DEEPPLANNING-ACE: CAN LANGUAGE AGENTS ACE INTERACTIVE LONG-HORIZON PLANNING?

Anonymous authors
Paper under double-blind review

ABSTRACT

Long-horizon planning benchmarks typically assume that user requirements are complete, unambiguous, and fixed from the outset. Real-world planning, however, is inherently interactive: users disclose constraints gradually, clarify underspecified preferences, revise earlier decisions, and sometimes make requests for which no valid solution exists. We introduce **DeepPlanning-ACE**, named after capabilities essential to interactive planning: resolving **A**mbiguity, adapting to **C**hange, and **E**xplaining infeasibility. This also captures the benchmark's central question: can language agents truly ace deep planning as user intent evolves through interaction? DeepPlanning-ACE comprises bilingual episodes derived from verifiable base instances in travel planning and multi-product shopping. It preserves the original tasks and constructs ambiguous, changing, and infeasible variants when their defining conditions can be verified. To enable controlled and reproducible user simulation, the benchmark decouples semantic intent from linguistic realization. User requirements, intent changes, and infeasibility conditions are specified and verified in advance. A deterministic controller governs information disclosure, state transitions, turn limits, and termination, while a language model can select from authorized disclosures without modifying their canonical content. Evaluation covers both task completion and interaction quality along the ACE dimensions: clarifying ambiguous requirements, adapting to changing constraints, and explicitly explaining infeasibility. Experiments across nine language agents show that multi-turn interactive planning remains challenging for all evaluated models, revealing persistent limitations in maintaining valid plans, adapting to changing requirements, and explaining infeasibility with sufficient tool evidence.

1 INTRODUCTION

Planning a multi-day trip is an iterative process. Users rarely articulate all their requirements in a single request, and their preferences may evolve as they explore available options. A traveler may begin with a destination and budget, clarify transportation preferences after reviewing a proposed itinerary, revise a hotel requirement after comparing alternatives, or request an option that turns out to be unavailable. Similar interactions arise in shopping, where users refine product requirements and reconsider choices as they learn about prices, compatibility, and availability. Each interaction can affect decisions already made and the constraints governing subsequent choices. Effective planning therefore requires an assistant to elicit missing requirements, revise interdependent decisions, and explain limitations while remaining aligned with the user's evolving intent.

Representative planning benchmarks, including TravelPlanner (Xie et al., 2024) and DeepPlanning (Zhang et al., 2026), evaluate travel itineraries and purchasing plans under a single user request. Although agents may perform extensive reasoning and repeatedly invoke tools, the user's requirements remain fixed throughout the task. These benchmarks assess whether an agent can construct a valid plan for a given request, but do not directly test whether it can ask the user to resolve ambiguity or respond to a requirement revised after planning has begun. The distinction between tool interaction and user interaction is consequential: tools reveal facts about the environment, whereas subsequent user turns can change what constitutes a satisfactory solution. A plan that satisfies the original request may therefore cease to meet the user's needs as the conversation progresses.

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.

Recent benchmarks have begun to examine multi-turn interaction, including incremental preference disclosure (Qian et al., 2025), multi-turn and unsolvable travel requests (Cheng et al., 2026), and ambiguous, changing, or infeasible intents in tool-calling workflows (Wang et al., 2026). Building on these directions, we focus on how evolving requirements affect verifiable long-horizon planning with interdependent local and global constraints. Clarifying or revising one requirement can require changes across an itinerary or shopping cart, while an infeasible additional request calls for an evidence-based explanation that preserves the feasible parts of the task. This motivates an evaluation that combines controlled user interactions with checks of both plan validity and the decisions made along the way. Figure 1 illustrates this setting by contrasting a fixed, single-request task with a request that evolves through clarification, revision, and the discovery of unavailable options.

