized parameter-efficient fine-tuning. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 6476–6491, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.372. URL https://aclanthology.org/2024.emnlp-main.372/.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, Gaël Liu, Francesco Visin, Kathleen Kenealy, Lucas Beyer, Xiaohai Zhai, Anton Tsitsulin, Robert Busa-Fekete, Alex Feng, Noveen Sachdeva, Benjamin Coleman, Yi Gao, Basil Mustafa, Iain Barr, Emilio Parisotto, David Tian, Matan Eyal, Colin Cherry, Jan-Thorsten Peter, Danila Sinopalnikov, Surya Bhupatiraju, Rishabh Agarwal, Mehran Kazemi, Dan Malkin, Ravin Kumar, David Vilar, Idan Brusilovsky, Jiaming Luo, Andreas Steiner, Abe Friesen, Abhanshu Sharma, Abheesht Sharma, Adi Mayrav Gilady, Adrian Goedeckemeyer, Alaa Saade, Alex Feng, Alexander Kolesnikov, Alexei Bendebury, Alvin Abdagic, Amit Vadi, András György, André Susano Pinto, Anil Das, Ankur Bapna, Antoine Miech, Antoine Yang, Antonia Paterson, Ashish Shenoy, Ayan Chakrabarti, Bilal Piot, Bo Wu, Bobak Shahriari, Bryce Petrini, Charlie Chen, Charline Le Lan, Christopher A. Choquette-Choo, CJ Carey, Cormac Brick, Daniel Deutsch, Danielle Eisenbud, Dee Cattle, Derek Cheng, Dimitris Paparas, Divyashree Shivakumar Sreepathihalli, Doug Reid, Dustin Tran, Dustin Zelle, Eric Noland, Erwin Huizenga, Eugene Kharitonov, Frederick Liu, Gagik Amirkhanyan, Glenn Cameron, Hadi Hashemi, Hanna Klimczak-Plucińska, Harman Singh, Harsh Mehta, Harshal Tushar Lehri, Hussein Hazimeh, Ian Ballantyne, Idan Szpektor, Ivan Nardini, Jean Pouget-Abadie, Jetha Chan, Joe Stanton, John Wieting, Jonathan Lai, Jordi Orbay, Joseph Fernandez, Josh Newlan, Ju yeong Ji, Jyotinder Singh, Kat Black, Kathy Yu, Kevin Hui, Kiran Vodrahalli, Klaus Greff, Linhai Qiu, Marcella Valentine, Marina Coelho, Marvin Ritter, Matt Hoffman, Matthew Watson, Mayank Chaturvedi, Michael Moynihan, Min Ma, Nabila Babar, Natasha Noy, Nathan Byrd, Nick Roy, Nikola Momchev, Nilay Chauhan, Noveen Sachdeva, Oskar Bunyan, Pankil Botarda, Paul Caron, Paul Kishan Rubenstein, Phil Culliton, Philipp Schmid, Pier Giuseppe Sessa, Pingmei Xu, Piotr Stanczyk, Pouya Tafti, Rakesh Shivanna, Renjie Wu, Renke Pan, Reza Rokni, Rob Willoughby, Rohith Vallu, Ryan Mullins, Sammy Jerome, Sara Smoot, Sertan Girgin, Shariq Iqbal, Shashir Reddy, Shruti Sheth, Siim Põder, Sijal Bhatnagar, Sindhu Raghuram Panyam, Sivan Eiger, Susan Zhang, Tianqi Liu, Trevor Yacovone, Tyler Liechty, Uday Kalra, Utku Evci, Vedant Misra, Vincent Roseberry, Vlad Feinberg, Vlad Kolesnikov, Woohyun Han, Woosuk Kwon, Xi Chen, Yinlam Chow, Yuvein Zhu, Zichuan Wei, Zoltan Egyed, Victor Cotruta, Minh Giang, Phoebe Kirk, Anand Rao, Kat Black, Nabila Babar, Jessica Lo, Erica Moreira, Luiz Gustavo Martins, Omar Sanseviero, Lucas Gonzalez, Zach Gleicher, Tris Warkentin, Vahab Mirrokni, Evan Senter, Eli Collins, Joelle Barral, Zoubin Ghahramani, Raia Hadsell, Yossi Matias, D. Sculley, Slav Petrov, Noah Fiedel, Noam Shazeer, Oriol Vinyals, Jeff Dean, Demis Hassabis, Koray Kavukcuoglu, Clement Farabet, Elena Buchatskaya, Jean-Baptiste Alayrac, Rohan Anil, Dmitry, Lepikhin, Sebastian Borgeaud, Olivier Bachem, Armand Joulin, Alek Andreev, Cassidy Hardin, Robert Dadashi, and Léonard Hussenot. Gemma 3 technical report, 2025. URL https://arxiv.org/abs/2503.19786.

Yu-Min Tseng, Yu-Chao Huang, Teng-Yun Hsiao, Wei-Lin Chen, Chao-Wei Huang, Yu Meng, and Yun-Nung Chen. Two tales of persona in LLMs: A survey of role-playing and personalization. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 16612–16631, Miami, Florida, USA, November

2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.969. URL https://aclanthology.org/2024.findings-emnlp.969/.

Tiannan Wang, Meiling Tao, Ruoyu Fang, Huilin Wang, Shuai Wang, Yuchen Eleanor Jiang, and Wangchunshu Zhou. Ai persona: Towards life-long personalization of llms, 2024. URL https://arxiv.org/abs/2412.13103.

Yuxin Wang, Yiran Guo, Yining Zheng, Zhangyue Yin, Shuo Chen, Jie Yang, Jiajun Chen, Xuanjing Huang, and Xipeng Qiu. Familytool: A multi-hop personalized tool use benchmark. *CoRR*, abs/2504.06766, April 2025. URL https://doi.org/10.48550/arXiv.2504.06766.

Zhenting Wang, Qi Chang, Hemani Patel, Shashank Biju, Cheng-En Wu, Quan Liu, Aolin Ding, Alireza Rezazadeh, Ankit Shah, Yujia Bao, and Eugene Siow. MCP-bench: Benchmarking tool-using LLM agents with complex real-world tasks via MCP servers. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=fe8mzHwMxN.

Likang Wu, Zhi Zheng, Zhaopeng Qiu, Hao Wang, Hongchao Gu, Tingjia Shen, Chuan Qin, Chen Zhu, Hengshu Zhu, Qi Liu, et al. A survey on large language models for recommendation. *World Wide Web*, 27(5):60, 2024.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Fan Yang, Zheng Chen, Ziyan Jiang, Eunah Cho, Xiaojiang Huang, and Yanbin Lu. Palr: Personalization aware llms for recommendation, 2023. URL https://arxiv.org/abs/2305.07622.

Yifan Yao, Jinhao Duan, Kaidi Xu, Yuanfang Cai, Zhibo Sun, and Yue Zhang. A survey on large language model (llm) security and privacy: The good, the bad, and the ugly. *High-Confidence Computing*, 4(2):100211, 2024. ISSN 2667-2952. doi: https://doi.org/10.1016/j.hcc.2024.100211. URL https://www.sciencedirect.com/science/article/pii/S266729522400014X.

Da Yin, Faeze Brahman, Abhilasha Ravichander, Khyathi Chandu, Kai-Wei Chang, Yejin Choi, and Bill Yuchen Lin. Agent lumos: Unified and modular training for open-source language agents. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12380–12403, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.670. URL https://aclanthology.org/2024.acl-long.670/.

Peijie Yu, Wei Liu, Yifan Yang, Jinjian Li, Zelong Zhang, Xiao Feng, and feng zhang. Benchmarking LLM tool-use in the wild. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=yz7fL5vfpn.

Aohan Zeng, Mingdao Liu, Rui Lu, Bowen Wang, Xiao Liu, Yuxiao Dong, and Jie Tang. AgentTuning: Enabling generalized agent abilities for LLMs. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 3053–3077, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.181. URL https://aclanthology.org/2024.findings-acl.181/.

Xingshan Zeng, Weiwen Liu, Lingzhi Wang, Liangyou Li, Fei Mi, Yasheng Wang, Lifeng Shang, Xin Jiang, and Qun Liu. ToolACE-MT: Non-autoregressive generation for agentic multi-turn interaction. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=KznJt9Fhjc.

Zhehao Zhang, Ryan A. Rossi, Branislav Kveton, Yijia Shao, Diyi Yang, Hamed Zamani, Franck Dernoncourt, Joe Barrow, Tong Yu, Sungchul Kim, Ruiyi Zhang, Jiuxiang Gu, Tyler Derr, Hongjie Chen, Junda Wu, Xiang Chen, Zichao Wang, Subrata Mitra, Nedim Lipka, Nesreen Ahmed, and Yu Wang. Personalization of large language models: A survey, 2025. URL https://arxiv.org/abs/2411.00027.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. A survey of large language models, 2025. URL https://arxiv.org/abs/2303.18223.

Yuchen Zhuang, Yue Yu, Kuan Wang, Haotian Sun, and Chao Zhang. Toolqa: a dataset for llm question answering with external tools. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2024. Curran Associates Inc.

A LIMITATIONS

Single domain coverage. ToolTailor currently focuses on travel planning, which provides a natural setting for studying personalised tool use in realistic environments, requiring long-horizon planning, multi-tool orchestration and reasoning under user-specific constraints. Nevertheless, including only one domain limits the generality of our findings. Extending the benchmark to other domains such as productivity, event planning, or healthcare would support both the development of resources and models for a wider range of tasks. Our automated data generation pipeline makes it straightforward to produce large-scale training data for new domains with minimal adaptation; constructing reliable evaluation sets, however, would still require careful human validation to ensure the quality and fairness of the benchmark.

Evaluation uses static reference trajectories. ToolTailor evaluates by comparing generated tool calls against references at the argument level rather than executing them, which limits evaluation in two ways. First, distinct tool calls can yield the same outcome yet differ in surface form and therefore in score; semantic argument matching mitigates but does not eliminate this, since reference-based comparison cannot confirm outcome equivalence without execution. Second, evaluation is off-policy: once a model deviates from the reference trajectory, its later actions are scored against states that no longer reflect its own policy, even when the alternative path would complete the task. Dependency-aware matching credits valid alternative calls but cannot substitute for interactive evaluation in an executable environment with dynamic user and tool feedback. Executable tool environments enabling outcome-based, on-policy evaluation are an important direction for future work.

Synthetic data generation introduces biases. Although our data generation pipeline enables scalable creation of complex personalised tool-use trajectories, it may inherit biases from the generation process and the underlying models used for generation. These biases can persist even after human validation and limit the diversity and realism of the resulting scenarios (Schroeder et al., 2025). For example, our pipeline produces retrieval calls far more frequently than execution calls, biasing both training and evaluation towards information-seeking behaviour and contributing to the weaker execution-subgoal performance reported in Section 5. Future work may investigate methods for identifying and mitigating biases introduced during synthetic data generation to improve the reliability and coverage of personalised tool-use benchmarks.

B TOOLS IN TOOLTAILOR

Table 3 lists the full set of tools included in ToolTailor, organised by domain. The benchmark spans six travel-related domains – Attractions, Restaurants, Accommodation, Navigation, Transport, and Utility – comprising 27 tools in total. Within a domain, multiple tools offer overlapping functionality (e.g., Accommodation includes seven distinct booking applications spanning hotels, serviced apartments, and short-term rentals, while Transport separates ground options such as coach and rail from air travel). This redundancy allows personalisation to influence not only how a tool's arguments are filled but also which tool is selected in the first place, rather than evaluating argument filling alone.

Table 4 reports, for each domain and individual tool, the number of instances in which it appears in the curated test set relative to ToolTailor-Full, the corresponding set of complete test trajectories (test / ToolTailor-Full). Because the test set is constructed via balanced sampling over individual tool-calling steps rather than drawn directly from the natural trajectory distribution, per-tool counts are considerably more even than in ToolTailor-Full: for example, the seven Accommodation tools each occur 65–67 times in the test set despite ranging from 71 to 105 occurrences in ToolTailor-Full, and the four ground Transport tools are represented equally (85 instances each) despite occurring between 99 and 371 times in ToolTailor-Full. This balancing ensures that evaluation is not dominated by a small number of high-frequency tools and provides comparable per-tool sample sizes for reliable evaluation.

Table 3: The 27 tools included in **ToolTailor**, grouped into six domains (Attractions, Restaurants, Accommodation, Navigation, Transport, and Utility). Multiple tools within each domain offer overlapping functionality, enabling personalisation to affect tool selection as well as argument filling.

Domain	Tool Name	Functions
Attractions	ActivitiesBooker	search_activities, book_activity
	ExperienceGo	find_experiences, get_experience_info, reserve_experience
Restaurants	DineFinder	find_restaurants, book_table, check_bookings
	TableTurner	search_restaurants, get_restaurant_availability, make_reservation, view_reservations
Accommodation	StayFinder	search_accommodations, book_accommodation, check_bookings, manage_booking
	ElysianCollection	find_properties, reserve_suite, check_bookings, view_itinerary
	HotelHive	search_hotels, reserve_room, check_bookings, get_reservation_details
	PrestigeStays	check_availability, confirm_reservation, check_bookings, manage_booking
	SerenityHotels	check_availability, create_booking, check_bookings, manage_stay
	CottageFinder	find_rentals, book_rental, view_booking
	HomePort	search_rentals, reserve_property, check_bookings, get_booking_info
Navigation	GeoSphere	search_places, get_place_details
	MapView	find_places, get_place_details
	Waypoint	find_points_of_interest, get_location_info
Transport	CoachConnect	find_coaches, reserve_coach_seat, view_reservation
	RailDirect	search_routes, purchase_ticket, lookup_booking
	RailRunner	search_trains, book_train_ticket, get_booking_details
	TrackRoamer	search_transport, book_transport, manage_booking
	AeroBooker	search_flights, book_flight, check_booking
	EmeraldSwift	find_fares, create_booking, check_booking, manage_booking
	GlobalWingAirlines	search_flights, book_flight, check_booking
	QuantumAir	search_flights, book_flight, check_booking, manage_booking
	SparrowFly	find_flights, confirm_booking, check_booking, add_services_to_booking
	SummitJet	search_flights, book_flight, manage_booking, get_booking_number
Utility	Calendar	create_event, list_events, update_event
	Scheduler	schedule_appointment, view_schedule, modify_appointment
	TimeWeaver	add_event_to_calendar, list_events, update_event

C Dataset Construction Details

C.1 Dataset Statistics

The size of each ToolTailor split is reported in the main text (Table 2). The corresponding per-tool breakdown of test instances, contrasting the curated *Test Balanced* set with the complete *Test Full* (ToolTailor-Full) trajectories, is provided in Table 4 (Appendix B).

C.2 Dataset Generation

C.2.1 Models

All training data is generated using Qwen3-8B-Instruct, while the test and dev sets are generated using the larger Qwen3-14B model. We intentionally use a stronger model for evaluation data generation to reduce artifacts associated with weaker generation models. All prompts used for generation can be found in the projects repository at https://anonymous.4open.science/r/tool-tailor/.

C.2.2 Scenario Themes

To encourage diversity in both user requests and task trajectories, we introduce a set of 15 scenario themes (Table 5) that capture common travel-planning situations. Rather than relying solely on unconstrained generation from a small collection of manually collected examples, each generated

Table 4: Number of test instances in which each tool appears, reported for the curated **Tool Tailor** test set and for **Tool Tailor**-Full, broken down by domain and individual tool.

