PROEVENT: An Event-centric Benchmark for Proactive Agents

Anonymous ACL submission

Abstract

Proactive agents are expected to anticipate user needs and provide autonomous assistance by perceiving environmental context without explicit instructions. A fundamental capability of such agents is to identify and track users' upcoming events, enabling continuous and event-specific assistance. For example, by recording the time and location of a planned hike, an agent can deliver weather reminders in advance or provide navigation support before departure. However, existing works on proactive agents largely overlook event-centric assistance, and the open-ended nature of proactive assistance poses challenges for reliable evaluation.

To bridge these gaps, we introduce PROEVENT, the first event-centric benchmark designed to assess an agent's ability to proactively maintain a user's timetable based on ongoing instant messaging chats. PROEVENT provides synthesized yet realistic chats that consider the dynamic interaction among users, concurrent chat threads, and noise in the real world, and evaluates proactive agents on response timing, single-step response correctness, and multi-step response correctness. Experiments on eight LLMs and pipelines reveal that current agents frequently overact and struggle with event cancellation. Notably, even the state-of-the-art GPT-5.1 provides redundant assistance in 30% of cases and achieves only 26.7% recall in event cancellation scenarios. Further qualitative analysis reveals fundamental limitations of current LLMs as proactive agents, particularly in detecting implicit events and reasoning from the user's first–person perspective.

1 Introduction

Recent advances in large language models (LLMs) have enabled agentic systems to support humans in diverse tasks (Xie et al., 2024; Zhang et al., 2025). However, most existing LLM-based agents remain reactive, largely relying on explicit user instructions, which may result in repetitive human intervention and the risk of missing time-sensitive information. To address these limitations, it is essential to develop proactive agents that can autonomously perceive and respond to their environment.

Events, defined as the activities a user plans to attend, are critical for proactive agents. **For one thing**, anticipating a user's upcoming events enables proactive agents to provide continuous and seamless assistance in advance. For example, when a user schedules a hiking trip, recording the time and location allows agents to proactively remind the user of weather conditions in advance, as well as offer navigation assistance before departure. **For another thing**, maintaining event records enables agents to deliver assistance tailored to specific activities. For instance, a proactive agent may suggest reserving a table for a planned meal or remind the user to prepare for an upcoming meeting.

There have been several proactive agent benchmarks focusing on human-–device interactions (Lu et al., 2024; Yang et al., 2025c; Zhao et al., 2025; Pasternak et al., 2025), aiming to provide immediate assistance for users (e.g., summarizing a webpage when the user opens it). However, they suffer from two limitations. First, to the best of our knowledge, **no existing benchmarks evaluate proactive agents' ability to track users' upcoming events**, which limits their capacity to provide continuous and long-term assistance for future activities. Second, most existing benchmarks **lack reliable evaluation protocols** due to the open-ended nature of the predicted responses. They typically rely on semantic similarity (Yang et al., 2025c) or use LLMs to simulate a human judge (Lu et al., 2024; Pasternak et al., 2025), which may not accurately reflect the correctness or usefulness of the agent response.

In light of these limitations, we propose an Event-centric Benchmark for Proactive Agents (PROEVENT). PROEVENT focuses on a concrete task: maintaining a user's timetable by proactively collecting event information from the ongoing in-