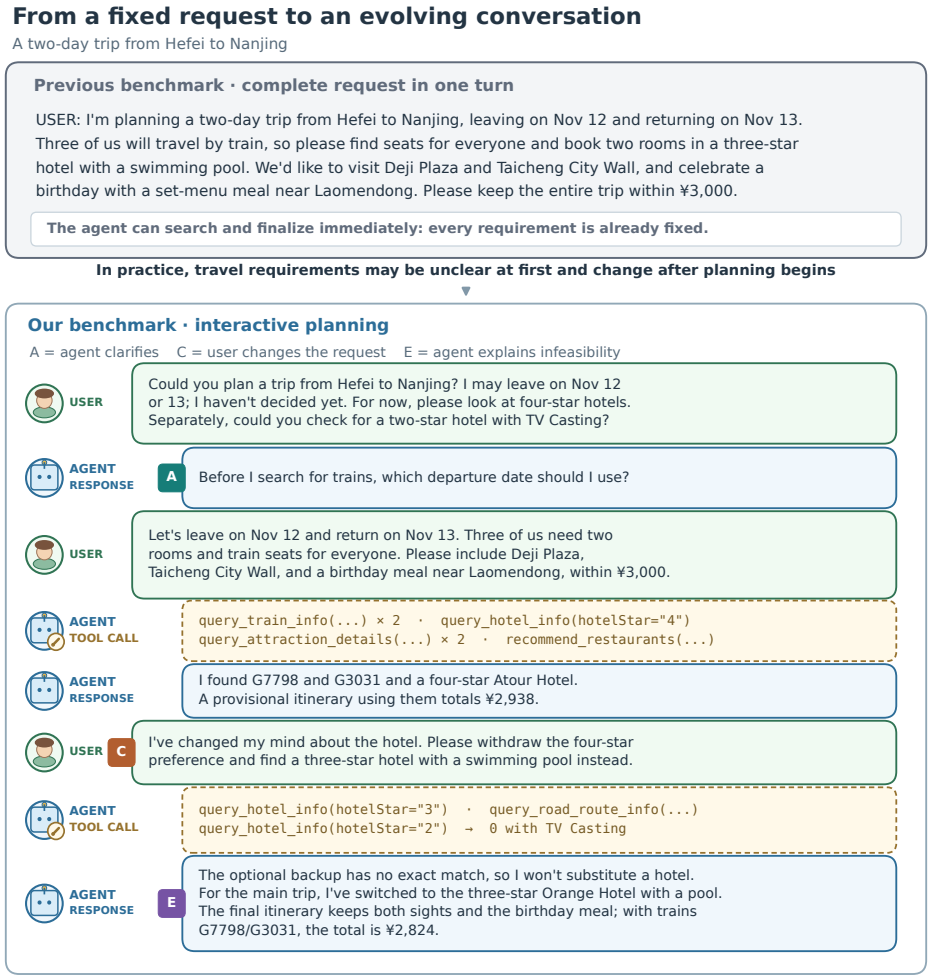

Figure 1: Travel planning with a fully specified request (top) and requirements that evolve through interaction (bottom). **A** denotes clarification of an ambiguous requirement, **C** a revision to the user's hotel requirement, and **E** a tool-grounded explanation that an optional backup is unavailable. The lower dialogue combines the three interaction types for illustration; the benchmark evaluates them in separate episodes.

We introduce **DeepPlanning-ACE**, an interactive planning benchmark organized around three capabilities: resolving **A**mbiguity, adapting to **C**hange, and **E**xplaining infeasibility. ACE extends DeepPlanning's verifiable travel and multi-product shopping environments with controlled, multi-turn user interactions. It draws on 360 base instances: 120 Chinese travel instances, 120 English travel instances, and 120 English shopping instances. We retain each base task under normal disclosure and construct interactive variants where the required alternatives and certificates can be verified. In an ambiguous episode, the agent must clarify an unresolved requirement before committing to

a candidate. In a changing episode, the user revises a provisional requirement after the agent has performed relevant work, requiring it to update the affected downstream decisions. In an infeasible episode, an additional request admits no exact solution in the task environment; the agent must establish this through tool evidence, explain the limitation, and continue to satisfy the feasible base requirements.

Reliable evaluation requires these interactions to be controlled and auditable. An unconstrained simulated user may inadvertently alter the task, reveal solution information, or disclose a critical requirement at inconsistent points across agent runs. DeepPlanning-ACE therefore separates *semantic intent* from its conversational presentation. An offline generator specifies canonical requirements, revisions, and infeasibility certificates. At run time, a deterministic controller governs information availability, event order, turn limits, and termination. During ordinary dialogue turns, the user model may select authorized requirements for disclosure, while the controller supplies the corresponding canonical facts. Scripted revisions and infeasibility events are likewise drawn from verified case data. This design supports conversational interaction while keeping task semantics and evaluation conditions explicit.

We evaluate both the final plan and the interaction process. The original domain checkers assess whether the itinerary or shopping cart satisfies the base task. Additional trajectory-based checks determine whether the agent clarified ambiguity before committing, adapted to a revision after performing relevant work, and gathered appropriate evidence before reporting infeasibility. An interactive episode succeeds only if both the final outcome and the required interaction behavior pass. This joint criterion captures failures that final-answer evaluation alone can overlook, including plans based on superseded requirements and unsupported claims that no valid option exists.

Our contributions are threefold: (i) we extend single-request travel and shopping planning to controlled multi-turn settings, where requirements are clarified or revised during planning and additional requests may prove infeasible; (ii) we develop verified episode generation, deterministic interaction control, and process-aware evaluation for the three ACE capabilities; and (iii) we use the benchmark to analyze how language agents clarify requirements, revise interdependent decisions, and justify infeasibility during interaction.

2 Related Work

PlanBench evaluates classical planning (Valmeekam et al., 2023), while TravelPlanner and DeepPlanning study tool-grounded planning under fixed user requirements (Xie et al., 2024; Zhang et al., 2026). DeepPlanning-ACE retains DeepPlanning's coupled local and global constraints but tests whether agents maintain valid plans as requirements are clarified or revised. TravelBench already includes multi-turn preference elicitation and unsolvable travel tasks (Cheng et al., 2026). Our distinction lies in constructing verified interaction variants from identifiable base instances across both travel and shopping: revisions follow relevant work under provisional requirements, and infeasible backup requests must be explained while the feasible base task remains active. This construction connects interaction failures to an independently checkable planning problem.