Domain	# inst.	Tool Name	# inst.
Attractions	144/1434	ActivitiesBooker	51/515
		ExperienceGo	93/919
Restaurants	357/612	DineFinder	167/310
		TableTurner	190/302
Accommodation	461/563	StayFinder	66/78
		ElysianCollection	65/71
		HotelHive	66/75
		PrestigeStays	65/78
		SerenityHotels	66/105
		CottageFinder	67/76
		HomePort	66/80
Navigation	219/1283	GeoSphere	73/527
		MapView	73/381
		Waypoint	73/375
Transport	439/856	CoachConnect	85/127
		RailRunner	85/99
		RailDirect	85/160
		TrackRoamer	85/371
		AeroBooker	7/7
		EmeraldSwift	21/21
		GlobalWingAirlines	15/15
		QuantumAir	26/26
		SparrowFly	18/18
		SummitJet	12/12
Utility	325/1023	Calendar	108/310
		TimeWeaver	109/334
		Scheduler	108/379

scenario is conditioned on one of these themes, encouraging coverage of distinct planning requirements and user needs.

For each theme, we manually define one or more natural-language prompts describing the desired travel scenario. These prompts are not intended to specify complete user requests; instead, they provide high-level guidance about the traveller's situation or objectives. Examples include:

- A business traveller with a strict conference schedule seeking leisure activities around work commitments;
- A traveller requiring wheelchair-accessible accommodation and attractions.

Seed scenario generation is performed in a few-shot manner. For each theme, the language model receives (i) a theme prompt and (ii) ten manually collected seed scenarios as demonstrations. The model is then prompted to generate ten additional seed scenarios that satisfy the thematic constraints while remaining distinct from the provided examples. This procedure is repeated multiple times to produce a large set of diverse seed scenarios.

C.2.3 Constraints Used For Generation

Table 6 details the constraint types instantiated during scenario template generation (Section 3.2.2), grouped into two categories: resource constraints, which impose financial or temporal limits, and contextual constraints, which impose geospatial or accessibility requirements. For each generated scenario, the language model instantiates one or more of these constraint types into concrete, scenario-specific details — for example, a fixed budget, a booking deadline, a maximum travel radius, or a wheelchair-accessibility requirement — which the resulting task completion trajectory must satisfy throughout.

Table 5: The 15 scenario themes used to guide seed scenario generation, capturing common travel-planning situations (e.g., accessibility needs, business travel, last-minute cancellations).

Theme	Description
Accessibility	Travelers requiring accessible accommodation, transport, and attractions (e.g., wheelchair access, elevators, minimal walking).
Avoiding Tourist Traps	Users seeking hidden gems, niche attractions, or unconventional destinations instead of popular tourist sites.
Business Travel	Business travelers with fixed work schedules who wish to incorporate leisure activities around professional commitments.
Cancellation	Users whose travel plans have been disrupted (e.g., delayed flights or bad weather) and require immediate replanning.
Closed Venues	Travelers looking for activities during periods when many attractions are unavailable (e.g., Sundays, holidays, late nights).
Environmentalist	Environmentally conscious travelers prioritizing sustainable transport, locally sourced food, and conservation-focused activities.
Event-Centered Travel	Trips organized around a specific event such as a concert or sporting event, requiring coordinated travel logistics.
Extreme Sports	Travelers seeking adventure activities such as paragliding, bungee jumping, or challenging hiking routes.
Food Restrictions	Users with dietary requirements or severe food allergies requiring safe restaurant recommendations.
Budget Travel	Travelers seeking exclusively free or low-cost accommodation, transport, and activities.
Hobby-Focused Travel	Trips centered around personal interests or hobbies, including specialist shops, museums, workshops, conventions, or enthusiast communities.
Multi-City	Complex itineraries involving multiple destinations with coordinated accommodation and transportation planning.
New Traveler	Inexperienced or anxious travelers who require detailed, safe, and easy-to-follow itineraries.
Airport Transfer	Users with short layovers who wish to leave the airport briefly while ensuring timely return for their onward flight.
Working While Traveling	Travelers balancing remote work with sightseeing, requiring reliable Wi-Fi, coworking spaces, and flexible scheduling.

Table 6: The four constraint types instantiated during scenario template generation, grouped into resource constraints (financial, temporal) and contextual constraints (geospatial, accessibility), with example instantiations for each.

Category	Name	Description	Examples
Resource	Financial	Strict monetary limits that force trade-offs	Total budget, a per-person or per-item cap, or tiered spending rules
	Temporal	Restrictions on when or how fast actions can occur	A hard deadline for the task, specific time windows (e.g., 'must be done during business hours'), or a required sequence (e.g., 'Step A must be completed before Step B can begin')
Contextual	Geospatial	Geographical boundaries or location-based rules that the solution must adhere to	Requiring all venues to be within a specific radius, a destination must be reachable by a certain mode of transport (e.g., 'by train'), or that a specific area must be avoided
	Accessibility	Mandate specific accommodations for a user's health, sensory, or mobility needs	Wheelchair ramp access, accommodating severe food allergies (e.g., celiac disease)

C.2.4 SUBGOAL TEMPLATE

Table 7 lists the eight candidate subgoals used to construct scenario-specific task-completion plans, organised into four planning stages: Foundation (clarifying missing trip parameters), Critical Planning (researching and securing approval for transport and accommodation choices), Itinerary Planning (researching activities and dining options and drafting an itinerary), and Execution (making approved bookings and finalising the itinerary). Each subgoal specifies the required planning behaviour, the tool domains it draws on, and whether it requires an explicit user checkpoint, i.e., user approval before the plan may proceed. Five of the eight subgoals require such a checkpoint, reflect-

Table 7: The eight candidate subgoals used to construct task-completion plans in **ToolTailor**, organised into four planning stages. Each subgoal specifies required planning behaviour, available tools, and whether user confirmation is required before continuing.

Stage	Subgoal	Task	User Checkpoint	Tools
Foundation	Clarify Initial Request	Identify missing trip parameters (destination, dates, duration, etc.) through clarification, or research candidate destinations if the destination is intentionally unspecified.	✓	–
Critical Planning	Research & Propose: Long-Distance Transport	Research 2–3 transport options, explain trade-offs, and obtain user approval before proceeding.	✓	Transport
	Research & Propose: Accommodation	Research 2–3 accommodation options, explain trade-offs, and obtain user approval before proceeding.	✓	Accommodation
Itinerary Planning	Research: Activities	Identify suitable attractions and collect logistical information (location, opening hours, booking requirements, duration).		Attractions, Navigation
	Research: Food	Identify restaurants and food venues together with logistical information.		Restaurants
	Propose: Draft Itinerary	Combine researched activities and dining options into a coherent itinerary, distinguishing must-book attractions from optional activities, and obtain user approval.	✓	–
Execution	Execute: Critical Bookings	Execute bookings only for options explicitly approved in previous planning stages using previously returned identifiers.		Transport, Accommodation, Attractions, Restaurants
	Finalise: Itinerary & Confirmations	Generate the final itinerary containing confirmations, booking details, and optionally add events to the user's calendar.	✓	Calendar

Table 8: The six message types constraining communication between the four generation components (Action LLM, User LLM, Tool Simulator, and Task Completion Checker) during task completion trajectory generation.

Type	Description
INFORM	Message conveying information, results, or updates. Can be produced by both the assistant (e.g., presenting tool results or summaries) and the user (e.g., providing missing or corrective information)
CLARIFY	Assistant request for missing information required to proceed
CONFIRM	User decision or feedback in response to an assistant INFORM message, including acceptance, rejection, or modification of proposed actions
TOOL_CALL	Assistant invocation of a tool with fully specified arguments, a Python-style code string
TOOL_OUTPUT	Simulated output returned by the Tool Simulator in response to a TOOL_CALL
COMPLETE_SUBGOAL	Signal indicating that the active subgoal has been completed and generation must be stopped

ing decisions (e.g., choice of transport, accommodation, or final itinerary) that materially affect the rest of the plan. For each scenario, the model selects and adapts the relevant subset of these subgoals to produce the scenario-specific plan used to guide trajectory generation.

C.2.5 Task Completion Trajectories

Each task completion trajectory is simulated through the interaction of four components described in Section 3.2.2 (Algorithm 1). Communication between the components is constrained through six message types (Table 8). The Action LLM may request missing information (CLARIFY), invoke a tool (TOOL_CALL), or communicate information to the user (INFORM). Assistant INFORM messages, in turn, trigger the User LLM to either provide missing information (INFORM) or provide feedback in response (CONFIRM), while TOOL_CALL messages trigger the Tool Simulator to produce a corresponding TOOL_OUTPUT. Following each simulated tool output and user response, the Task Completion Checker is prompted with the chat history, the active subgoal, and its associated stopping criteria (Figure 5) and evaluates whether the active subgoal has been satisfied. If not, it

Algorithm 1: Task Completion Trajectory Generation

Input: Scenario $\mathcal{S}$, Subgoal Plan $\mathcal{G}$, Tool Schemas $\mathcal{T}$, Databases $\mathcal{D}$, History H
Output: New Log Entries L, Tool Episodes E, Updated Subgoal Plan $\mathcal{G}'$

$L \leftarrow \emptyset$;
$E \leftarrow \emptyset$;
for $g \in \mathcal{G}$ **do**
  hint $\leftarrow \perp$;
  role $\leftarrow$ ASSISTANT;
  while ***true*** **do**
    if role $=$ ASSISTANT **then**
      $m \leftarrow$ ActionLLM$(\mathcal{S}, \mathcal{G}, g, \mathcal{T}, H,$ hint$)$;
      hint $\leftarrow \perp$;
      $L \leftarrow L \cup \{m\}$; $H \leftarrow H \cup \{m\}$;
      if $\tau(m) =$ COMPLETE_SUBGOAL **then**
        $\mathcal{G}' \leftarrow$ UpdateSubgoal$(\mathcal{G}, g, m)$;
        break;
      else if $\tau(m) =$ TOOL_CALL **then**
        $e \leftarrow$ ToolSimulator$(m, \mathcal{T}, \mathcal{D}, H)$;
        $E \leftarrow E \cup \{e\}$; $H \leftarrow H \cup \{e\}$;
      else // $\tau(m) \in \{$clarify, inform$\}$
        role $\leftarrow$ USER;
        continue;
    else
      $m \leftarrow$ UserLLM$(\mathcal{S}, \mathcal{G}, g, H)$;
      $L \leftarrow L \cup \{m\}$; $H \leftarrow H \cup \{m\}$;
      role $\leftarrow$ ASSISTANT;
    // completion is checked after every tool output or user response
    $c \leftarrow$ CompletionChecker$(\mathcal{S}, g, H)$;
    if $\tau(c) =$ COMPLETE_SUBGOAL **then**
      $L \leftarrow L \cup \{c\}$; $H \leftarrow H \cup \{c\}$;
      $\mathcal{G}' \leftarrow$ UpdateSubgoal$(\mathcal{G}, g, c)$;
      break;
    else
      hint $\leftarrow$ next_step(c);
return $L, E, \mathcal{G}'$

provides a next-step hint that is shown to the Action LLM during the subsequent generation step; otherwise, it emits a COMPLETE_SUBGOAL message. Although the completion checker is primarily responsible for determining subgoal completion, the Action LLM is also permitted to terminate a subgoal directly. This iterative process continues until all subgoals have been completed and the overall task has been achieved. We use $\tau(m)$ to denote the type of message m, mapping to one of the six categories defined in Table 8.

C.2.6 Expert Trajectory Construction

ToolTailor includes manually verified task completion trajectories for all evaluation instances. Rather than generating each trajectory in a single pass, trajectories are constructed incrementally at the subgoal level. Starting from the initial user message, the model generates the full trajectory one subgoal at a time. The generated dialogue segment is then reviewed and corrected by an expert annotator.

To improve efficiency, the expert annotator is assisted by ClaudeCode, which is used to identify potential inconsistencies (schema inconsistencies, missed relevant preferences, incorrect arguments, etc.) and propose edits. The expert reviews each suggested modification, accepting or rejecting it individually, and manually completes any remaining corrections. The expert remains responsible for the final annotation. The corrected dialogue history is subsequently fed back into the generation pipeline to generate the next subgoal segment. This iterative process continues until all subgoals have been completed, producing a fully validated task completion trajectory for each benchmark instance. Figure 6 illustrates an example of a full trajectory.

Condition 1: Functional Completion

- Analyze the description of the subgoal. Break it down into required actions (e.g., "Find flights" AND "Check budget").
- Scan the `INPUT_CONVERSATION_HISTORY` for `TOOL_CALL` entries or `INFORM` messages that correspond to these actions.
- Verdict: Have all technical steps been executed?

Condition 2: Technical Success

- Look at the `TOOL_OUTPUT` for the relevant calls. Did the tools return valid data (status: "success")?
- If a tool returned an error or "no results found", the subgoal is NOT COMPLETE (it might be failed or require replanning, but it is not successfully finished).
- *Abstraction Principle:* Judge completion at the tool's level of abstraction. Do not require verification of properties the tools cannot represent. Evaluate intent coverage, not semantic exhaustiveness (e.g., when the subgoal requires very specific dishes, the tool might abstract this to a cuisine category, which should still count as a step towards subgoal completion).
- *Facet Coverage Rule:* Some subgoals require coverage across multiple distinct facets (e.g., different activity types, viewpoints, or user intents). When a subgoal description includes multiple facets joined by "and", "as well as", or descriptive examples:
  - A single tool call covering only one facet is NOT sufficient.
  - The subgoal is COMPLETE only if all major facets are addressed, either within one tool call or across multiple tool calls.
  - *Example:* the subgoal "Identify 2–3 activities that fit user's preferences for outdoor activities and historical places" includes two main facets: (1) outdoor activities and (2) historical places. To mark this subgoal as COMPLETE, the model must make at least one tool call to look for 2–3 outdoor activities and another call to search for 2–3 historical places.

Condition 3: User Agreement (the "Checkpoint" Rule)

- Check the `userCheckpoint` field in `INPUT_ACTIVE_SUBGOAL`.
- If `False`: the task is complete as so as all the required tool outputs and messages are successfully generated and logged.
- If `True`:
  - If the subgoal is an information-gathering or clarification task: the task is COMPLETE once the user provides the requested information.
  - If the subgoal is an action, selection, or commitment task: explicit user agreement is required.

Figure 5: Task Completion Checker stopping criteria, specifying how subgoal satisfaction is judged across functional completion, technical success, and user-agreement conditions.

D Human Validation Details

D.1 Annotation Protocol

We validate the test and development sets through three independent human annotation projects: (i) user query validation, (ii) scenario setting validation, and (iii) subgoal identification. The same recruitment and quality control procedure is used across all three projects.

Recruitment Details Annotators are recruited through Prolific and filtered to include only participants whose primary language is English and who have completed a Bachelor's degree or higher.

Main Annotation Study Each assignment contains ten examples, and participants can complete the assignment up to five times. As the tasks include open-ended outputs rather than fixed-label decisions, we collect a single annotation per sample rather than aggregating multiple responses. Annotation quality is ensured through participant screening, hidden control tasks, and expert review of all final outputs. Tables 9–15 represent shortened versions of our instructions and web interface.

Training Phase Before participating in the main annotation study, participants complete a training task consisting of five examples. The training examples follow the same format and annotation procedure as the main task. Participants who meet the required quality threshold (80% or 4/5 examples) proceed to the main annotation phase, while participants who fail are automatically screened out and receive a fixed compensation fee for their participation.