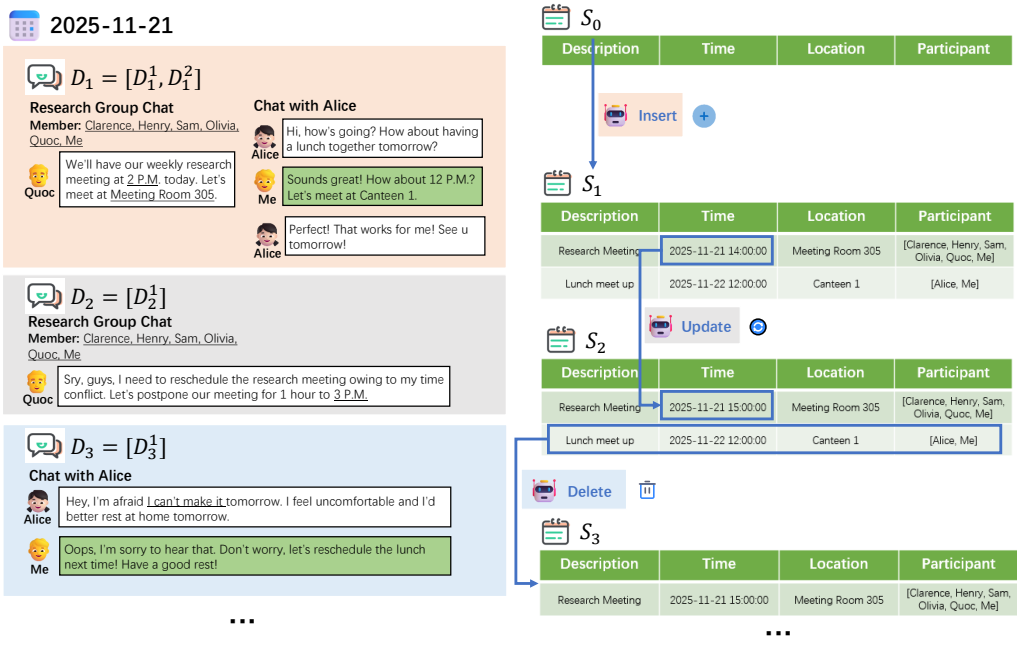


Figure 1: Illustration of the timetable maintenance task. The proactive agent needs to maintain the timetable on the right by proactively seeking information from the chats on the left. D_t refers to the messages received between $t-1$ and t. S_t refers to the user's timetable at time t.

stant messaging chats. We focus on instant messaging chats because they are ubiquitous in the real world yet highly challenging, due to **dynamic speaker interactions**, **concurrent chat threads**, and **pervasive noise** (Zhang et al., 2020; Sapkota et al., 2025). To construct PROEVENT, we develop a data generation pipeline that synthesizes realistic chats with rigorous human quality control. Furthermore, we provide a comprehensive evaluation suite that assesses both the timeliness and the correctness of proactive responses.

In this work, using PROEVENT, we evaluate eight LLMs and pipelines, and find that they tend to overreact for users and struggle with event cancellation scenarios. Furthermore, we show that real-world chat complexities, including dynamic interactions, concurrent chats, and noise, all substantially challenge LLMs' performance as proactive agents. Our qualitative analysis further reveals key deficiencies in current LLMs, including difficulties in capturing implicit events and making decisions from the user's first-person perspective.

2 Related Work

Proactive Agent Benchmark. Recently, several benchmarks have been proposed to evaluate proactive agents. ProactiveBench (Lu et al., 2024) and FingerTip (Yang et al., 2025c) formulate proactive response as a sequential prediction problem, predicting the user's next-step action, whereas PROBE (Pasternak et al., 2025) provides rich contextual information and requires agents to address users' potential needs with tools. Beyond benchmarks, ProactiveVA (Zhao et al., 2025) and ProAgent (Yang et al., 2025b) propose proactive agent pipelines for GUI operations and smart glasses, evaluated in real-world settings with feedback from users. All these works evaluate both response timing and response correctness, with correctness being more challenging to assess due to the open-ended nature of the response. ProactiveBench (Lu et al., 2024) addresses this by fine-tuning a small model to simulate human judgment, whereas FingerTip (Yang et al., 2025c) relies on semantic similarity between predictions and ground truth. However, both approaches lack objective grounding for evaluation. To address this limitation, PROEVENT introduces a comprehensive evaluation suite that enables objective assessment of response correctness based on the correctness of the user's timetable. In addition, with respect to data sources, most of existing works face challenges in acquiring real–world data and therefore rely on synthetic datasets. While we also synthesize chats, we design a pipeline to maximize chat realism and ensure high data quality through rigorous quality control.

Chat Synthesis with LLMs Prior works have applied LLMs to synthesize realistic dialogues by firstly defining the role profiles and then guiding the interactions between roles with structured prompts (Wu et al., 2023; Qiu and Lan, 2024; Wang et al., 2023; Park et al., 2023). While these works provide useful insights, synthesizing realistic instant messaging chats must capture additional key characteristics, including interaction dynamics, concurrent chat threads, and pervasive noise (Zhang et al., 2020; Sapkota et al., 2025). Specifically, in negotiation scenarios, conversational goals often evolve over time, resulting in dynamic interactions among speakers that are challenging for models to accurately track the current interaction state (Peng et al., 2018; Wu et al., 2019). Moreover, prior work has shown that handling concurrent chats introduces additional complexity (Sapkota et al., 2025), while the noise can obscure critical information and distract models (Yang et al., 2023; Sapkota et al., 2025). Accordingly, to account for these challenges, we synthesize realistic chats with multiple negotiation turns during event planning, construct scenarios with varying numbers of concurrent chats, and inject diverse noise, as detailed in Section 4.

3 Problem Formulation and Evaluation

3.1 Problem Formulation

As PROEVENT introduces a novel task, we first formalize the problem and then present the corresponding evaluation suite in next section.