τ-bench studies tool–agent–user interaction (Yao et al., 2025), UserBench emphasizes eliciting underspecified preferences (Qian et al., 2025), and ToolSandbox evaluates intermediate and final milestones in stateful tool use (Lu et al., 2025). We build on these interaction and process-evaluation perspectives to test how evolving requirements affect interdependent decisions across a complete itinerary or cart. Trajectory2Task is closest in its intent taxonomy: it converts valid tool-call trajectories into verifiable tasks with ambiguous, changing, and infeasible intents for training and evaluation (Wang et al., 2026). DeepPlanning-ACE shares these categories but starts from fixed planning instances and their domain checkers. It verifies alternatives and infeasibility certificates before evaluation, gates revisions on agent actions, and requires both global plan validity and the corresponding clarification, adaptation, or evidence-grounded explanation. The contribution is this controlled planning evaluation, rather than the A/C/E taxonomy itself.

SimulatorArena examines agreement between simulated and human users (Dou et al., 2025), while goal-alignment research improves simulators' ability to track user goals (Mehri et al., 2026). Our controller addresses the complementary problem of preserving task semantics across agent runs: canonical requirements, event order, and termination are controlled deterministically, and the user

model selects only authorized disclosures. This supports reproducible evaluation of evolving intent without relying on free-form simulation to preserve the specification.

3 DEEPPLANNING-ACE

DeepPlanning-ACE transforms a constrained planning task into a controlled interaction among an agent, a user, and a tool environment. Its design comprises three components: an offline generator that specifies each episode's semantics, a deterministic controller that governs user disclosure, and a verifier that evaluates both the resulting plan and the interaction that produced it.

Figure 2 illustrates the benchmark through parallel travel-planning and shopping-assistant conversations. In each column, an unresolved choice first requires clarification from the agent (**A**). After the agent begins tool-based work, the user revises a requirement (**C**), prompting an update to the plan or cart. Finally, the agent investigates an unavailable backup option and explains its infeasibility using tool evidence (**E**). The model row lists the evaluated agent families, and the evaluation cards distinguish two pass conditions. Base-task success (B) requires either a valid itinerary that satisfies all eight planning dimensions and every case-specific hard constraint, or a cart that exactly matches an accepted reference and is verified after its final edit. Process success (P) requires the behavior specified by the episode's ACE condition. An interactive episode succeeds only when both conditions hold, as formalized in Eq. (3). Each illustrated conversation combines independently generated A, C, and E cases for exposition; the benchmark evaluates these conditions in separate episodes.

3.1 TASK FORMULATION AND EPISODE CONSTRUCTION

Let a base instance be $x = (D, q, R, y^\star)$, where D comprises the task's sandboxed database and tools, q is the original request, $R = \{(r_i, v_i)\}_{i=1}^{n}$ is a set of canonical user requirements, and $y^\star$ denotes the reference information used by the domain evaluator. The agent accesses D exclusively through the provided tools and must produce either a travel plan or a shopping cart. We construct episodes from 360 DeepPlanning base instances: 120 Chinese travel instances, 120 English travel instances, and 120 English shopping instances. Both domains retain their original planning challenges, including coupled itinerary constraints and cart-level product, delivery, coupon, and budget constraints (Zhang et al., 2026).

The generator first extracts user-visible disclosure units from each base instance. Reproducible interactive environments and stateful tool execution are established concerns in agent evaluation (Zhou et al., 2024; Lu et al., 2025); accordingly, our construction separates user-visible facts from evaluator-only answers. Travel disclosure units include dates, party size, room count, and textual descriptions of hard constraints. Shopping units include item-level requests and any explicitly stated total-price interval. Answer fields, such as reference products, hotels, and coupons, are excluded from the user simulator's view. Each derived episode stores a link to its base instance and the corresponding hash, together with an *interaction script*

$$\sigma = (k, m_0, E, C), \qquad E = (e_1, \ldots, e_K), \tag{1}$$

where $k \in \{\text{AMBIGUOUS}, \text{CHANGING}, \text{INFEASIBLE}\}$ specifies the interaction condition, m_0 is the initial perturbation, E is an ordered event list, and C contains private construction certificates and process requirements. Each event $e_j = (g_j, u_j, L_j)$ defines a trigger g_j, a canonical user message u_j, and the requirement identifiers L_j unlocked by that message. The normal condition uses the base instance with gradual requirement disclosure and no scripted perturbation. Generation preserves the source cases and records template counts, source hashes, and skipped cases in a manifest. A derived episode is omitted if its required witness cannot be established.

3.2 CONTROLLED INTENT CONDITIONS

Ambiguity, change, and infeasibility have also been studied in tool-calling task synthesis (Wang et al., 2026). Here, each condition is constructed from a fixed planning instance and verified against its source data before the evaluated dialogue begins.

Ambiguity. The initial request presents two possible values but withholds the canonical choice. Determining when a missing distinction warrants a question is a central challenge in clarification research (Zhang & Choi, 2025; Zhang et al., 2024). In travel, the ambiguous field is the departure date.