Quality Control Out of the 10 examples, 2 are hidden quality control items. Control examples are sampled without replacement across attempts, ensuring that the same participant does not encounter the same control example more than once. All control examples are constructed using the same format as the main annotation examples and are used to monitor annotation quality throughout the study. Annotators who pass the required quality checks (100% or 2/2 correct answers) are included in the final annotation pool.

User: I'm free from 6–10pm on Tuesday can you help me with a place to eat.

Assistant: `TableTurner.search_restaurants(location="Tokyo, Japan", query="ramen", cuisine="japanese",open_after="18:00")`

Tool Output:

- **Ramen House Tokyo** --- Shinjuku, $$, 4.5★, open 11:00--22:00
- **Tsukiji Ramen Bar** --- Ginza, $$$, 4.7★, open 12:00--23:00
- **Ramen Daiko** --- Shibuya, $, 4.3★, open 10:30--21:30

✓ **Subgoal complete:** `itinerary_2` — Identified three highly-rated ramen shops with detailed logistical information.

Assistant: Here is your finalized itinerary: Tsukiji Ramen Bar (4.7 stars, 456 Ginza Avenue).

User: This looks great. I'm happy to confirm Tsukiji Ramen Bar for dinner. Please go ahead and finalize the bookings.

✓ **Subgoal complete:** `itinerary_3` — User approved the draft itinerary. Approved dining: Tsukiji Ramen Bar.

Assistant: `TableTurner.make_reservation(restaurant_id="rst-4e5f...8a9b", date="2024-02-05", time="19:00", party_size=1, lead_booker_name="Kaito Matsuda")`

Tool Output:

`reservation_id = res-8a9b0c1d-... (success)`

✓ **Subgoal complete:** `execution_1` — Executed booking: `make_reservation` → `res-8a9b0c1d...`

Assistant: `Calendar.create_event(title="Tsukiji Ramen Bar Reservation", start_time="2024-02-05T19:00:00", end_time="2024-02-05T20:00:00", location="456 Ginza Avenue")`

Tool Output:

`event_id = evt_single_travel_350_execution_2_0 (success)`

Assistant: I have successfully booked your reservation at Tsukiji Ramen Bar for 7 PM on Tuesday, February 5th. The confirmation number is `res-8a9b0c1d-...`. I added this booking to your calendar for a reminder.

User: All set. Thanks for the help!

✓ **Subgoal complete:** `execution_2` — User confirmed the Tsukiji Ramen Bar reservation on 2024-02-05.

Figure 6: An example of a full task-completion trajectory in ToolTailor.

Table 9: Annotation instructions for the user query validation project (abridged), covering the task description, decision rules, editing guidelines, and a worked example.

Task

- You are given a user message and a list of user preferences.
- Your task is to check whether the user message is suitable for an assistant to answer.
- A valid user message should:
  - clearly describe the user's request;
  - not explicitly mention information already provided as user preferences;
  - contain only information relevant to completing the task;
  - be natural and grammatically correct.
- If the message is already valid, leave it unchanged.
- If there is a problem, make the smallest possible edit required to fix it.

Decision Rules

1. **Message is already valid**
   The message clearly describes the task and does not reveal user preferences.
2. **Message needs a small fix**
   The message can be corrected by minimal edits, such as removing preference mentions or irrelevant information.
3. **Message should be discarded**
   The message is too vague or cannot be corrected without adding new information.

Editing Guidelines

- Remove preference mentions already provided separately.
- Do not add new information or infer missing details.
- Preserve the original meaning.
- Prefer deletion over rewriting.
- Keep the final message natural and grammatically correct.

Example

User Preferences:

- Prefers cheaper accommodation.
- Avoids luxury hotels.

User Message:
I am planning a trip to Paris and need help finding a cheap hotel because I do not like spending much on accommodation.
Decision: Fix
Corrected Message:
I am planning a trip to Paris and need help finding a hotel.
Explanation:
The message explicitly reveals accommodation preferences already provided in the preference list. The preference information is removed while keeping the original task unchanged.

Table 10: Example annotation interface for the user query validation project, showing the user message, associated preferences, and the decision and editing controls presented to annotators.

Review the user message and user preferences.
A good user message should:

- clearly state what the user needs help with;
- not explicitly mention user preferences;
- contain only relevant information;
- read naturally.

If the message is already correct, leave it unchanged.

User Message

I am planning a trip to Paris and need help finding a hotel.

User Preferences

- *Avoids luxury hotels.*
- *Prefers cheaper accommodation.*

Decision

- ○ Message is already valid
- ○ Message needs a small fix
- ○ Message cannot be fixed and should be discarded

Correct the message (minimal edits only)

I am planning a trip to Paris and need help finding a hotel.

Compensation Participants are compensated at an average rate of £13.19 per hour. Participants who fail the training phase receive a fixed compensation of £1.50 for the user query and subgoal

annotation studies, and £2.00 for the scenario setting annotation study, reflecting the time required for task setup.

Expert Review Following crowdsourced annotation, all annotated samples are reviewed by an expert annotator. The expert resolves inconsistencies, verifies adherence to the annotation guidelines, and produces the final gold-standard annotations used in the test set.

Table 11 reports, for each project, both the crowd correction rate (how often crowd annotators revised the LLM's generated output) and the expert correction rate (how often the expert subsequently revised the crowd-validated output). Crowd correction is substantial across all three projects, ranging from 62.6% (User Query) and 63.7% (Scenario Setting) to 86.3% (Subgoals). Expert correction rate is substantially lower than the crowd correction rate for every project. The scenario setting and user query annotation tasks required relatively few modifications (24.4% each), indicating that project participants generally produced high-quality annotations for these tasks. Subgoal annotation required substantially more expert revisions (62.7%), reflecting the greater complexity of constructing coherent multi-step task trajectories. The expert's edits, however, are concentrated in one predictable gap: annotators tend to miss subgoals implied but not explicitly stated in the query. The expert rarely disagreed with what the crowd kept, but frequently added missing steps, most often execution-phase subgoals (e.g. booking or finalising the itinerary) that annotators left out.

Table 11: Human-validation crowd correction rate (vs. the LLM's proposed annotation) and expert correction rate (vs. the crowd-validated output), for each of the three human-validation projects.

Project	**Crowd Correction**	**Expert correction**
User query	62.6%	24.4%
Scenario setting	63.7% location: 40.0% datetime: 24.3% temporality: 14.7%	24.4% location: 15.8% datetime: 5.5% temporality: 6.1%
Subgoals	86.3% removed: 21.9% added: 414 (avg. 0.84/instance)	62.7% removed: 5.3% added: 193 (avg. 0.62/instance)

D.2 Annotation Projects

User Query Validation The user query annotation project validates the initial user message provided in each scenario. Annotators are shown two pieces of information: (i) the initial user message sent to the assistant, and (ii) a list of user preferences associated with the user profile. The goal is to ensure that the user message clearly specifies the task while avoiding explicit mentions of information that is already presented in the user preferences.

A total of 181 participants were recruited. Following the training and quality control procedure, 73 participants passed and their annotations were included in the final dataset.

Scenario Setting Validation The scenario setting annotation project validates the contextual information associated with each scenario, including location, date, time, and whether the user requires immediate assistance or is planning for a future event. Annotators verify that these attributes are logically consistent with the scenario.

A total of 135 participants were recruited, of which 68 passed the quality checks.

Subgoals The subgoal annotation project identifies the intermediate objectives required to complete each scenario. Annotators determine which subgoals are necessary for successful task completion. We only show participants six out of eight available subgoals. The two remaining ones ("Propose: Draft Itinerary" and "Finalise: Itinerary & Confirmations") are added automatically, whenever their prerequisite subgoals are present. For example, we always add "Propose: Draft Itinerary" if at least one of the other "Itinerary Planning" subgoals is chosen.

A total of 163 participants were recruited, of which 66 passed the quality checks.

Table 12: Annotation instructions for the scenario setting validation project (abridged), covering the task description, decision rules, and a worked example for verifying that a scenario's location, date/time, and task temporality are consistent with the user message.

Task

- You are given a user message and a scenario describing the context of the conversation.
- The scenario contains:
  - **Location**: the user's current location when sending the message;
  - **Date/Time**: the current date and time;
  - **Task Temporality**: whether the user needs help now (*immediate*) or is planning for the future (*planning*).
- Your task is to check whether the scenario is logically consistent with the user message.
- If the scenario is already valid, leave it unchanged.
- If there is a problem, make the smallest possible edit required to fix it.

Decision Rules

1. **Scenario is already valid**
   The location, date/time, and task temporality are consistent with the user message.
2. **Scenario needs a small fix**
   One or more fields contradict the user message and can be corrected with minimal edits.

Editing Guidelines

- Change only fields that clearly contradict the user message.
- Keep plausible values unchanged.
- Check all fields independently; multiple fields may require correction.
- For **planning** tasks:
  - the current date should be before the planned event;
  - the user location should not be the destination location.
- For **immediate** tasks:
  - the location should match where the action takes place;
  - the time should be realistic for the request.
- Prefer minimal changes (e.g., adjust only the month or time when possible).

Example

User Message:
Find me a restaurant that is open now.

Scenario:

- Location: Leeds, UK
- Date/Time: 11 March 2026, 10:00
- Task Temporality: planning

Decision: Fix
Corrected Scenario:

- Location: Leeds, UK
- Date/Time: 11 March 2026, 19:00
- Task Temporality: immediate

Explanation:
The user requests immediate assistance ("open now"), so the task temporality should be changed to *immediate*. The time is adjusted to a realistic evening time for restaurant availability.

E Evaluation Setup

Evaluation Protocol Evaluation is performed by comparing predicted and reference tool calls at the level of *tool-call 4-tuples*: each tool call is decomposed into a set of 4-tuples of the form ⟨`tool`, `function`, `argument name`, `argument value`⟩, allowing partial correctness to be measured even when only a subset of arguments is predicted correctly.

Normalisation Prior to comparison, both predicted and reference tool calls are normalised against the corresponding tool schema. Missing optional arguments are filled with their schema-defined default values, since omitting an optional argument and explicitly supplying its default are functionally equivalent at execution time; this ensures that skipped defaults are not penalised. Argument names are matched exactly after normalisation.

Open-Text Arguments For arguments whose values correspond to free-form natural language (e.g., search queries or user-provided descriptions), exact string matching is replaced with semantic matching. Following Reimers & Gurevych (2019), we encode both the predicted and reference values using the `all-MiniLM-L6-v2`[5] sentence embedding model and compute cosine similarity be-

[5] sentence-transformers/all-MiniLM-L6-v2

Table 13: Example annotation interface for the scenario setting validation project, showing the user message, the scenario fields to be checked, and the decision and editing controls presented to annotators.

Review the user message and scenario.

Check whether each component of the scenario is logically consistent with the user request.

- Correct temporality: immediate vs. planning
- Logical location
- Logical date and time

Only edit fields that clearly contradict the user message.

User Message

I need help booking a hotel in Paris next weekend.

Scenario

- Location: *Paris, France*
- Date/Time: *30 May 2026, 10:00*
- Task Temporality: *planning*

Decision

- ○ Scenario is already valid
- ○ Scenario needs a small fix

Correct the scenario (if needed)

Location

Paris, France

Date/Time

Year: 2026 Month: 05 Day: 30 Time: 10:00

Task Temporality

- ○ Planning
- ○ Immediate

tween the resulting embeddings. A prediction is considered correct if the cosine similarity is at least 0.8; otherwise, it is treated as incorrect. We use a threshold of 0.8 as a cutoff for semantic equivalence, allowing minor paraphrasing while avoiding matches between loosely related values. All remaining argument values, including numerical values, dates, booleans, identifiers, and categorical options, are evaluated using with exact match.

Multiple Tool Orders Because task trajectories explicitly encode dependencies between subgoals, multiple valid execution orders may exist. We account for this with a dependency-aware matching setting: the model still receives the fixed reference dialogue history and predicts a single next tool call, but instead of requiring an exact match against the reference, we compare the prediction against the set of all valid immediate next actions permitted by the dependency graph at that point. This set is determined from the dependency metadata specified in the subgoal template, together with simple heuristics derived from the task structure (e.g., a booking call cannot be made before the required identifier appears in a search result). The prediction is assigned the best matching score among these valid candidates, ranked first by *Function Match* and *Tool Match*, then by macro F1. This allows independent branches to be evaluated in any valid topological order, while preserving the relative ordering of calls within the same dependency chain.

Evaluation Settings We evaluate models under three settings: *All Preferences & All Tools*, *Relevant Preferences*, and *Relevant Preferences & Tools*. These settings isolate the model's ability to incorporate relevant user preferences, select appropriate tools from a larger toolset, and generate correct tool calls. The *Relevant Preferences* and *Relevant Tools* subsets are predefined deterministically from the dataset metadata and user profile:

- *Relevant preferences* consist of two components. First, we include the subset of the user's tool-specific preferences, along with simple heuristics derived from the task structure (e.g., a booking call cannot be made before the required identifier appears on the category of the target tool call. For TRANSPORT and ACCOMMODATION calls, preferences are selected using a curated set of category-specific keywords (e.g., "flight," "layover," "hotel," 'amenity"). For ATTRACTIONS,

Table 14: Annotation instructions for the subgoal validation project (abridged), covering the task description, the full list of candidate subgoals, and a worked example for determining which subgoals are required to complete a given user request.

Task

- You are given a user message describing a travel planning request.
- Your task is to review a list of proposed subgoals required to complete the task.
- Subgoals represent high-level steps that an assistant should complete to solve the user's request.
- Determine which subgoals are necessary and which are unnecessary.

Available Subgoals

1. **Clarify Initial Request**
   Ask for missing key information such as destination, dates, or trip duration.
2. **Long-Distance Transport**
   Research transportation options to the destination.
3. **Accommodation**
   Find accommodation options matching the task requirements.
4. **Activities**
   Find activities and attractions relevant to the user's request.
5. **Food**
   Find restaurants, cafes, or food options relevant to the user's request.
6. **Critical Bookings**
   Complete required bookings after the user approves the plan.

Decision Rules

- Include only subgoals needed to complete the user's request.
- Do not add subgoals for components that are not requested.
- Full trip planning requests usually require multiple components.
- Immediate requests usually do not require clarification or booking unless explicitly requested.
- Booking is required only when the user asks for bookings or when accommodation/transport planning requires it.

Example

User Message:
I am planning a trip to Rome next month. Can you help me find a hotel?

Subgoals:

- Clarify Initial Request
- Long-Distance Transport
- Accommodation
- Activities
- Food
- Critical Bookings

Decisions:

- Clarify Initial Request: Keep
- Long-Distance Transport: Remove
- Accommodation: Keep
- Activities: Remove
- Food: Remove
- Critical Bookings: Keep

Explanation:
The user requests help finding a hotel and provides a vague travel time. Therefore, clarification and accommodation are required. Other components are not necessary because they were not requested.

NAVIGATION, and RESTAURANTS calls, the corresponding activity- or food-related general preferences are included in full. General preferences are excluded for UTILS calls.

- *Relevant Tools* are the subset of the user's personal toolset whose category is permitted by the active subgoal (Table 7); they are therefore determined from the subgoal template rather than from the specific reference tool call. Unlike *Relevant Preferences*, which are filtered based on the app associated with the target tool call, relevant tools reflect the tool categories available at the current planning stage.