As shown in Figure 1, given a set of N chats $D_t = \{D_t^1, D_t^2, \ldots, D_t^N\}$ received by the user between time $t-1$ and time t, the goal of PROEVENT is to update the user's timetable from the previous state S_{t-1} to a new state S_t.

$$S_t = f(S_{t-1}, D_t) \tag{1}$$

Each timetable $S_t = \{E_1, E_2, \ldots, E_M\}$ consists of M events. Following the widely adopted iCalendar data format, each event contains structured attributes including start time, end time, location, participants, and a brief description. Each chat D_t^i is a multi-turn dialogue between the user and a contact or within a group conversation.

To better reflect real-world proactive agent settings (e.g., GPT-Pulse[1] and Mine Context[2]), we adopt a discrete-time formulation in which agents collect environmental context changes within fixed time intervals. Hence, all chats received within the same interval are processed jointly, and a single event may be updated across multiple time steps.

To avoid repeatedly outputting unchanged events at each time step, we require agents to generate explicit *timetable operations* rather than directly outputting the updated timetable. We define three operations: *Insert*, *Update*, and *Delete*.

Insert$(time, location, participants, description)$: Insert a new event into the timetable with the specified attributes.

Update$(id, attribute, value)$: Modify the specified attribute of the event identified by *id* using the provided value.

Delete(id): Remove the event with the specified *id* from the timetable.

Hence, at each time step t, the agent outputs a list of timetable operations.

3.2 Evaluation Suite

Given an agent's predicted operations at time t_i, denoted as P_{t_i}, and the ground-truth operations G_{t_i}, we evaluate proactive agents from three perspectives: (1) response timing, (2) single-step correctness, and (3) multi-step correctness.

Response Timing. Response timing evaluates whether the agent triggers assistance or stays await at appropriate time steps. We adopt the following two metrics.

False Detection Rate (FDR) measures the proportion of time steps where the agent produces operations when no ground-truth operation is required:

$$\text{FDR} = \frac{\sum_{t_i} \mathbb{I}\big(|P_{t_i}| > 0 \wedge |G_{t_i}| = 0\big)}{\sum_{t_i} \mathbb{I}\big(|G_{t_i}| = 0\big)}. \tag{2}$$

Missed Need Rate (MNR) measures the proportion of time steps where the agent fails to respond when ground-truth operations exist:

$$\text{MNR} = \frac{\sum_{t_i} \mathbb{I}\big(|P_{t_i}| = 0 \wedge |G_{t_i}| > 0\big)}{\sum_{t_i} \mathbb{I}\big(|P_{t_i}| = 0\big)}. \tag{3}$$

Single–step Response Correctness. At each time step t_i, the agent predicts a set of operations P_{t_i}, which is compared against the ground truth set G_{t_i}. To evaluate the response correctness at each time step, we adopt *Precision* and *Recall*:

$$\text{Precision} = \frac{\sum_{t_i} |P_{t_i} \cap G_{t_i}|}{\sum_{t_i} |P_{t_i}|}, \tag{4}$$

[1] https://openai.com/zh-Hans-CN/index/introducing-chatgpt-pulse/

[2] https://github.com/volcengine/MineContext

$$\text{Recall} = \frac{\sum_{t_i} |P_{t_i} \cap G_{t_i}|}{\sum_{t_i} |G_{t_i}|}. \quad (5)$$

Multi–step Response Correctness. We evaluate response correctness in a multi step setting using two metrics: *Event Success Rate (ESR)* and *Timetable Success Rate (TSR)*. *ESR* measures the proportion of events that are correctly predicted after multiple steps of accumulation, where an event is considered correct only if all its attributes match the ground truth. *TSR* measures the proportion of completely correct timetables, where a timetable is considered correct only when all events it contains are correct. Notably, beyond evaluating multi–step response correctness, *ESR* and *TSR* are also designed to resolve ambiguities in single–step evaluation. For example, an *Update* operation may lead to the same final event state as a *Delete* followed by an *Insert*. Although the two responses differ at the single-step level, both should be considered correct. Evaluation on success rate resolves this ambiguity.

4 PROEVENT

4.1 Data Construction

To better understand PROEVENT, we first introduce its construction pipeline, which underpins the analysis of its diversity and quality.

Step one: Contact Pool Construction. To ensure consistency in the generated dialogues, we predefine a profile for each contact, including their name, role, and events to plan. For example, a colleague is more likely to schedule a project meeting, whereas a sports enthusiast may arrange a hiking. Additionally, to promote diversity and realism of these profiles, we draw inspiration from the defined personalities in PersonaChat (Zhang et al., 2018) and define 50 different profiles.