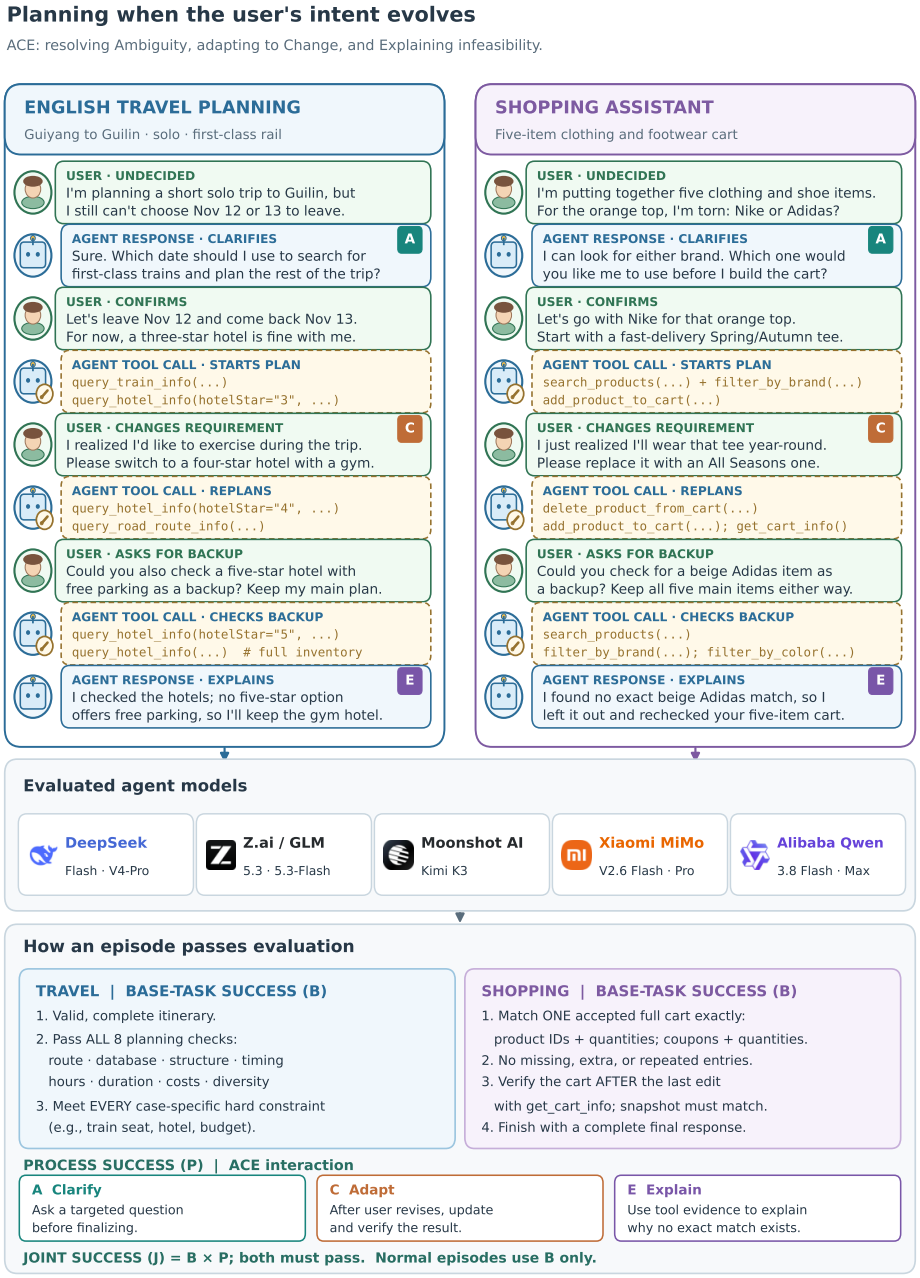

Figure 2: Travel and shopping examples of agent clarification (A), user revision (C), and evidence-grounded infeasibility explanation (E). The lower panels summarize the evaluated model families and the criteria for base-task, process, and joint success. Each column combines separate ACE episodes for illustration.

In shopping, it is an item attribute, such as color or size, or, when no suitable attribute alternative exists, whether the user wants an additional backup item. The corresponding requirement remains locked until the agent asks a targeted clarification question, at which point the controller reveals the canonical resolution. A final plan submitted before clarification does not count as resolving the ambiguity, even if the controller later discloses the answer to advance the dialogue. This construction tests whether the agent seeks clarification before committing to a choice rather than guessing the hidden value.

Change. A changing episode begins with a provisional requirement that differs from the canonical final requirement. In travel, the revised field may be hotel star rating, total budget, or room count. In shopping, the generator changes one supported equality attribute in an item request while preserving all other encoded conditions. A provisional product is eligible only if it is in stock and satisfies both the provisional item conditions and the applicable user-profile requirements. If the task specifies a cart budget, the generator also requires a full-cart budget witness, obtained by substituting the provisional product into a reference cart. Candidate episodes are skipped when requirement wording is unsupported or ambiguous, stock is unavailable, or the necessary witness cannot be established.