F EXPERIMENTAL DETAILS

F.1 MAIN EXPERIMENTAL SETTING

Training Configuration All models are fine-tuned for 4 epochs using AdamW (Loshchilov & Hutter, 2019) with a learning rate of 2×10^{-4}, cosine learning-rate decay, a warm-up ratio of 0.05, weight decay of 0.01, and an effective batch size of 32 examples per optimization step. Training is performed in bfloat16 precision with gradient checkpointing enabled. The maximum sequence

Table 15: Example annotation interface for the subgoal validation project, showing the user message and the keep/remove decision presented to annotators for each candidate subgoal

Review the user message and proposed subgoals.

Determine whether each subgoal is necessary for completing the task.

- Keep subgoals that are required.
- Remove subgoals that are unnecessary or unrelated.

User Message

I need help finding a hotel in Paris for next weekend.

Subgoals

Clarify Initial Request

- ○ Keep
- ○ Remove

Long-Distance Transport

- ○ Keep
- ○ Remove

Accommodation

- ○ Keep
- ○ Remove

Activities

- ○ Keep
- ○ Remove

Food

- ○ Keep
- ○ Remove

Critical Bookings

- ○ Keep
- ○ Remove

length and token budget are both set to 17,408 tokens. Evaluation is performed every 50 training steps, and checkpoints are selected based on performance on the human-validated development split using the slot-wise EM described in Section 4.

LoRA Configuration We use LoRA adapters for parameter-efficient fine-tuning. Unless otherwise stated, all experiments use a LoRA rank of 16 and a LoRA scaling factor (α) of 32. The LoRA dropout rate is set to 0.05. Adapters are applied to the attention and feed-forward projection layers:

- `q_proj`
- `k_proj`
- `v_proj`
- `o_proj`
- `gate_proj`
- `up_proj`
- `down_proj`

Inference Configuration During evaluation, models are provided with the complete dialogue context, user preferences, and tool schemas according to the evaluation setting. Models generate a single next tool call, which is evaluated using the metrics described in Section 4. Each experiment is repeated with three random seeds (42, 123, 456) using `temperature=0.7`, and we report the mean and standard deviation across the three runs. We fix temperature rather than using greedy decoding to capture variance across seeds. The specific temperature value is not critical, as prior work finds no statistically significant effect of sampling temperature on LLM problem-solving performance in the 0–1 range (Renze, 2024). All models are evaluated in a zero-shot setting, since selecting representative demonstrations is difficult given the complexity and diversity of scenarios in ToolTailor, and each example, comprising a full dialogue history and tool schemas, is already long enough that adding even one demonstration would exceed the context window of many of the models we evaluate.

Table 16: **Lora-rank ablation results.** Overall EM and F1 scores for Qwen3-4B model under *All Preferences & Tools* configuration. All scores are in %. Best performance is in **bold**, second best is underlined.

LoRA rank	LoRA α	Trainable params	EM	F1
8	16	16,515,072	12.50 $\pm$ 0.2	28.32 $\pm$ 0.2
16	32	33,030,144	**12.85** $\pm$ 0.3	**29.02** $\pm$ 0.1
32	64	66,060,288	11.71 $\pm$ 0.1	27.73 $\pm$ 0.6

Computational Resources All experiments are conducted on a single NVIDIA RTX 6000 Ada GPU (48 GB). Training time varies depending on the model size and evaluation setting, with each fine-tuning run requiring approximately 3.5–18.1 hours. To enable efficient training of larger models, we apply implementation-specific memory optimizations (e.g., fused cross-entropy kernels; Hsu et al. 2025). Gemma3-4B additionally uses SDPA attention during generation instead of FlashAttention due to framework compatibility with its decoding cache.

F.2 LoRA Ablation

To investigate the effect of adapter capacity, we additionally evaluate LoRA rank on Qwen3-4B under the *All Preferences & All Tools* setting. We compare three commonly used rank-scaling configurations: $(r, \alpha) = (8, 16)$, $(16, 32)$, and $(32, 64)$, keeping the scaling ratio constant across experiments. Compared with the large gains obtained through supervised fine-tuning, increasing either LoRA adapter capacity provides relatively modest improvements and no consistent advantage (Table 16). Thus, for all our experiments, we use the default setting of LoRA rank of 16 and scaling factor α of 32.

G Evaluation Results

This section reports the complete evaluation results underlying Section 4, extending the summary figures and tables in the main text with the full set of metrics and matching criteria.

Table 17 provides the full overall results referenced throughout the main text, including tool match, function match, exact match, F1, precision, and recall for every model and evaluation configuration, alongside the Gemini-3.6-Flash zero-shot reference point.

Table 18 reports the corresponding personalised metrics, computed only over personalised arguments, for every model, method, and evaluation configuration.

Table 19 reports the full schema compliance results discussed in Section 4, breaking overall compliance down into argument-name precision, argument-type precision, categorical-option precision, and the rate of tool calls missing a required argument, for every model and method.

Table 21 complements Figure 4 with full per-subgoal results – exact match, F1, precision, and recall, under both strict and dependency-aware matching – for every model and method under the *All Preferences & Tools* configuration.

Table 20 instead aggregates subgoals by tool type, contrasting *Retrieval* functions (e.g., searching for transport, accommodation, or activities) with *Actionable* functions that modify the external environment (e.g., bookings and calendar updates), providing a coarser view of the retrieval-versus-execution gap discussed in Section 4.

Table 22 and Table 23 report the full metrics – exact match, F1, precision, and recall – underlying Figure 3, grouped by the number of relevant user preferences under the *All Preferences & Tools* and *Relevant Preferences & Tools* configurations respectively, including the Gemini-3.6-Flash zero-shot reference point.

Table 17: **Overall evaluation results on ToolTailor.** All LoRA models use rank-α=(16-32). For each metric, we report results under two matching criteria: strict matching against the reference tool call (left) and dependency-aware matching against any valid tool call permitted by the task structure (right). **Tool**: tool match score. **Func**: function match. **EM**: exact match. **Prec**: precision. **Rec**: recall. All scores are in %. Best performance for each evaluation setting is in **bold**, second best is underlined.

Model	Method	Tool	Func	EM	F1	Prec	Rec
All Preferences & Tools							
Gemma3-4B	Zero-shot	12.38 ± 0.1 / 19.78 ± 0.2	10.30 ± 0.2 / 16.66 ± 0.2	3.32 ± 0.1 / 3.59 ± 0.1	5.98 ± 0.1 / 8.21 ± 0.0	6.04 ± 0.1 / 8.44 ± 0.0	5.95 ± 0.1 / 8.13 ± 0.0
	LoRA	40.59 ± 0.9 / 53.64 ± 0.7	36.69 ± 0.5 / 48.24 ± 0.4	9.81 ± 0.2 / 11.42 ± 0.4	25.13 ± 0.5 / 32.46 ± 0.5	25.09 ± 0.5 / 32.42 ± 0.5	25.18 ± 0.5 / 32.52 ± 0.5
Qwen3-4B	Zero-shot	41.34 ± 0.2 / 57.03 ± 0.3	36.39 ± 0.2 / 48.50 ± 0.2	**13.85** ± 0.1 / **15.50** ± 0.0	25.61 ± 0.3 / 32.91 ± 0.2	25.72 ± 0.3 / 33.10 ± 0.2	25.54 ± 0.2 / 32.81 ± 0.2
	LoRA	**45.75** ± 0.2 / **57.72** ± 0.1	**41.67** ± 0.3 / **53.70** ± 0.2	12.85 ± 0.3 / 14.66 ± 0.4	29.02 ± 0.1 / 37.23 ± 0.3	29.00 ± 0.1 / 37.17 ± 0.3	29.05 ± 0.1 / 37.31 ± 0.3
Qwen3-8B	Zero-shot	37.16 ± 0.2 / 49.83 ± 0.3	28.78 ± 0.2 / 37.28 ± 0.2	13.79 ± 0.1 / 15.32 ± 0.3	21.83 ± 0.1 / 28.06 ± 0.1	21.88 ± 0.1 / 28.11 ± 0.1	21.82 ± 0.1 / 28.09 ± 0.1
	LoRA	40.46 ± 1.2 / 55.92 ± 1.4	37.65 ± 1.1 / 51.60 ± 0.9	10.89 ± 0.7 / 12.12 ± 0.9	25.92 ± 0.9 / 34.68 ± 0.8	25.89 ± 0.9 / 34.59 ± 0.8	25.97 ± 0.9 / 34.80 ± 0.8
Qwen3-14B	Zero-shot	41.22 ± 0.3 / 54.31 ± 0.5	33.27 ± 0.4 / 41.99 ± 0.5	12.32 ± 0.2 / 14.30 ± 0.2	23.50 ± 0.4 / 29.52 ± 0.5	23.95± 0.4 / 30.14 ± 0.5	23.25 ± 0.4 / 29.21 ± 0.5
	LoRA	44.89 ± 1.1 / 57.19 ± 0.8	41.57 ± 1.0 / 52.82 ± 0.3	13.48 ± 0.1 / 15.48 ± 0.2	**29.76** ± 0.5/ **37.65** ± 0.1	**29.73** ± 0.5 / **37.56** ± 0.2	**29.82** ± 0.4 / **37.77** ± 0.1
Gemini-3.6-Flash	Zero-shot	36.90 ± 2.5 / 54.91 ± 2.8	32.88 ± 2.5 / 47.69 ± 3.2	10.52 ± 0.7 / 12.48 ± 0.8	22.02 ± 1.6 / 30.83 ± 2.0	22.93 ± 1.7 / 32.90 ± 2.2	21.45 ± 1.5 / 29.70 ± 1.9
GPT-4o-mini	Zero-shot	21.48 ± 0.4 / 33.88 ± 0.7	15.85 ± 0.3 / 23.45 ± 0.7	6.42 ± 0.1 / 8.04 ± 0.2	11.57 ± 0.3 / 16.79 ± 0.4	11.99 ± 0.3 / 17.61 ± 0.5	11.29 ± 0.3 / 16.25 ± 0.4
GPT-5.6-terra	Zero-shot	34.49 ± 0.7 / 50.95 ± 0.9	30.88 ± 0.7 / 42.93 ± 0.9	9.73 ± 0.4 / 11.61 ± 0.5	20.10 ± 0.4 / 27.09 ± 0.3	20.87 ± 0.4 / 28.10 ± 0.3	19.89 ± 0.3 / 26.80 ± 0.3
Relevant Preferences							
Gemma3-4B	Zero-shot	14.34 ± 0.4 / 20.76 ± 0.4	11.56 ± 0.3 / 16.74 ± 0.3	3.41 ± 0.1 / 3.77 ± 0.1	6.41 ± 0.1 / 8.41 ± 0.2	6.47 ± 0.1 / 8.58 ± 0.2	6.37 ± 0.1 / 8.33 ± 0.2
	LoRA	84.01 ± 0.4 / 84.83 ± 0.6	63.74 ± 0.3 / 67.33 ± 0.3	12.83 ± 0.3 / 14.50 ± 0.4	41.49 ± 0.2 / 45.32 ± 0.3	41.39 ± 0.2 / 45.19 ± 0.3	41.66 ± 0.2 / 45.50 ± 0.3
Qwen3-4B	Zero-shot	45.85 ± 0.0 / 57.11 ± 0.2	39.79 ± 0.2 / 48.22 ± 0.1	13.93 ± 0.1 / 15.58 ± 0.0	27.15 ± 0.1 / 32.58 ± 0.2	27.29 ± 0.1 / 32.79 ± 0.2	27.09 ± 0.1 / 32.49 ± 0.2
	LoRA	85.72 ± 0.5 / 86.91 ± 0.4	67.39 ± 0.9 / 70.20 ± 0.7	15.79 ± 1.0 / 17.15 ± 0.8	44.79 ± 0.8 / 48.10 ± 0.7	44.71 ± 0.8 / 48.02 ± 0.7	44.91 ± 0.8 / 48.22 ± 0.7
Qwen3-8B	Zero-shot	41.69 ± 0.4 / 52.44 ± 0.5	32.22 ± 0.2 / 39.32 ± 0.4	14.42 ± 0.2 / 16.34 ± 0.2	24.12 ± 0.0 / 29.63 ± 0.1	24.18 ± 0.0 / 29.67 ± 0.1	24.11 ± 0.0 / 29.65 ± 0.1
	LoRA	**89.74** ± 0.1 / **89.94** ± 0.1	70.55 ± 0.1 / **73.97** ± 0.3	15.97 ± 0.6 / 17.48 ± 0.6	46.51 ± 0.1 / 50.23 ± 0.2	46.42 ± 0.1 / 50.14 ± 0.2	46.65 ± 0.1 / 50.39 ± 0.2
Qwen3-14B	Zero-shot	47.18 ± 0.2 / 57.45 ± 0.2	38.49 ± 0.3 / 44.99 ± 0.1	13.32 ± 0.2 / 15.32 ± 0.1	26.67 ± 0.2 / 31.80 ± 0.1	27.38± 0.2 / 32.56 ± 0.1	26.29 ± 0.2 / 31.40 ± 0.0
	LoRA	87.97 ± 0.1 / 88.52 ± 0.1	**71.22** ± 0.3 / 73.73 ± 0.4	**16.74** ± 0.4 / **18.44** ± 0.8	**47.68** ± 0.6 / **51.11** ± 0.8	**47.56** ± 0.6 / **50.99** ± 0.8	**47.88** ± 0.6 / **51.30** ± 0.8
Gemini-3.6-Flash	Zero-shot	47.42 ± 2.6 / 59.03 ± 2.1	41.81 ± 2.0 / 51.11 ± 2.2	10.87 ± 0.1 / 12.63 ± 0.2	26.32 ± 1.1 / 32.63 ± 1.3	27.86 ± 1.2 / 34.89 ± 1.4	25.36 ± 1.1 / 31.36 ± 1.3
GPT-4o-mini	Zero-shot	24.78 ± 0.2 / 35.10 ± 0.4	17.87 ± 0.0 / 23.72 ± 0.3	7.20 ± 0.1 / 8.69 ± 0.3	12.82 ± 0.1 / 17.02 ± 0.3	13.22 ± 0.1 / 17.71 ± 0.3	12.54 ± 0.1 / 16.54 ± 0.3
GPT-5.6-terra	Zero-shot	44.20 ± 0.8 / 54.93 ± 0.7	38.79 ± 0.5 / 46.81 ± 0.7	11.05 ± 0.3 / 13.14 ± 0.1	23.81 ± 0.5 / 29.31 ± 0.4	25.02 ± 0.6 / 30.85 ± 0.5	23.38 ± 0.5 / 28.79 ± 0.4
Relevant Preferences & Tools							
Gemma3-4B	Zero-shot	40.26 ± 0.3 / 47.07 ± 0.1	29.51 ± 0.2 / 33.47 ± 0.3	5.24 ± 0.1 / 6.00 ± 0.1	14.49 ± 0.0 / 17.63 ± 0.1	14.99 ± 0.0 / 18.17 ± 0.1	14.16 ± 0.0 / 17.30 ± 0.1
	LoRA	76.85 ± 0.2 / 84.15 ± 0.3	64.45 ± 0.2 / 69.26 ± 0.4	11.79 ± 0.2 / 13.67 ± 0.3	40.25 ± 0.0 / 45.06 ± 0.2	40.17 ± 0.0 / 44.97 ± 0.2	40.42 ± 0.0 / 45.24 ± 0.2
Qwen3-4B	Zero-shot	59.80 ± 0.3 / 69.39 ± 0.2	51.48 ± 0.3 / 58.27 ± 0.3	14.15 ± 0.1 / 15.50 ± 0.1	32.14 ± 0.3 / 37.10 ± 0.2	32.62 ± 0.3 / 37.59 ± 0.2	31.83 ± 0.2 / 36.79 ± 0.2
	LoRA	79.97 ± 0.2 / 86.64 ± 0.1	**68.41** ± 0.3 / **73.40** ± 0.1	**14.81** ± 0.3 / **16.74** ± 0.3	**45.52** ± 0.4/ **50.57** ± 0.2	**45.45** ± 0.4 / **50.45** ± 0.2	**45.66** ± 0.4 / **50.75** ± 0.2
Qwen3-8B	Zero-shot	57.39 ± 0.3 / 67.92 ± 0.3	47.97 ± 0.3 / 54.40 ± 0.3	14.28 ± 0.3 / 16.24 ± 0.3	31.38 ± 0.2 / 36.89 ± 0.2	31.82 ± 0.3 / 37.30 ± 0.2	31.08 ± 0.2 / 36.64 ± 0.2
	LoRA	76.89 ± 0.2 / 84.95 ± 0.1	66.61 ± 0.1 / 73.06 ± 0.4	13.30 ± 0.4 / 15.72 ± 0.2	42.94 ± 0.2 / 49.00 ± 0.3	42.84 ± 0.2 / 48.84 ± 0.3	43.12 ± 0.2 / 49.23 ± 0.3
Qwen3-14B	Zero-shot	62.90 ± 0.2 / 73.75 ± 0.2	53.78 ± 0.2 / 61.02 ± 0.3	14.46 ± 0.1 / 16.40 ± 0.2	34.85 ± 0.1 / 40.58 ± 0.1	35.92 ± 0.1 / 41.72 ± 0.2	34.11 ± 0.1 / 39.81 ± 0.1
	LoRA	**80.03** ± 0.1 / **87.19** ± 0.0	67.57 ± 0.2 / 73.16 ± 0.3	13.48 ± 0.3 / 15.99 ± 0.3	44.22 ± 0.0 / 49.88 ± 0.1	44.15 ± 0.1 / 49.78 ± 0.1	44.35 ± 0.0 / 50.03 ± 0.1
Gemini-3.6-Flash	Zero-shot	59.60 ± 1.8 / 69.28 ± 1.4	50.25 ± 1.6 / 57.76 ± 1.5	11.44 ± 0.1 / 13.50 ± 0.2	30.90 ± 1.1 / 36.66 ± 1.0	33.22 ± 1.2 / 39.43 ± 1.2	29.44 ± 1.0 / 35.00 ± 0.9
GPT-4o-mini	Zero-shot	45.14 ± 0.5 / 56.54 ± 0.7	35.43 ± 0.3 / 41.65 ± 0.6	9.16 ± 0.2 / 10.85 ± 0.2	23.01 ± 0.1 / 28.09 ± 0.3	24.41 ± 0.0/ 29.80 ± 0.3	22.07 ± 0.1 / 26.94 ± 0.3
GPT-5.6-terra	Zero-shot	48.32 ± 1.1 / 58.80 ± 1.1	40.87 ± 0.9 / 48.58 ± 0.8	11.20 ± 0.3 / 12.93 ± 0.3	24.91 ± 0.7 / 30.50 ± 0.6	25.88 ± 0.7 / 31.61 ± 0.6	24.52 ± 0.7 / 30.07 ± 0.6

Table 18: **Personalised metrics results on ToolTailor.** All LoRA models use rank-α=(16-32). For each metric, we report results under two matching criteria: strict matching against the reference tool call (left) and dependency-aware matching against any valid tool call permitted by the task structure (right). **EM**: exact match. **Prec**: precision. **Rec**: recall. All scores are in %. Best performance for each evaluation setting is in **bold**, second best is underlined.

Model	Method	EM	F1	Prec	Rec
		All Preferences & Tools			
Gemma3-4B	Zero-shot	6.30 ± 0.1 / 7.46 ± 0.1	7.25 ± 0.1 / 9.21 ± 0.1	7.31 ± 0.1 / 9.47 ± 0.1	7.22 ± 0.1 / 9.11 ± 0.1
	LoRA	22.48 ± 0.8 / 26.78 ± 1.1	30.48 ± 0.6 / 38.86 ± 0.7	30.47 ± 0.6 / 38.85 ± 0.7	30.52 ± 0.6 / 38.90 ± 0.7
Qwen3-4B	Zero-shot	26.07 ± 0.4 / 31.00 ± 0.3	29.92 ± 0.2 / 37.40 ± 0.2	29.95 ± 0.2 / 37.65 ± 0.2	29.90 ± 0.2 / 37.29 ± 0.2
	LoRA	28.62 ± 0.9 / 34.12 ± 1.1	36.21 ± 0.4 / **45.81** ± 0.3	36.20 ± 0.4 / **45.80** ± 0.3	36.26 ± 0.4 / **45.88** ± 0.3
Qwen3-8B	Zero-shot	25.72 ± 0.3 / 31.12 ± 0.4	26.81 ± 0.2 / 33.11 ± 0.2	26.85 ± 0.2 / 33.20 ± 0.1	26.78 ± 0.2 / 33.06 ± 0.2
	LoRA	24.47 ± 1.2 / 30.88 ± 1.4	31.91 ± 1.2 / 42.68 ± 1.1	31.90 ± 1.2 / 42.67 ± 1.1	31.95 ± 1.2 / 42.73 ± 1.0
Qwen3-14B	Zero-shot	25.47 ± 0.2 / 30.00 ± 0.3	28.30 ± 0.4 / 34.17 ± 0.4	28.66 ± 0.4 / 34.66 ± 0.4	28.13 ± 0.3 / 33.94 ± 0.4
	LoRA	**29.25** ± 0.3 / **35.00** ± 0.3	**36.49** ± 0.8 / 45.57 ± 0.3	**36.47** ± 0.8 / 45.54 ± 0.3	**36.58** ± 0.8/ 45.65 ± 0.3
Gemini-3.6-Flash	Zero-shot	22.11 ± 1.5 / 27.27 ± 1.7	26.13 ± 2.0 / 33.82 ± 2.4	27.35 ± 2.1 / 36.20 ± 2.7	25.62 ± 1.9 / 32.91 ± 2.3
GPT-4o-mini	Zero-shot	12.54 ± 0.2 / 16.68 ± 0.3	13.57 ± 0.3 / 18.77 ± 0.4	13.83 ± 0.3 / 19.36 ± 0.5	13.43 ± 0.3 / 18.47 ± 0.4
GPT-5.6-terra	Zero-shot	21.42 ± 0.2 / 26.05 ± 0.3	25.10 ± 0.2 / 32.24 ± 0.2	25.72 ± 0.3 / 33.11 ± 0.3	24.84 ± 0.2 / 31.84 ± 0.2
		Relevant Preferences			
Gemma3-4B	Zero-shot	6.53 ± 0.2 / 7.85 ± 0.3	7.74 ± 0.2 / 9.52 ± 0.3	7.91 ± 0.2 / 9.82 ± 0.2	7.69 ± 0.2 / 9.42 ± 0.3
	LoRA	36.24 ± 0.2 / 38.91 ± 0.3	52.44 ± 0.3 / 55.77 ± 0.3	52.42 ± 0.3 / 55.75 ± 0.3	52.53 ± 0.3 / 55.86 ± 0.3
Qwen3-4B	Zero-shot	27.62 ± 0.1 / 32.31 ± 0.4	32.20 ± 0.1 / 37.86 ± 0.3	32.35 ± 0.1 / 38.13 ± 0.3	32.13 ± 0.1 / 37.74 ± 0.3
	LoRA	39.89 ± 0.5 / 42.67 ± 0.4	55.50 ± 0.6 / 58.42 ± 0.5	55.46 ± 0.6 / 58.38 ± 0.5	55.60 ± 0.6 / 58.52 ± 0.5
Qwen3-8B	Zero-shot	27.13 ± 0.1 / 32.22 ± 0.1	29.01 ± 0.1 / 34.62 ± 0.2	29.07 ± 0.1 / 34.74 ± 0.2	28.97 ± 0.1 / 34.57 ± 0.2
	LoRA	42.36 ± 0.6 / 45.18 ± 0.6	58.20 ± 0.2 / 61.49 ± 0.3	58.16 ± 0.2 / 61.47 ± 0.3	58.30 ± 0.2 / 61.58 ± 0.3
Qwen3-14B	Zero-shot	27.35 ± 0.3 / 31.90 ± 0.1	31.35 ± 0.3 / 36.38 ± 0.1	32.04 ± 0.3 / 37.13 ± 0.1	31.03 ± 0.3 / 36.03 ± 0.1
	LoRA	**43.83** ± 1.0 / **46.81** ± 1.4	**59.43** ± 0.9 / **62.27** ± 1.1	**59.38** ± 0.9 / **62.22** ± 1.1	**59.55** ± 0.9 / **62.39** ± 1.1
Gemini-3.6-Flash	Zero-shot	25.09 ± 0.7 / 29.63 ± 0.9	31.08 ± 1.2 / 36.63 ± 1.5	33.12 ± 1.2 / 39.14 ± 1.5	30.22 ± 1.1 / 35.62 ± 1.4
GPT-4o-mini	Zero-shot	13.66 ± 0.2 / 17.42 ± 0.3	14.87 ± 0.2 / 19.18 ± 0.3	15.14 ± 0.2 / 19.79 ± 0.2	14.73 ± 0.2 / 18.91 ± 0.3
GPT-5.6-terra	Zero-shot	25.15 ± 0.6 / 29.70 ± 0.5	29.87 ± 0.6 / 35.51 ± 0.6	30.81 ± 0.7 / 36.64 ± 0.6	29.47 ± 0.6 / 35.03 ± 0.6
		Relevant Preferences & Tools			
Gemma3-4B	Zero-shot	13.20 ± 0.1 / 15.85 ± 0.1	17.76 ± 0.1 / 20.71 ± 0.1	18.10 ± 0.1 / 21.07 ± 0.2	17.60 ± 0.0 / 20.54 ± 0.1
	LoRA	35.12 ± 0.5 / 39.40 ± 0.3	51.33 ± 0.3 / 56.05 ± 0.3	51.32 ± 0.3 / 56.09 ± 0.3	51.42 ± 0.3 / 56.12 ± 0.3
Qwen3-4B	Zero-shot	30.43 ± 0.2 / 35.20 ± 0.2	37.62 ± 0.2 / 43.16 ± 0.1	38.06 ± 0.1 / 43.60 ± 0.1	37.40 ± 0.2 / 42.93 ± 0.1
	LoRA	**41.91** ± 0.5 / **46.46** ± 0.1	**57.41** ± 0.5 / **62.43** ± 0.1	**57.38** ± 0.5 / **62.40** ± 0.1	**57.51** ± 0.5 / **62.53** ± 0.1
Qwen3-8B	Zero-shot	31.19 ± 0.1 / 36.69 ± 0.2	37.40 ± 0.2 / 43.30 ± 0.2	37.93 ± 0.2 / 43.84 ± 0.2	37.13 ± 0.1 / 43.03 ± 0.1
	LoRA	37.92 ± 1.2 / 43.24 ± 1.2	54.03 ± 0.3 / 60.05 ± 0.6	54.02 ± 0.3 / 60.04 ± 0.6	54.13 ± 0.3 / 60.15 ± 0.6
Qwen3-14B	Zero-shot	31.49 ± 0.2 / 36.71 ± 0.2	39.24 ± 0.2 / 45.21 ± 0.3	40.44 ± 0.3 / 46.45 ± 0.3	38.68 ± 0.2 / 44.65 ± 0.3
	LoRA	39.95 ± 0.3 / 44.81 ± 0.1	55.28 ± 0.2 / 60.64 ± 0.1	55.25 ± 0.2 / 60.63 ± 0.1	55.38 ± 0.2 / 60.75 ± 0.1
Gemini-3.6-Flash	Zero-shot	27.04 ± 0.6 / 31.63 ± 0.3	35.32 ± 0.8 / 40.74 ± 0.5	38.55 ± 0.9 / 44.26 ± 0.5	34.00 ± 0.8 / 39.33 ± 0.4
GPT-4o-mini	Zero-shot	18.91 ± 0.1 / 23.92 ± 0.3	24.89 ± 0.2 / 30.26 ± 0.5	26.58 ± 0.2 / 32.05 ± 0.5	24.12 ± 0.2 / 29.46 ± 0.5
GPT-5.6-terra	Zero-shot	25.27 ± 0.6 / 30.37 ± 0.5	31.01 ± 0.7 / 37.14 ± 0.6	31.90 ± 0.6 / 38.15 ± 0.6	30.60 ± 0.7 / 36.68 ± 0.6

Table 19: **Schema compliance results on ToolTailor.** All LoRA models use rank-α=(16-32). Results are computed using strict matching against the reference tool call. **Overall**: percentage of predicted arguments that are fully schema-compliant (have a valid name, correct type, and, if categorical, a valid option value). **Name Prec**: percentage of arguments with correct names. **Type Prec**: percentage of arguments with correct value type. **Option Prec**: percentage of categorical arguments with values from the allowed list of options. **Missing Req**: percentage of tool calls missing at least one required argument. All scores are in %. Best performance for each evaluation setting is in **bold**, second best is underlined.