The revision is disclosed only after the agent successfully performs relevant tool-based work under the provisional requirement. In travel, this requires a relevant transport or hotel query, including a matching star filter when the hotel rating changes. In shopping, the agent must add a valid provisional product to the cart, and the product must remain there at the event boundary. The script then explicitly withdraws the provisional value and activates the next value. We support either one revision or two ordered revisions, with each subsequent revision requiring new work after the preceding event. Shopping success additionally requires removing superseded products, adding a final replacement, and verifying the cart after its last modification.

Infeasibility. Previous travel benchmarks have also examined unsolvable requests (Cheng et al., 2026). Our infeasible episodes introduce an independent backup request while keeping the original feasible task active. The generator verifies that the backup has no exact solution in the episode's finite database and stores a private zero-match certificate with an inventory hash. Travel supports hotel star-and-service requests with no exact match, as well as a joint hotel-budget template. For the latter, let $p_{\min}$ denote the lowest valid fixed price per room-night, r the requested number of rooms, n the number of nights, and B_{backup} the backup budget. The request is certified infeasible when

$$rnp_{\min} > B_{\text{backup}}. \tag{2}$$

The certificate also records feasible relaxations, such as reducing one quantity or increasing the budget to the lower bound. This proof applies to the benchmark's fixed-price quote: the request explicitly excludes discounts, taxes, date availability, and capacity. Shopping episodes use an attribute pair for which no product is an exact match. The agent must establish this absence through appropriate tool results, identify the infeasible backup request to the user, avoid substituting an unauthorized near match, and continue satisfying the original requirements.

3.3 Deterministic Interaction Controller

Simulated conversations enable scalable agent evaluation (Wang et al., 2024; Yao et al., 2025; Qian et al., 2025), but simulator reliability and goal drift can affect comparisons among assistants (Dou et al., 2025; Mehri et al., 2026). Our controller preserves the same target task across agent runs.

At turn t, the controller examines the dialogue and tool history h_t and advances the next scripted event only if its trigger $g_j(h_t)$ is satisfied. Otherwise, it selects from the eligible requirements that have not yet been disclosed. Scripted event messages use precomputed canonical text. On ordinary disclosure turns, a language model may select relevant requirement identifiers; the controller filters this selection to the authorized set and renders the corresponding canonical facts in the user message. A requirement is marked as disclosed only once its fact appears in the recorded message. Event-dependent requirements remain locked until the corresponding event occurs. This design prevents the model from changing requirements, revealing future revisions, or exposing evaluator-only answers through free-form generation.

The controller also enforces turn limits and completion criteria. If the agent submits a plan prematurely, the simulator continues by disclosing outstanding requirements rather than accepting the submission. On the last available user turn, remaining ordinary requirements may be disclosed to avoid a dialogue deadlock. Any forced scripted event whose trigger remains unmet is recorded and causes the process check to fail. Only the controller may emit the termination marker, and it does so after all requirements and events have been disclosed and the agent has submitted the domain-specific final block: `<plan>` for travel or `<final_cart>` for shopping. Shopping also requires a successful cart inspection after the latest cart modification. This separation makes task semantics and event order reproducible while allowing the agent's actions to determine when a scripted revision or clarification is disclosed.

3.4 Outcome and Process Verification

For a derived episode i, let $B_i \in \{0, 1\}$ denote the original domain's strict case-level success indicator, and let $P_i \in \{0, 1\}$ indicate whether all required interaction checks pass. Prior work on tool-use evaluation likewise examines intermediate trajectory milestones in addition to final outcomes (Lu et al., 2025). We define joint success as

$$J_i = B_i P_i. \tag{3}$$

Normal episodes retain the base-task evaluation and are excluded from the ACE process denominator. For derived episodes, the verifier recomputes P_i from the raw dialogue, ordered messages, interaction script, and tool results rather than relying on a stored process score. It verifies that all canonical facts and events appeared, that the final response followed them, that the expected final tag and termination marker are present, and that tool calls have unique identifiers and matching replies. Failed calls and unlinked results are not accepted as evidence.

Condition-specific checks enforce the required behavior. Ambiguity requires a targeted question before resolution and disallows any earlier final submission. Change requires successful, relevant work before each revision; shopping additionally requires a post-revision cart inspection confirming that provisional products have been removed and a final replacement is present. Infeasibility requires a valid zero-match certificate, a tool-grounded probe preceding the explanation, and no subsequent contradictory availability claim or forbidden substitution. A limited search result does not establish absence: the verifier requires complete tool-visible candidate sets or, for hotel-budget cases, recomputes the lower bound from an unfiltered hotel result. Natural-language claims are assessed conservatively. Uncertain paraphrases may be referred for independent semantic review, but such review cannot compensate for missing tool evidence or incorrect event order.

We report base success, process success, and joint success by intent type, template, and number of scripted events. Missing expected trajectories remain in the denominator. This decomposition distinguishes failures in constrained planning from failures to clarify requirements, adapt to revisions, or explain infeasibility as user intent evolves.