Model	Method	Overall	Name Prec	Type Prec	Option Prec	Missing Req
All Preferences & Tools						
Gemma3-4B	Zero-shot	39.59 ± 0.5	44.28 ± 0.5	43.48 ± 0.5	40.72 ± 0.7	54.40 ± 0.6
	LoRA	80.72 ± 0.2	98.96 ± 0.1	81.87 ± 0.2	95.51 ± 0.4	0.61 ± 0.1
Qwen3-4B	Zero-shot	77.96 ± 0.2	83.55 ± 0.1	82.34 ± 0.1	70.20 ± 0.3	15.34 ± 0.1
	LoRA	81.37 ± 0.3	**99.73** ± 0.1	81.78 ± 0.2	97.53 ± 0.1	**0.20** ± 0.1
Qwen3-8B	Zero-shot	77.81 ± 0.4	84.02 ± 0.3	81.12 ± 0.4	77.29 ± 0.2	15.72 ± 0.5
	LoRA	82.38 ± 0.2	99.35 ± 0.1	83.50 ± 0.3	94.64 ± 0.4	0.35 ± 0.1
Qwen3-14B	Zero-shot	**82.42** ± 0.1	87.89 ± 0.1	**86.12** ± 0.1	79.57 ± 0.3	11.61 ± 0.1
	LoRA	82.21 ± 0.1	99.35 ± 0.2	82.72 ± 0.1	**97.70** ± 0.1	0.51 ± 0.1
Gemini-3.6-Flash	Zero-shot	79.15 ± 0.4	85.92 ± 0.3	79.98 ± 0.4	82.09 ± 0.3	14.07 ± 0.3
GPT-4o-mini	Zero-shot	63.44 ± 0.7	65.20 ± 0.7	65.02 ± 0.6	64.23 ± 0.6	34.02 ± 0.7
GPT-5.6-terra	Zero-shot	78.16 ± 0.5	85.15 ± 0.5	81.01 ± 0.4	72.42 ± 0.3	20.84 ± 0.7
Relevant Preferences						
Gemma3-4B	Zero-shot	39.84 ± 0.2	44.64 ± 0.2	43.96 ± 0.2	42.53 ± 0.6	53.60 ± 0.5
	LoRA	81.11 ± 0.2	98.53 ± 0.2	82.41 ± 0.1	95.21 ± 0.6	0.82 ± 0.2
Qwen3-4B	Zero-shot	78.01 ± 0.1	83.66 ± 0.2	82.61 ± 0.1	70.08 ± 0.4	14.95 ± 0.2
	LoRA	**83.70** ± 0.1	**99.82** ± 0.0	84.31 ± 0.0	**96.57** ± 0.4	**0.10** ± 0.0
Qwen3-8B	Zero-shot	77.75 ± 0.1	84.07 ± 0.3	80.97 ± 0.2	76.24 ± 0.3	15.62 ± 0.3
	LoRA	82.40 ± 0.4	99.33 ± 0.1	83.50 ± 0.4	94.45 ± 0.5	0.45 ± 0.1
Qwen3-14B	Zero-shot	82.85 ± 0.5	88.44 ± 0.5	**85.83** ± 0.5	78.42 ± 0.3	11.97 ± 0.3
	LoRA	82.63 ± 0.0	99.48 ± 0.1	83.37 ± 0.2	96.42 ± 0.1	0.35 ± 0.1
Gemini-3.6-Flash	Zero-shot	80.78 ± 0.4	86.95 ± 0.5	81.45 ± 0.4	83.85 ± 0.3	13.07 ± 0.5
GPT-4o-mini	Zero-shot	64.65 ± 0.8	66.40 ± 0.8	66.10 ± 0.8	65.49 ± 0.5	32.94 ± 0.8
GPT-5.6-terra	Zero-shot	78.90 ± 0.5	85.95 ± 0.1	81.56 ± 0.4	73.69 ± 0.2	20.40 ± 0.4
Relevant Preferences & Tools						
Gemma3-4B	Zero-shot	57.45 ± 0.2	61.99 ± 0.3	60.77 ± 0.4	54.51 ± 0.3	37.49 ± 0.4
	LoRA	79.91 ± 0.4	98.30 ± 0.2	81.55 ± 0.4	92.09 ± 0.8	1.33 ± 0.4
Qwen3-4B	Zero-shot	82.76 ± 0.2	88.87 ± 0.2	85.76 ± 0.2	72.65 ± 0.3	11.46 ± 0.1
	LoRA	82.32 ± 0.0	**99.76** ± 0.1	82.77 ± 0.1	**97.65** ± 0.1	**0.22** ± 0.1
Qwen3-8B	Zero-shot	87.24 ± 0.3	93.83 ± 0.4	89.81 ± 0.4	80.23 ± 0.4	6.10 ± 0.4
	LoRA	82.91 ± 0.3	99.35 ± 0.2	83.88 ± 0.3	95.05 ± 0.2	0.41 ± 0.1
Qwen3-14B	Zero-shot	**89.00** ± 0.4	96.07 ± 0.4	**92.02** ± 0.4	79.12 ± 0.2	3.90 ± 0.5
	LoRA	83.08 ± 0.2	99.06 ± 0.1	83.77 ± 0.2	95.77 ± 0.2	0.86 ± 0.0
Gemini-3.6-Flash	Zero-shot	85.39 ± 0.4	91.19 ± 0.4	86.41 ± 0.5	86.38 ± 0.7	8.81 ± 0.4
GPT-4o-mini	Zero-shot	76.87 ± 0.4	79.89 ± 0.4	78.03 ± 0.5	73.99 ± 0.1	20.07 ± 0.5
GPT-5.6-terra	Zero-shot	77.65 ± 0.5	84.77 ± 0.6	80.00 ± 0.6	72.11 ± 0.7	18.05 ± 0.6

Table 20: **Evaluation results by tool type in ToolTailor under the *All Preferences & Tools* configuration.** All LoRA models use rank-α=(16-32). For each metric, we report results under two matching criteria: strict matching against the reference tool call (left) and dependency-aware matching against any valid tool call permitted by the task structure (right). **Prec**: precision. **Rec**: recall. All scores are in %. Best performance for each evaluation setting is in **bold**, second best is underlined.

Model	Method	EM	F1	Prec	Rec
		Retrieval			
Gemma3-4B	Zero-shot	4.54 ± 0.2 / 4.61 ± 0.2	6.80 ± 0.2 / 8.84 ± 0.1	6.87 ± 0.2 / 9.22 ± 0.1	6.77 ± 0.2 / 8.67 ± 0.1
	LoRA	15.16 ± 0.3 / 17.29 ± 0.5	37.05 ± 0.7 / 47.63 ± 0.6	37.02 ± 0.7 / 47.60 ± 0.6	37.11 ± 0.7 / 47.69 ± 0.6
Qwen3-4B	Zero-shot	17.64 ± 0.2 / 18.27 ± 0.1	26.67 ± 0.2 / 32.51 ± 0.3	26.67 ± 0.2 / 32.60 ± 0.3	26.68 ± 0.2 / 32.46 ± 0.3
	LoRA	18.33 ± 0.5 / **20.34** ± 0.6	**40.77** ± 0.3 / **52.50** ± 0.5	**40.75** ± 0.3 / **52.42** ± 0.5	**40.81** ± 0.3 / **52.59** ± 0.5
Qwen3-8B	Zero-shot	18.43 ± 0.2 / 19.13 ± 0.3	22.18 ± 0.3 / 25.15 ± 0.1	22.21 ± 0.3 / 25.20 ± 0.1	22.15 ± 0.3 / 25.12 ± 0.1
	LoRA	15.67 ± 0.8 / 16.62 ± 1.0	36.34 ± 1.4 / 48.05 ± 1.3	36.33 ± 1.4 / 47.97 ± 1.3	36.37 ± 1.4 / 48.16 ± 1.3
Qwen3-14B	Zero-shot	14.90 ± 0.2 / 15.79 ± 0.2	21.95 ± 0.5 / 25.67 ± 0.4	22.40 ± 0.6 / 26.42 ± 0.4	21.70 ± 0.5 / 25.27 ± 0.4
	LoRA	**18.49** ± 0.1 / **20.34** ± 0.1	40.05 ± 0.7 / 50.43 ± 0.2	40.03 ± 0.7 / 50.36 ± 0.2	40.10 ± 0.6 / 50.53 ± 0.2
Gemini-3.6-Flash	Zero-shot	11.38 ± 0.7 / 12.04 ± 0.7	20.46 ± 1.8 / 28.28 ± 2.2	21.65 ± 1.9 / 31.32 ± 2.4	19.70 ± 1.7 / 26.61 ± 2.1
GPT-4o-mini	Zero-shot	7.88 ± 0.1 / 8.29 ± 0.0	10.35 ± 0.4 / 13.04 ± 0.5	10.65 ± 0.4 / 13.79 ± 0.6	10.15 ± 0.3 / 12.58 ± 0.4
GPT-5.6-terra	Zero-shot	11.19 ± 0.7 / 11.88 ± 0.6	19.20 ± 0.5 / 24.76 ± 0.5	20.24 ± 0.5 / 26.13 ± 0.5	18.80 ± 0.5 / 24.17 ± 0.5
		Actionable			
Gemma3-4B	Zero-shot	1.77 ± 0.1 / 2.58 ± 0.3	6.17 ± 0.2 / 9.54 ± 0.3	6.22 ± 0.2 / 9.51 ± 0.3	6.12 ± 0.2 / 9.61 ± 0.3
	LoRA	1.56 ± 0.1 / 2.58 ± 0.5	7.78 ± 0.3 / 10.56 ± 0.7	7.74 ± 0.3 / 10.49 ± 0.7	7.84 ± 0.3 / 10.66 ± 0.8
Qwen3-4B	Zero-shot	10.25 ± 0.1 / 14.60 ± 0.3	31.63 ± 0.4 / 44.41 ± 0.3	32.04 ± 0.4 / 44.88 ± 0.3	31.39 ± 0.4 / 44.19 ± 0.4
	LoRA	5.30 ± 0.7 / 7.26 ± 0.4	13.32 ± 0.4 / 16.67 ± 0.3	13.30 ± 0.4 / 16.61 ± 0.3	13.34 ± 0.4 / 16.74 ± 0.3
Qwen3-8B	Zero-shot	8.35 ± 0.0 / 12.15 ± 0.3	28.17 ± 0.2 / 43.37 ± 0.1	28.27 ± 0.2 / 43.42 ± 0.1	28.19 ± 0.2 / 43.51 ± 0.1
	LoRA	4.21 ± 0.8 / 6.45 ± 1.2	12.06 ± 0.9 / 17.34 ± 0.8	11.99 ± 0.9 / 17.22 ± 0.9	12.16 ± 0.9 / 17.51 ± 0.8
Qwen3-14B	Zero-shot	10.79 ± 0.2 / 15.75 ± 0.1	**34.43** ± 0.5 / **47.32** ± 0.8	**35.02** ± 0.4 / **47.87** ± 0.8	**34.07** ± 0.5 / **47.07** ± 0.8
	LoRA	7.13 ± 0.4 / 10.12 ± 0.4	17.42 ± 0.2 / 22.53 ± 0.8	17.34 ± 0.2 / 22.37 ± 0.8	17.53 ± 0.2 / 22.75 ± 0.8
Gemini-3.6-Flash	Zero-shot	**12.08** ± 0.8 / **17.45** ± 1.1	32.50 ± 1.8 / 46.25 ± 2.2	33.09 ± 1.9 / 46.92 ± 2.3	32.15 ± 1.6 / 45.91 ± 2.0
GPT-4o-mini	Zero-shot	5.36 ± 0.4 / 10.12 ± 0.7	17.94 ± 0.2 / 30.23 ± 0.6	18.73 ± 0.2 / 31.47 ± 0.6	17.39 ± 0.2 / 29.33 ± 0.6
GPT-5.6-terra	Zero-shot	9.78 ± 0.4 / 14.80 ± 0.5	28.54 ± 0.9 / 40.87 ± 0.5	28.98 ± 0.9 / 41.41 ± 0.5	28.67 ± 0.8 / 41.09 ± 0.5

Table 21: **Evaluation results for each subgoal in Tool Tailor under the *All Preferences & Tools* configuration.** All LoRA models use rank-α=(16-32). For each metric, we report results under two matching criteria: strict matching against the reference tool call (left) and dependency-aware matching against any valid tool call permitted by the task structure (right). **EM**: exact match. **Prec**: precision. **Rec**: recall. All scores are in %. Best performance for each evaluation setting is in **bold**, second best is underlined.

Model	Method	EM	F1	Prec	Rec
		Transport			
Gemma3-4B	Zero-shot	4.94 ± 0.3 / 5.25 ± 0.3	6.28 ± 0.1 / 9.45 ± 0.1	6.28 ± 0.1 / 9.97 ± 0.1	6.28 ± 0.1 / 9.14 ± 0.1
	LoRA	15.90 ± 1.3 / 20.22 ± 1.9	44.59 ± 0.7 / 58.05 ± 0.6	44.57 ± 0.7 / 57.98 ± 0.5	44.62 ± 0.8 / 58.13 ± 0.6
Qwen3-4B	Zero-shot	16.36 ± 0.3 / 16.67 ± 0.5	22.25 ± 0.2 / 27.96 ± 0.5	22.25 ± 0.2 / 28.17 ± 0.5	22.25 ± 0.2 / 27.82 ± 0.5
	LoRA	**20.68** ± 0.5 / **24.07** ± 0.5	**45.58** ± 3.5 / **60.07** ± 2.2	**45.59** ± 3.5 / **60.01** ± 2.2	**45.58** ± 3.4 / **60.14** ± 2.2
Qwen3-8B	Zero-shot	16.05 ± 0.7 / 16.82 ± 0.5	19.42 ± 0.7 / 22.46 ± 0.6	19.42 ± 0.7 / 22.55 ± 0.6	19.42 ± 0.7 / 22.38 ± 0.6
	LoRA	15.74 ± 0.5 / 17.75 ± 1.2	40.18 ± 2.0 / 55.85 ± 1.4	40.18 ± 2.0 / 55.77 ± 1.4	40.19 ± 2.0 / 55.94 ± 1.4
Qwen3-14B	Zero-shot	13.12 ± 1.2 / 13.89 ± 0.9	18.33 ± 1.6 / 22.35 ± 0.9	18.82 ± 1.6 / 23.46 ± 0.9	18.05 ± 1.6 / 21.84 ± 0.9
	LoRA	17.59 ± 0.5 / 19.75 ± 0.7	43.70 ± 0.3 / 56.90 ± 1.7	43.68 ± 0.3 / 56.81 ± 1.7	43.72 ± 0.3 / 57.00 ± 1.7
Gemini-3.6-Flash	Zero-shot	7.56 ± 1.0 / 8.02 ± 0.5	15.08 ± 1.2 / 25.86 ± 0.8	16.37 ± 1.2 / 30.22 ± 1.0	14.18 ± 1.1 / 23.36 ± 0.7
GPT-4o-mini	Zero-shot	8.80 ± 0.5 / 9.72 ± 0.8	10.08 ± 0.1 / 13.23 ± 0.3	10.16 ± 0.2 / 13.81 ± 0.2	10.02 ± 0.1 / 12.87 ± 0.2
GPT-5.6-terra	Zero-shot	10.96 ± 0.7 / 11.73 ± 0.5	19.73 ± 0.8 / 27.02 ± 1.1	21.36 ± 1.0 / 29.20 ± 1.3	19.09 ± 0.8 / 26.06 ± 1.1
		Accommodation			
Gemma3-4B	Zero-shot	5.47 ± 0.3 / 5.47 ± 0.3	7.19 ± 0.3 / 7.41 ± 0.4	7.24 ± 0.3 / 7.54 ± 0.3	7.15 ± 0.3 / 7.35 ± 0.4
	LoRA	17.99 ± 1.1 / 20.11 ± 1.6	47.22 ± 0.4 / 53.78 ± 0.4	47.18 ± 0.5 / 53.72 ± 0.4	47.28 ± 0.4 / 53.86 ± 0.4
Qwen3-4B	Zero-shot	13.76 ± 0.0 / 14.81 ± 0.0	20.89 ± 0.4 / 22.88 ± 0.5	20.89 ± 0.5 / 22.97 ± 0.5	20.90 ± 0.4 / 22.85 ± 0.4
	LoRA	20.81 ± 1.7 / 21.87 ± 1.7	47.21 ± 1.1 / 56.12 ± 0.9	47.21 ± 1.1 / 56.08 ± 0.9	47.21 ± 1.1 / 56.18 ± 0.9
Qwen3-8B	Zero-shot	12.70 ± 0.0 / 13.76 ± 0.0	15.78 ± 0.3 / 17.17 ± 0.2	15.80 ± 0.3 / 17.27 ± 0.3	15.77 ± 0.3 / 17.09 ± 0.2
	LoRA	16.75 ± 1.1 / 17.46 ± 1.4	41.56 ± 1.7 / 49.46 ± 2.5	41.56 ± 1.7 / 49.35 ± 2.5	41.56 ± 1.7 / 49.60 ± 2.5
Qwen3-14B	Zero-shot	12.70 ± 0.5 / 13.93 ± 0.6	23.85 ± 0.3 / 24.72 ± 0.4	24.29 ± 0.2 / 25.18 ± 0.4	23.50 ± 0.3 / 24.38 ± 0.4
	LoRA	**22.75** ± 1.1 / **25.57** ± 0.8	**48.61** ± 0.3 / **56.41** ± 0.7	**48.66** ± 0.3 / **56.41** ± 0.6	**48.60** ± 0.3 / **56.45** ± 0.7
Gemini-3.6-Flash	Zero-shot	10.23 ± 1.1 / 11.11 ± 0.9	18.10 ± 2.1 / 21.73 ± 2.2	18.86 ± 2.2 / 24.18 ± 2.4	17.51 ± 1.9 / 20.40 ± 2.0
GPT-4o-mini	Zero-shot	6.70 ± 0.8 / 7.23 ± 0.8	7.74 ± 0.7 / 8.23 ± 1.1	7.83 ± 0.8 / 8.47 ± 1.4	7.67 ± 0.7 / 8.11 ± 1.0
GPT-5.6-terra	Zero-shot	11.29 ± 1.3 / 11.82 ± 1.3	18.37 ± 0.7 / 20.60 ± 1.1	20.29 ± 1.2 / 22.66 ± 1.6	17.83 ± 0.6 / 19.96 ± 0.9
		Activities			
Gemma3-4B	Zero-shot	5.95 ± 0.6 / 5.95 ± 0.6	6.60 ± 0.6 / 10.85 ± 0.2	6.75 ± 0.6 / 11.66 ± 0.2	6.53 ± 0.6 / 10.56 ± 0.2
	LoRA	19.08 ± 1.1 / 20.41 ± 1.4	25.98 ± 1.9 / 46.82 ± 2.3	25.92 ± 1.9 / 46.81 ± 2.3	26.09 ± 2.0 / 46.90 ± 2.2
Qwen3-4B	Zero-shot	28.10 ± 0.6 / 28.62 ± 0.6	29.59 ± 0.9 / 43.07 ± 1.2	29.60 ± 0.9 / 43.19 ± 1.2	29.57 ± 0.9 / 43.00 ± 1.2
	LoRA	23.28 ± 1.5 / 25.85 ± 1.6	30.72 ± 1.8 / **53.32** ± 1.4	30.66 ± 1.8 / **53.16** ± 1.4	30.84 ± 1.7 / **53.52** ± 1.4
Qwen3-8B	Zero-shot	**35.28** ± 0.6 / **35.90** ± 0.6	**38.51** ± 0.7 / 45.04 ± 0.3	**38.55** ± 0.6 / 45.03 ± 0.3	**38.49** ± 0.7 / 45.09 ± 0.4
	LoRA	20.82 ± 2.6 / 21.23 ± 2.7	26.86 ± 2.1 / 49.11 ± 1.7	26.84 ± 2.1 / 48.97 ± 1.6	26.90 ± 2.1 / 49.27 ± 1.6
Qwen3-14B	Zero-shot	25.13 ± 0.4 / 26.15 ± 0.5	27.47 ± 0.5 / 36.04 ± 0.7	27.84 ± 0.5 / 36.93 ± 0.7	27.41 ± 0.5 / 35.65 ± 0.7
	LoRA	24.82 ± 1.0 / 26.97 ± 1.4	33.57 ± 1.0 / 53.26 ± 0.7	33.50 ± 1.0 / 53.10 ± 0.7	33.70 ± 1.0 / 53.46 ± 0.7
Gemini-3.6-Flash	Zero-shot	18.77 ± 1.2 / 20.10 ± 1.9	21.75 ± 1.9 / 37.42 ± 2.7	22.63 ± 2.0 / 41.20 ± 3.0	21.36 ± 1.8 / 35.62 ± 2.5
GPT-4o-mini	Zero-shot	10.97 ± 0.5 / 10.97 ± 0.5	11.65 ± 0.5 / 17.86 ± 0.8	11.88 ± 0.6 / 19.11 ± 1.0	11.55 ± 0.4 / 17.16 ± 0.8
GPT-5.6-terra	Zero-shot	15.90 ± 1.2 / 17.13 ± 0.9	18.53 ± 1.1 / 30.20 ± 1.0	18.61 ± 1.2 / 30.89 ± 0.9	18.50 ± 1.2 / 29.87 ± 1.0
		Restaurants			
Gemma3-4B	Zero-shot	2.30 ± 0.2 / 2.30 ± 0.2	7.14 ± 0.2 / 7.21 ± 0.2	7.16 ± 0.2 / 7.23 ± 0.3	7.13 ± 0.2 / 7.20 ± 0.2
	LoRA	8.99 ± 0.5 / 10.45 ± 0.2	37.20 ± 1.2 / 37.75 ± 1.3	37.20 ± 1.2 / 37.75 ± 1.2	37.21 ± 1.2 / 37.76 ± 1.3
Qwen3-4B	Zero-shot	10.14 ± 0.5 / 10.87 ± 0.4	30.11 ± 0.1 / 30.52 ± 0.1	30.09 ± 0.1 / 30.51 ± 0.1	30.16 ± 0.1 / 30.57 ± 0.1
	LoRA	**10.24** ± 0.4 / **11.29** ± 0.3	**43.92** ± 0.5 / **44.38** ± 0.4	**43.91** ± 0.5 / **44.37** ± 0.4	**43.94** ± 0.4 / **44.40** ± 0.4
Qwen3-8B	Zero-shot	6.27 ± 0.3 / 6.79 ± 0.2	11.20 ± 0.4 / 11.44 ± 0.4	11.26 ± 0.4 / 11.50 ± 0.4	11.15 ± 0.4 / 11.39 ± 0.4
	LoRA	9.72 ± 0.8 / 10.66 ± 0.8	40.30 ± 0.8 / 40.87 ± 0.9	40.28 ± 0.8 / 40.85 ± 0.9	40.34 ± 0.8 / 40.91 ± 0.9
Qwen3-14B	Zero-shot	7.00 ± 0.2 / 7.63 ± 0.2	17.64 ± 0.2 / 17.91 ± 0.2	18.17 ± 0.3 / 18.44 ± 0.4	17.29 ± 0.1 / 17.55 ± 0.1
	LoRA	10.14 ± 0.7 / 10.87 ± 0.7	39.11 ± 1.0 / 39.63 ± 1.1	39.10 ± 1.0 / 39.62 ± 1.1	39.13 ± 1.0 / 39.65 ± 1.1
Gemini-3.6-Flash	Zero-shot	7.11 ± 0.7 / 7.11 ± 0.7	24.18 ± 3.1 / 24.50 ± 3.0	25.88 ± 3.4 / 26.23 ± 3.3	23.04 ± 2.9 / 23.34 ± 2.9
GPT-4o-mini	Zero-shot	4.81 ± 0.2 / 5.22 ± 0.2	10.75 ± 0.7 / 10.85 ± 0.7	11.41 ± 0.9 / 11.51 ± 0.9	10.28 ± 0.6 / 10.38 ± 0.6
GPT-5.6-terra	Zero-shot	6.48 ± 0.2 / 6.69 ± 0.2	20.03 ± 1.5 / 20.14 ± 1.5	21.13 ± 1.7 / 21.25 ± 1.8	19.49 ± 1.4 / 19.59 ± 1.4
		Booking			
Gemma3-4B	Zero-shot	2.05 ± 0.2 / 3.28 ± 0.5	4.75 ± 0.2 / 9.49 ± 0.3	4.82 ± 0.2 / 9.44 ± 0.3	4.69 ± 0.2 / 9.60 ± 0.3
	LoRA	1.74 ± 0.2 / 2.77 ± 0.5	4.09 ± 0.2 / 7.35 ± 0.4	4.04 ± 0.1 / 7.24 ± 0.4	4.18 ± 0.3 / 7.50 ± 0.5
Qwen3-4B	Zero-shot	11.90 ± 0.4 / 18.46 ± 0.5	35.10 ± 0.5 / **54.18** ± 0.7	35.67 ± 0.5 / 54.83 ± 0.6	34.78 ± 0.6 / **53.88** ± 0.8
	LoRA	4.62 ± 1.1 / 7.08 ± 0.8	6.73 ± 0.5 / 10.88 ± 0.5	6.71 ± 0.5 / 10.79 ± 0.5	6.76 ± 0.5 / 10.99 ± 0.6
Qwen3-8B	Zero-shot	8.92 ± 0.0 / 14.67 ± 0.5	26.45 ± 0.3 / 48.76 ± 0.1	26.60 ± 0.3 / 48.82 ± 0.1	26.46 ± 0.3 / 48.98 ± 0.2
	LoRA	4.21 ± 1.0 / 7.18 ± 1.2	8.18 ± 1.1 / 15.00 ± 0.6	8.09 ± 1.1 / 14.84 ± 0.7	8.31 ± 1.1 / 15.25 ± 0.6
Qwen3-14B	Zero-shot	**12.82** ± 0.4 / **20.31** ± 0.0	**35.64** ± 0.5 / 54.06 ± 1.0	**36.48** ± 0.5 / **54.84** ± 1.0	**35.12** ± 0.6 / 53.70 ± 1.1
	LoRA	7.38 ± 0.5 / 11.38 ± 0.5	12.14 ± 0.4 / 18.93 ± 1.1	12.02 ± 0.4 / 18.69 ± 1.1	12.31 ± 0.4 / 19.26 ± 1.1
Gemini-3.6-Flash	Zero-shot	12.10 ± 0.6 / 19.59 ± 1.2	33.28 ± 2.0 / 52.83 ± 2.6	33.86 ± 2.2 / 53.45 ± 2.8	32.99 ± 1.9 / 52.61 ± 2.4
GPT-4o-mini	Zero-shot	5.85 ± 0.3 / 13.03 ± 0.8	18.76 ± 0.2 / 36.78 ± 0.7	19.89 ± 0.2 / 38.56 ± 0.7	17.95 ± 0.2 / 35.47 ± 0.7
GPT-5.6-terra	Zero-shot	10.15 ± 0.3 / 17.44 ± 0.5	28.52 ± 0.9 / 46.28 ± 0.4	29.12 ± 0.9 / 47.03 ± 0.4	28.75 ± 0.9 / 46.66 ± 0.4
		Calendar			
Gemma3-4B	Zero-shot	1.20 ± 0.0 / 1.20 ± 0.0	8.94 ± 0.4 / 9.64 ± 0.3	8.96 ± 0.4 / 9.65 ± 0.3	8.93 ± 0.4 / 9.63 ± 0.3
	LoRA	1.20 ± 0.0 / 2.21 ± 0.7	15.00 ± 0.6 / 16.85 ± 1.3	15.00 ± 0.6 / 16.84 ± 1.3	15.01 ± 0.6 / 16.85 ± 1.3
Qwen3-4B	Zero-shot	7.03 ± 0.3 / 7.03 ± 0.3	24.84 ± 0.7 / 25.29 ± 0.7	24.93 ± 0.7 / 25.39 ± 0.7	24.77 ± 0.7 / 25.20 ± 0.7
	LoRA	6.63 ± 0.6 / 7.63 ± 0.3	26.21 ± 0.4 / 28.01 ± 1.1	26.21 ± 0.4 / 28.01 ± 1.1	26.21 ± 0.4 / 28.01 ± 1.1
Qwen3-8B	Zero-shot	7.23 ± 0.0 / 7.23 ± 0.0	31.53 ± 0.0 / 32.82 ± 0.0	31.53 ± 0.1 / 32.84 ± 0.0	31.55 ± 0.1 / 32.82 ± 0.0
	LoRA	4.22 ± 0.6 / 5.02 ± 1.3	19.66 ± 0.5 / 21.91 ± 1.3	19.64 ± 0.6 / 21.88 ± 1.4	19.68 ± 0.5 / 21.95 ± 1.3
Qwen3-14B	Zero-shot	6.83 ± 0.3 / 6.83 ± 0.3	**32.08** ± 0.4 / **34.13** ± 0.4	**32.16** ± 0.4 / **34.22** ± 0.4	**32.04** ± 0.3 / **34.07** ± 0.4
	LoRA	6.63 ± 0.0 / 7.63 ± 0.3	27.76 ± 0.4 / 29.58 ± 0.2	27.76 ± 0.4 / 29.58 ± 0.2	27.76 ± 0.4 / 29.58 ± 0.2
Gemini-3.6-Flash	Zero-shot	**12.05** ± 1.2 / **13.25** ± 1.2	30.96 ± 1.2 / 33.36 ± 1.4	31.58 ± 1.2 / 34.14 ± 1.5	30.52 ± 1.2 / 32.81 ± 1.4
GPT-4o-mini	Zero-shot	4.42 ± 0.7 / 4.42 ± 0.7	16.35 ± 0.7 / 17.40 ± 0.5	16.44 ± 0.8 / 17.57 ± 0.5	16.29 ± 0.7 / 17.31 ± 0.4
GPT-5.6-terra	Zero-shot	9.04 ± 1.0 / 9.64 ± 1.0	28.59 ± 0.9 / 30.27 ± 0.6	28.70 ± 0.9 / 30.41 ± 0.6	28.52 ± 0.8 / 30.17 ± 0.6

Table 22: **Personalised F1 grouped by the number of relevant user preferences under the *All Preferences & Tools* evaluation configuration.** All LoRA models use rank-α=(16-32). For each metric, we report results under two matching criteria: strict matching against the reference tool call (left) and dependency-aware matching against any valid tool call permitted by the task structure (right). **Prec**: precision. **Rec**: recall. All scores are in %. Best performance for each method is in **bold**, second best is underlined.