4 Experiments

4.1 Dataset

The 360 base cases comprise 120 Chinese travel tasks, 120 English travel tasks, and 120 English shopping tasks (50, 50, and 20 across three difficulty levels). We retain normal cases and generate ACE variants only after verifying the required alternative, provisional solution, or zero-match certificate. The fixed manifest records source hashes, template coverage, and skipped variants. Travel cases reuse read-only databases, whereas shopping cases use isolated database copies. Missing trajectories for generated cases are counted as failures.

4.2 Agent Models and Inference Protocol

We evaluate nine agents across all three domain–language settings: DeepSeek-Flash, DeepSeek-V4-Pro, GLM-5.3, GLM-5.3-Flash, Kimi-K3, MiMo-V2.6-Flash, MiMo-V2.6-Pro, Qwen3.8-Flash, and Qwen3.8-Max. DeepSeek-Flash serves as the separate user simulator in both domains, with the controller restricting it to authorized disclosures. Evaluation uses a fixed case manifest and evaluation code. Missing or incomplete trajectories count as failures, and the reported strict automatic scores do not award passes to cases awaiting semantic review.

4.3 Evaluation Metrics

As defined in Section 3.4, normal episodes are evaluated using base success B, and interactive episodes using joint success $J = BP$. We report these rates by domain and condition, alongside generated-case counts to make differences in coverage explicit. Strict automatic scoring treats unresolved natural-language explanations as failures. Semantic review may resolve these judgments but cannot override missing tool evidence or invalid event order.

Model	Normal	Ambiguous	Changing	Infeasible
Chinese travel planning				
DeepSeek-V4-Pro	7.5%	15.0%	12.5%	3.6%
GLM-5.3-Flash	27.5%	38.3%	30.8%	30.9%
Qwen3.8-Flash	13.3%	11.7%	7.5%	12.7%
DeepSeek-Flash	45.8%	47.5%	42.5%	40.0%
MiMo-V2.6-Flash	39.2%	39.2%	34.2%	32.7%
GLM-5.3	48.3%	54.2%	46.7%	47.3%
Kimi-K3	12.5%	0.0%	2.5%	0.0%
Qwen3.8-Max	30.8%	52.5%	50.8%	50.9%
MiMo-V2.6-Pro	**58.3%**	**60.8%**	**51.7%**	**56.4%**
English travel planning				
DeepSeek-V4-Pro	16.7%	16.7%	21.7%	16.4%
GLM-5.3-Flash	46.7%	50.8%	45.8%	52.7%
Qwen3.8-Flash	51.7%	50.8%	37.5%	36.4%
DeepSeek-Flash	58.3%	48.3%	55.0%	47.3%
MiMo-V2.6-Flash	38.3%	45.0%	37.5%	34.5%
GLM-5.3	51.7%	50.8%	48.3%	**63.6%**
Kimi-K3	51.7%	58.3%	58.3%	50.9%
Qwen3.8-Max	60.8%	57.5%	**65.0%**	60.0%
MiMo-V2.6-Pro	**61.7%**	**63.3%**	55.0%	54.5%
Multi-product shopping				
DeepSeek-V4-Pro	74.2%	56.7%	56.4%	31.7%
GLM-5.3-Flash	75.8%	57.5%	51.5%	56.7%
Qwen3.8-Flash	75.8%	55.0%	40.7%	43.3%
DeepSeek-Flash	78.3%	**71.7%**	**66.3%**	**74.2%**
MiMo-V2.6-Flash	78.3%	70.8%	58.2%	70.8%
GLM-5.3	75.0%	52.5%	48.5%	54.2%
Kimi-K3	72.5%	45.8%	50.5%	50.8%
Qwen3.8-Max	74.2%	53.3%	53.8%	50.0%
MiMo-V2.6-Pro	**80.8%**	41.7%	51.6%	61.7%

Table 1: Case-level success (%) in every domain and interaction condition. Normal uses the original task score; Ambiguous, Changing, and Infeasible require both task and process success. Bold marks the highest score within each domain and condition.

4.4 Results and Analysis

Table 1 reports strict case-level scores for each model and condition in Chinese travel, English travel, and shopping. Normal episodes use base-task success B, while the three interactive conditions use joint success $J = BP$. Each model is evaluated on 120 normal and 120 ambiguous cases per setting, as well as 120 changing and 55 infeasible cases per travel language. Shopping includes 120 infeasible cases. Its changing condition contains 91 cases for Qwen3.8-Flash, MiMo-V2.6-Flash, Qwen3.8-Max, and MiMo-V2.6-Pro, and 101 cases for the remaining five agents, reflecting differences between case-generation batches. Consequently, pooling cases gives unequal weight to the settings, and rates computed over different case sets should not be interpreted as paired comparisons.

Performance varies across domains and interaction types. MiMo-V2.6-Pro achieves the highest pooled score (58.5%) and leads all four Chinese-travel conditions. However, its shopping score drops sharply from 80.8% on normal episodes to 41.7% on ambiguous episodes. DeepSeek-Flash leads every interactive shopping condition, scoring 71.7% on ambiguity, 66.3% on change, and 74.2% on infeasibility. In English travel, Qwen3.8-Max leads the changing condition at 65.0%, while GLM-5.3 leads infeasibility at 63.6%. These rank reversals indicate that performance on final-plan construction and interaction handling does not produce a consistent ordering of agents across settings.