Model	Method	Bucket	EM	F1	Prec	Rec
Gemma3-4B	Zero-shot	1-2	**9.45** ± 0.4 / **10.29** ± 0.5	**10.34** ± 0.4 / **12.21** ± 0.3	**10.39** ± 0.4 / **12.60** ± 0.3	**10.31** ± 0.4 / **12.07** ± 0.4
		3	4.90 ± 0.2 / 6.40 ± 0.2	6.03 ± 0.3 / 8.18 ± 0.2	6.03 ± 0.3 / 8.32 ± 0.3	6.02 ± 0.2 / 8.12 ± 0.2
		4	3.94 ± 0.4 / 5.96 ± 0.2	5.27 ± 0.4 / 8.18 ± 0.6	5.48 ± 0.5 / 8.44 ± 0.7	5.20 ± 0.4 / 8.07 ± 0.6
	LoRA	1-2	22.65 ± 1.1 / 27.18 ± 1.2	30.87 ± 1.1 / 39.41 ± 1.1	30.86 ± 1.1 / 39.40 ± 1.1	30.94 ± 1.1 / 39.47 ± 1.1
		3	25.02 ± 1.2 / 29.21 ± 0.9	35.31 ± 1.3 / 44.18 ± 0.7	35.29 ± 1.3 / 44.16 ± 0.7	35.36 ± 1.3 / 44.23 ± 0.7
		4	**29.71** ± 0.6 / **35.78** ± 2.1	**37.76** ± 1.8 / **49.31** ± 0.6	**37.77** ± 1.8 / **49.31** ± 0.6	**37.75** ± 1.8 / **49.30** ± 0.6
Qwen3-4B	Zero-shot	1-2	**31.79** ± 0.3 / **36.37** ± 0.4	**35.33** ± 0.5 / **42.43** ± 0.8	**35.35** ± 0.5 / **42.71** ± 0.8	**35.32** ± 0.5 / **42.31** ± 0.7
		3	25.59 ± 0.6 / 32.13 ± 0.8	30.29 ± 0.8 / 39.40 ± 1.1	30.32 ± 0.8 / 39.70 ± 1.1	30.27 ± 0.8 / 39.30 ± 1.1
		4	26.20 ± 0.8 / 32.06 ± 0.7	31.45 ± 0.6 / 41.24 ± 0.8	31.54 ± 0.6 / 41.49 ± 0.8	31.39 ± 0.6 / 41.07 ± 0.8
	LoRA	1-2	31.22 ± 1.3 / 36.24 ± 1.4	39.71 ± 0.7 / 48.93 ± 0.5	39.68 ± 0.7 / 48.90 ± 0.5	39.84 ± 0.6 / 49.06 ± 0.4
		3	30.49 ± 1.5 / 36.11 ± 1.2	39.74 ± 1.2 / 50.40 ± 0.3	39.76 ± 1.2 / 50.42 ± 0.3	39.72 ± 1.2 / 50.47 ± 0.2
		4	**34.08** ± 0.7 / **43.34** ± 0.9	**40.83** ± 1.1 / **54.63** ± 0.5	**40.83** ± 1.1 / **54.63** ± 0.5	**40.83** ± 1.1 / **54.63** ± 0.5
Qwen3-8B	Zero-shot	1-2	**32.85** ± 0.2 / **37.60** ± 0.5	**33.71** ± 0.2 / **39.40** ± 0.2	**33.74** ± 0.2 / **39.47** ± 0.1	**33.69** ± 0.2 / **39.34** ± 0.2
		3	22.39 ± 0.2 / 28.86 ± 0.2	23.46 ± 0.1 / 30.82 ± 0.2	23.46 ± 0.1 / 30.83 ± 0.2	23.46 ± 0.1 / 30.81 ± 0.2
		4	26.52 ± 0.8 / 34.61 ± 0.7	28.72 ± 0.6 / 38.14 ± 0.4	28.88 ± 0.6 / 38.40 ± 0.3	28.63 ± 0.6 / 38.00 ± 0.5
	LoRA	1-2	25.68 ± 0.7 / 32.28 ± 0.5	33.29 ± 0.5 / 44.02 ± 0.2	33.26 ± 0.5 / 43.99 ± 0.2	33.38 ± 0.5 / 44.11 ± 0.2
		3	27.08 ± 1.7 / 33.76 ± 2.3	36.21 ± 1.7 / 48.37 ± 2.3	36.23 ± 1.7 / 48.38 ± 2.2	36.20 ± 1.8 / 48.41 ± 2.3
		4	**30.03** ± 2.6 / **38.87** ± 3.4	**38.33** ± 3.0 / **52.61** ± 3.3	**38.33** ± 3.0 / **52.61** ± 3.3	**38.33** ± 3.0 / **52.61** ± 3.3
Qwen3-14B	Zero-shot	1-2	**33.42** ± 0.3 / **37.03** ± 0.5	**36.08** ± 0.4 / **41.07** ± 0.6	**36.53** ± 0.5 / **41.71** ± 0.7	**35.89** ± 0.3 / **40.80** ± 0.5
		3	22.32 ± 0.3 / 28.43 ± 0.5	25.68 ± 0.3 / 32.98 ± 0.6	26.05 ± 0.3 / 33.38 ± 0.5	25.48 ± 0.3 / 32.76 ± 0.6
		4	23.86 ± 1.0 / 30.56 ± 0.7	27.78 ± 1.1 / 36.58 ± 0.5	28.10 ± 1.0 / 37.11 ± 0.3	27.62 ± 1.1 / 36.32 ± 0.5
	LoRA	1-2	32.23 ± 0.3 / 37.73 ± 0.5	40.94 ± 1.0 / 49.12 ± 1.0	40.91 ± 1.0 / 49.08 ± 1.0	41.10 ± 1.0 / 49.28 ± 1.0
		3	30.77 ± 0.5 / 37.53 ± 1.1	38.56 ± 1.0 / 49.59 ± 0.9	38.53 ± 1.0 / 49.56 ± 0.9	38.61 ± 1.0 / 49.63 ± 0.9
		4	**34.61** ± 1.1 / **42.39** ± 1.3	**41.15** ± 0.8 / **54.09** ± 0.5	**41.15** ± 0.8 / **54.09** ± 0.5	**41.15** ± 0.8 / **54.09** ± 0.5
Gemini-3.6-Flash	Zero-shot	1-2	**29.60** ± 2.5 / **33.91** ± 2.5	**33.13** ± 2.5 / **39.62** ± 2.7	**34.48** ± 2.4 / **42.06** ± 2.8	**32.62** ± 2.6 / **38.74** ± 2.7
		3	18.12 ± 1.0 / 25.30 ± 1.7	23.61 ± 1.9 / 33.96 ± 2.4	24.95 ± 2.2 / 36.74 ± 2.6	22.97 ± 1.7 / 32.85 ± 2.3
		4	21.19 ± 1.3 / 28.01 ± 2.1	26.26 ± 2.0 / 36.74 ± 3.2	27.59 ± 2.5 / 39.58 ± 4.1	25.65 ± 1.8 / 35.57 ± 2.9
GPT-4o-mini	Zero-shot	1-2	**16.31** ± 0.6 / **20.18** ± 0.7	**17.27** ± 0.6 / **22.43** ± 0.8	**17.58** ± 0.6 / **23.14** ± 0.8	**17.13** ± 0.6 / **22.10** ± 0.8
		3	10.31 ± 0.1 / 15.64 ± 0.5	11.39 ± 0.0 / 17.94 ± 0.3	11.70 ± 0.1 / 18.60 ± 0.3	11.21 ± 0.0 / 17.59 ± 0.4
		4	13.10 ± 0.3 / 18.21 ± 0.3	14.75 ± 0.5 / 20.65 ± 0.6	14.97 ± 0.5 / 21.20 ± 0.9	14.61 ± 0.4 / 20.39 ± 0.5
GPT-5.6-terra	Zero-shot	1-2	**27.62** ± 0.5 / **31.57** ± 0.6	**30.49** ± 0.8 / **36.89** ± 0.6	**31.17** ± 0.8 / **37.83** ± 0.6	**30.25** ± 0.8 / **36.52** ± 0.6
		3	18.27 ± 1.3 / 25.09 ± 1.9	23.42 ± 0.8 / 33.02 ± 1.0	24.16 ± 0.9 / 33.91 ± 0.9	23.07 ± 0.8 / 32.57 ± 1.0
		4	22.04 ± 0.6 / 27.37 ± 0.2	27.31 ± 0.9 / 36.16 ± 0.4	27.94 ± 0.9 / 37.28 ± 0.3	26.97 ± 0.9 / 35.59 ± 0.4

Table 23: **Personalised F1 grouped by the number of relevant user preferences under the *Relevant Preferences & Tools* evaluation configuration.** All LoRA models use rank-α=(16-32). For each metric, we report results under two matching criteria: strict matching against the reference tool call (left) and dependency-aware matching against any valid tool call permitted by the task structure (right). **Prec**: precision. **Rec**: recall. All scores are in %. Best performance for each method is in **bold**, second best is underlined.

Model	Method	Bucket	EM	F1	Prec	Rec
Gemma3-4B	Zero-shot	1-2	**19.92** ± 0.1 / **22.25** ± 0.3	**22.69** ± 0.1 / **25.32** ± 0.3	**22.87** ± 0.2 / **25.51** ± 0.3	**22.59** ± 0.1 / **25.21** ± 0.3
		3	8.96 ± 0.2 / 12.37 ± 0.0	15.35 ± 0.1 / 19.20 ± 0.2	15.67 ± 0.1 / 19.58 ± 0.2	15.22 ± 0.1 / 19.05 ± 0.2
		4	10.01 ± 0.2 / 13.63 ± 0.2	18.44 ± 0.5 / 22.29 ± 0.7	19.38 ± 0.5 / 23.24 ± 0.8	18.03 ± 0.5 / 21.89 ± 0.7
	LoRA	1-2	38.48 ± 0.2 / 42.79 ± 0.1	56.64 ± 0.3 / 61.70 ± 0.3	56.62 ± 0.3 / 61.81 ± 0.3	56.77 ± 0.3 / 61.81 ± 0.4
		3	35.61 ± 1.5 / 40.65 ± 0.8	54.83 ± 1.6 / 60.35 ± 1.2	54.81 ± 1.6 / 60.33 ± 1.2	54.89 ± 1.6 / 60.41 ± 1.2
		4	**44.09** ± 1.9 / **49.31** ± 1.6	**59.32** ± 1.1 / **64.39** ± 0.7	**59.33** ± 1.1 / **64.39** ± 0.8	**59.36** ± 1.1 / **64.42** ± 0.7
Qwen3-4B	Zero-shot	1-2	**37.25** ± 0.1 / **41.73** ± 0.1	43.16 ± 0.1 / 48.21 ± 0.3	43.48 ± 0.2 / 48.53 ± 0.3	43.04 ± 0.1 / 48.09 ± 0.3
		3	27.22 ± 0.7 / 33.48 ± 0.4	36.58 ± 0.6 / 43.96 ± 0.3	37.24 ± 0.5 / 44.63 ± 0.4	36.20 ± 0.6 / 43.58 ± 0.4
		4	34.19 ± 1.2 / 39.83 ± 1.0	**44.89** ± 1.4 / **51.67** ± 1.2	**45.47** ± 1.4 / **52.25** ± 1.2	**44.53** ± 1.4/ **51.30** ± 1.2
	LoRA	1-2	44.85 ± 0.7 / 49.34 ± 0.3	63.10 ± 0.7 / 68.34 ± 0.4	63.06 ± 0.7 / 68.30 ± 0.4	63.27 ± 0.7 / 68.50 ± 0.4
		3	44.56 ± 1.1 / 49.68 ± 1.5	62.60 ± 0.6 / 68.24 ± 1.2	62.57 ± 0.6 / 68.21 ± 1.2	62.67 ± 0.5 / 68.31 ± 1.1
		4	**52.08** ± 2.0 / **58.25** ± 1.8	**65.00** ± 1.5 / **71.12** ± 1.2	**64.97** ± 1.5 / **71.10** ± 1.2	**65.06** ± 1.5 / **71.19** ± 1.2
Qwen3-8B	Zero-shot	1-2	**40.63** ± 0.1 / **45.56** ± 0.3	**45.31** ± 0.4 / **50.76** ± 0.5	**45.78** ± 0.5 / **51.24** ± 0.5	**45.11** ± 0.3 / **50.56** ± 0.4
		3	25.09 ± 0.3 / 32.41 ± 0.6	33.54 ± 0.5 / 41.32 ± 0.4	34.40 ± 0.6 / 42.19 ± 0.5	33.08 ± 0.5 / 40.84 ± 0.3
		4	33.33 ± 0.7 / 40.26 ± 0.3	43.01 ± 0.5 / 50.19 ± 0.5	43.46 ± 0.5 / 50.63 ± 0.6	42.75 ± 0.4 / 49.93 ± 0.4
	LoRA	1-2	41.25 ± 0.6 / 46.39 ± 0.5	59.59 ± 0.2 / 65.59 ± 0.4	59.55 ± 0.2 / 65.54 ± 0.4	59.76 ± 0.2 / 65.77 ± 0.4
		3	37.74 ± 1.5 / 44.78 ± 1.3	56.97 ± 1.1 / 64.85 ± 1.0	56.96 ± 1.1 / 64.85 ± 1.0	57.05 ± 1.1 / 64.91 ± 0.9
		4	**49.41** ± 2.8 / **55.27** ± 3.5	**63.60** ± 1.0 / **69.97** ± 2.1	**63.64** ± 1.0 / **70.01** ± 2.1	**63.62** ± 1.0 / **69.98** ± 2.1
Qwen3-14B	Zero-shot	1-2	**39.75** ± 0.3 / **43.84** ± 0.4	**46.32** ± 0.3 / 50.98 ± 0.5	**47.74** ± 0.4 / 52.40 ± 0.6	**45.71** ± 0.3 / 50.37 ± 0.5
		3	28.22 ± 0.1 / 36.03 ± 0.0	37.72 ± 0.8 / 46.42 ± 0.7	38.94 ± 0.9 / 47.76 ± 0.8	37.13 ± 0.7 / 45.79 ± 0.6
		4	32.37 ± 0.2 / 39.08 ± 0.5	44.33 ± 0.8 / **52.43** ± 0.6	45.58 ± 0.9 / **53.70** ± 0.7	43.66 ± 0.7 / **51.76** ± 0.6
	LoRA	1-2	42.48 ± 1.1 / 47.49 ± 0.6	59.80 ± 1.2 / 65.61 ± 0.7	59.75 ± 1.2 / 65.58 ± 0.7	59.96 ± 1.2 / 65.78 ± 0.7
		3	41.86 ± 1.9 / 47.55 ± 1.9	60.10 ± 1.4 / 66.22 ± 1.3	60.06 ± 1.4 / 66.17 ± 1.3	60.20 ± 1.5 / 66.31 ± 1.3
		4	**51.22** ± 1.3 / **56.98** ± 1.9	**65.18** ± 0.5 / **71.06** ± 0.8	**65.22** ± 0.5 / **71.11** ± 0.8	**65.20** ± 0.5 / **71.09** ± 0.8
Gemini-3.6-Flash	Zero-shot	1-2	**36.81** ± 0.5 / **40.85** ± 0.6	**44.01** ± 0.6 / **48.65** ± 0.6	**47.61** ± 0.5 / **52.54** ± 0.4	**42.74** ± 0.7 / **47.30** ± 0.7
		3	20.97 ± 0.7 / 27.01 ± 0.8	31.69 ± 0.9 / 38.97 ± 0.6	35.61 ± 1.3 / 43.32 ± 0.9	29.96 ± 0.9 / 37.09 ± 0.6
		4	26.20 ± 3.3 / 32.27 ± 2.5	37.63 ± 3.6 / 44.94 ± 2.5	40.59 ± 3.1 / 48.10 ± 1.8	36.16 ± 3.6 / 43.37 ± 2.6
GPT-4o-mini	Zero-shot	1-2	**22.91** ± 0.5 / 27.35 ± 0.7	28.13 ± 0.5 / 32.84 ± 0.7	30.26 ± 0.4 / 35.05 ± 0.6	27.26 ± 0.5 / 31.95 ± 0.7
		3	16.92 ± 0.1 / 23.74 ± 0.2	24.33 ± 0.3 / 31.84 ± 0.6	26.01 ± 0.3 / 33.74 ± 0.6	23.49 ± 0.3 / 30.91 ± 0.5
		4	21.83 ± 1.0 / **28.01** ± 1.2	**30.52** ± 1.3 / **37.04** ± 1.5	**31.99** ± 1.4 / **38.55** ± 1.6	**29.71** ± 1.2 / **36.21** ± 1.4
GPT-5.6-terra	Zero-shot	1-2	**32.37** ± 0.8 / **37.03** ± 0.7	**36.91** ± 1.1 / **42.82** ± 1.0	**37.84** ± 1.2 / **43.90** ± 1.0	**36.56** ± 1.1 / **42.41** ± 1.0
		3	21.89 ± 0.5 / 28.29 ± 0.5	29.51 ± 0.2 / 36.98 ± 0.1	30.65 ± 0.2 / 38.30 ± 0.0	28.94 ± 0.3 / 36.31 ± 0.2
		4	25.99 ± 0.5 / 32.80 ± 1.0	34.72 ± 0.8 / 42.50 ± 0.8	35.62 ± 0.8 / 43.40 ± 0.7	34.20 ± 0.8 / 41.97 ± 0.8