Travel failures primarily reflect invalid plans. Across the nine models, Chinese-travel success rates are 31.5%, 35.5%, 31.0%, and 30.5% for normal, ambiguous, changing, and infeasible episodes, respectively. The corresponding English-travel rates are 48.6%, 49.1%, 47.1%, and 46.3%. In this evaluation, every travel case with a valid base plan in the ambiguous and changing conditions also passes the process check: base and joint pass counts coincide (383 and 335 in Chinese; 530 and 509 in English). Checker logs identify invalid city-transportation evidence, insufficient transfer time, and geographic discontinuity as frequent sources of itinerary failure. Among

normal episodes, the city-transportation check fails on 425 Chinese and 260 English model–case evaluations, while the transfer-time check fails on 285 and 226, respectively; individual cases may fail multiple checks. Kimi-K3 illustrates this distinction: it passes the process check on 101 of 120 Chinese ambiguity episodes but produces no valid final itinerary, with transfer-time failures in 105 of its 117 evaluated plans. Infeasibility introduces an additional process bottleneck: among 173 Chinese and 244 English cases with valid base plans, only 151 and 229 also pass the explanation process. The Chinese–English performance gap cannot be attributed to language alone because the two travel sets contain different underlying cases.

Shopping reveals failures in adapting to evolving intent. Pooled shopping success decreases from 76.1% on normal episodes to 56.1% on ambiguity, 53.2% on change, and 54.8% on infeasibility. In ambiguous shopping episodes, 608 model–case evaluations pass the final-cart check and 606 pass both checks. Thus, most joint failures reflect incomplete or incorrect carts, with few otherwise valid carts failing the clarification check. Under change, however, 508 cases pass the base cart check but only 462 pass jointly. Process reports record 171 failures of the final-replacement check, 102 failures of the superseded-product-removal check, and 79 failures of the post-revision cart-verification check. Infeasibility similarly reduces 692 base-correct carts to 592 joint passes, with 131 failed grounding checks and 110 failed probe-evidence checks. These counts overlap because a single trajectory may violate multiple requirements.

Trajectory inspection and interpretation. One shopping change trajectory ends with the correct cart but fails the process check because the agent submits a final answer before the second user revision, forcing the controller to disclose the outstanding event. In a shopping ambiguity trajectory, the agent asks for and receives the missing brand requirement but includes an extra product in the final cart. Runtime completion also affects the results: MiMo-V2.6-Pro leaves 53 of 120 shopping-ambiguity cases incomplete, and 45 of these trajectories explicitly report a time limit. Qwen3.8-Max has 64 timed-out trajectories among 120 Chinese-travel normal cases, which helps explain why its ambiguous and changing scores exceed its normal score in that setting. These examples illustrate how strict joint scoring captures distinct failures in planning, process order, and completion. The reported results combine newer episodes with independently re-evaluated earlier travel trajectories. A GLM-5.3 shopping run also required restoration of source-matched default-quantity metadata in an isolated scoring copy. In addition, shopping change uses unequal case counts across model batches. We therefore avoid causal interpretations of small score differences and report strict automatic infeasibility scores pending semantic adjudication.

5 Conclusion

We introduced DeepPlanning-ACE, a benchmark that extends verifiable long-horizon planning to interactions involving initially ambiguous requirements, revisions after work has begun, and infeasible requests. Episodes are derived from Chinese and English travel-planning tasks and English multi-product shopping tasks only when the defining alternative, revision, or zero-match condition can be verified. By separating canonical intent from conversational disclosure, the deterministic controller makes each episode auditable while allowing agent actions to determine when information is revealed. Joint evaluation requires both a valid final itinerary or cart and the clarification, adaptation, or evidence-grounded explanation specified by the episode.

Evaluation of nine agents shows that success on a fixed planning task does not reliably predict success under evolving intent, and no agent leads across all domains and conditions. Travel performance is often limited by itinerary validity even when interaction checks pass. Shopping trajectories, meanwhile, reveal failures to update or verify carts after revisions and to substantiate infeasibility claims. These complementary failure modes motivate agents that track active requirements, revisit tool-grounded decisions when requirements change, and establish option unavailability from complete evidence. The benchmark's finite inventories and simulated users limit the scope of these conclusions. Evaluation in broader environments and with human users, together with adjudication of unresolved natural-language explanations, remains future work.

AI USE STATEMENT

During the preparation of this work, the authors utilized generative AI tools, specifically Codex, for English language editing and text polishing to improve clarity and readability. Additionally, Codex was employed as a coding assistant to facilitate debugging and accelerate the implementation of experimental scripts. All AI-generated suggestions and code snippets were thoroughly reviewed, verified, and integrated by the authors. The authors retain full responsibility for the scientific integrity, technical accuracy, and final content of this paper.

ETHICS STATEMENT

This work focuses on fundamental advancements in machine learning algorithms and does not directly involve human subjects, sensitive personal data, or high-risk applications. The datasets used in our experiments are standard, publicly available benchmarks, and their utilization complies fully with their respective licenses. We do not foresee any direct negative societal impacts or ethical concerns resulting from this research.

REPRODUCIBILITY STATEMENT

We provide detailed descriptions of the proposed method, model architecture, training procedure, hyperparameters, and evaluation protocol in the main text and supplementary material. All experiments use publicly available benchmark datasets, with preprocessing steps and data splits specified to enable replication. We report the random seeds and implementation details used in our experiments. The code and configuration files will be made available upon publication.

REFERENCES

Xiang Cheng, Yulan Hu, Xiangwen Zhang, Lu Xu, Lide Tan, Zheng Pan, Xin Li, and Yong Liu. Beyond itinerary planning—a real-world benchmark for multi-turn and tool-using travel tasks. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 29200–29251. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.acl-long.1347. URL https://aclanthology.org/2026.acl-long.1347/.

Yao Dou, Michel Galley, Baolin Peng, Chris Kedzie, Weixin Cai, Alan Ritter, Chris Quirk, Wei Xu, and Jianfeng Gao. SimulatorArena: Are user simulators reliable proxies for multi-turn evaluation of AI assistants? In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 35212–35290. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.emnlp-main.1786. URL https://aclanthology.org/2025.emnlp-main.1786/.

Jiarui Lu, Thomas Holleis, Yizhe Zhang, Bernhard Aumayer, Feng Nan, Haoping 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. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 1160–1183. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.findings-naacl.65. URL https://aclanthology.org/2025.findings-naacl.65/.

Shuhaib Mehri, Xiaocheng Yang, Takyoung Kim, Gokhan Tur, Shikib Mehri, and Dilek Hakkani-Tür. Goal alignment in LLM-based user simulators for conversational AI. *Transactions of the Association for Computational Linguistics*, 14:852–876, 2026. doi: 10.1162/tacl.a.687. URL https://aclanthology.org/2026.tacl-1.38/.

Cheng Qian, Zuxin Liu, Akshara Prabhakar, Zhiwei Liu, Jianguo Zhang, Haolin Chen, Heng Ji, Weiran Yao, Shelby Heinecke, Silvio Savarese, Caiming Xiong, and Huan Wang. UserBench: An interactive gym environment for user-centric agents, 2025. URL https://arxiv.org/abs/2507.22034.

Karthik Valmeekam, Matthew Marquez, Alberto Olmo, Sarath Sreedharan, and Subbarao Kambhampati. PlanBench: An extensible benchmark for evaluating large lan-

guage models on planning and reasoning about change. In *Advances in Neural Information Processing Systems*, volume 36, 2023. doi: 10.52202/075280-1693. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/7a92bcdede88c7afd108072faf5485c8-Abstract.html.

Xingyao Wang, Zihan Wang, Jiateng Liu, Yangyi Chen, Lifan Yuan, Hao Peng, and Heng Ji. MINT: Evaluating LLMs in multi-turn interaction with tools and language feedback. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/8a0d3ae989a382ce6e50312bc35bf7e1-Abstract-Conference.html.

Ziyi Wang, Yuxuan Lu, Yimeng Zhang, Pei Chen, Ziwei Dong, Jing Huang, Jiri Gesi, Xianfeng Tang, Chen Luo, Qun Liu, Yisi Sang, Hanqing Lu, Manling Li, Jin Lai, and Dakuo Wang. Trajectory2Task: Training robust tool-calling agents with synthesized yet verifiable data for complex user intents. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 44021–44044. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.acl-long.2037. URL https://aclanthology.org/2026.acl-long.2037/.

Jian Xie, Kai Zhang, Jiangjie Chen, Tinghui Zhu, Renze Lou, Yuandong Tian, Yanghua Xiao, and Yu Su. TravelPlanner: A benchmark for real-world planning with language agents. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 54590–54613. PMLR, 2024. URL https://proceedings.mlr.press/v235/xie24j.html.

Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. τ-bench: A benchmark for tool-agent-user interaction in real-world domains. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/1b126cc38b8638e07bef37e7b2bb72bf-Abstract-Conference.html.

Michael JQ Zhang and Eunsol Choi. Clarify when necessary: Resolving ambiguity through interaction with LMs. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 5541–5558. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.findings-naacl.306. URL https://aclanthology.org/2025.findings-naacl.306/.

Tong Zhang, Peixin Qin, Yang Deng, Chen Huang, Wenqiang Lei, Junhong Liu, Dingnan Jin, Hongru Liang, and Tat-Seng Chua. CLAMBER: A benchmark of identifying and clarifying ambiguous information needs in large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10746–10766. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.578. URL https://aclanthology.org/2024.acl-long.578/.

Yinger Zhang, Shutong Jiang, Renhao Li, Jianhong Tu, Yang Su, Lianghao Deng, Xudong Guo, ChenXu Lv, and Junyang Lin. DeepPlanning: Benchmarking long-horizon agentic planning with verifiable constraints. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7377–7407. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.acl-long.335. URL https://aclanthology.org/2026.acl-long.335/.

Shuyan Zhou, Frank F. Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, Uri Alon, and Graham Neubig. WebArena: A realistic web environment for building autonomous agents. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/4410c0711e9154a7a2d26f9b3816d1ef-Abstract-Conference.html.