Step two: Chat Synthesis. At this stage, we synthesize chats involving selected contacts from the contact pool. To simulate the dynamic interactions of real–world conversations, we generate dialogues for negotiating and updating planned events with varying numbers of turns. We first create a scheduling trajectory that records the true state of all events at each time step. This trajectory then guides the chat generation, ensuring that each dialogue aligns with the underlying event timeline.

2.1 Scheduling Trajectory Generation.

To generate sequential scheduling trajectories that capture how events evolve over time, we start from an initialized event and iteratively modify its attributes, including time, location, and participants. To enhance the flexibility and diversity of the generated trajectories, we define three atomic operations to model these modifications (Xu et al., 2024; Mathur et al., 2025): (1) *Simple Modification*, which simulates negotiation adjustments (e.g., "I have another meeting at 10 A.M., so can we change the time to 5 P.M.?"); (2) *Timetable Linking*, which references existing or historical events in the timetable (e.g., "Let's meet in the same conference room as last time"); and (3) *Detail Removal*, which represents tentative events with incomplete information (e.g., "Let's meet tomorrow; I'll provide the specific location later"). The outcome of each operation can be either successful or unsuccessful. By randomly selecting an atomic operation and a target attribute at each turn, we generate scheduling trajectories with diversity and realistic variations.

2.2 Chat Generation. Based on the generated scheduling trajectory, we construct a corresponding chat skeleton that serves as a high-level blueprint for dialogue generation (Appendix F). Finally, we use `gpt-3.5-turbo-0613` to generate dialogues that strictly adhere to the ground truths in the scheduling trajectory.

Step three: Noise Injection. Noise is an important factor in real-world proactive agent applications. To construct realistic test cases, we inject three types of noise into all generated chats: (1) *Message-level noise*, messages irrelevant to the target event but interleaved with relevant messages within a chat; (2) *Chat-level noise*, an entire chat unrelated to the target event; and (3) *Event-level noise*, where additional determined or historical events in the timetable may distract the agent from operating on the correct event. In terms of content, we diversify noise to include off-topic messages, discussions about events unrelated to the user, and failed attempts to schedule events (Li et al., 2025; Higashinaka et al., 2021). This variety ensures the evaluation in a realistic chat environment.

Step Four: Chat Scenario Construction. To simulate concurrent chat threads and the periodic context collection setting in the real world, we merge multiple chats and segment them into fixed time windows. Based on these time windows and the generated scheduling trajectories, we derive ground truth operations for each time step by comparing event states across successive windows. In this way, we construct scenarios which require agents to handle multiple chats and events simulta-

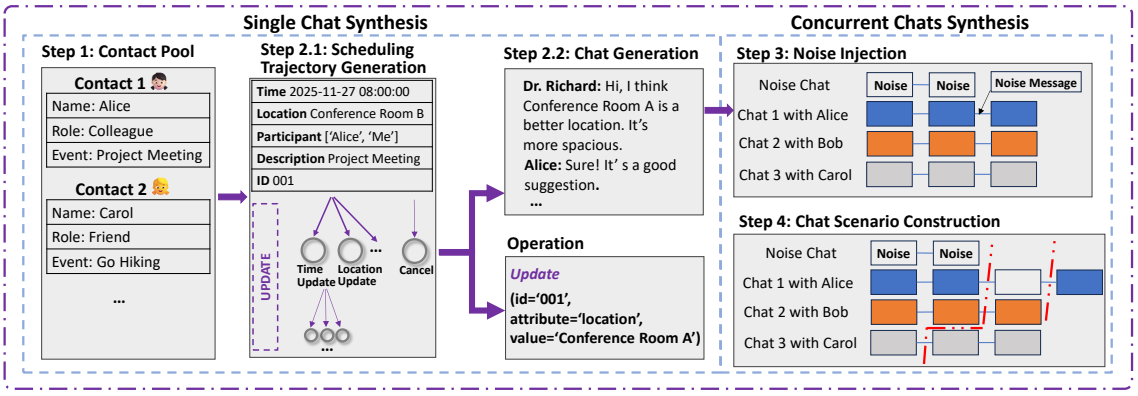


Figure 2: PROEVENT construction pipeline. **Left:** Single chat synthesis. A contact is selected from the contact pool, and a scheduling trajectory is generated to update an attribute of the event. This trajectory then guides the chat generation, producing a single chat. **Right:** To simulate real-world concurrent chat threads, multiple chats are combined, noise is injected, and the chats are segmented into fixed time windows.

neously while obtaining the ground truth operation at each time step.

4.2 Diversity

PROEVENT contains 2,049 single-step questions derived from 807 high-quality chats, 1,827 events and 622 timetables (Figure 3 (b)). Regarding operation types, *Delete* operations are substantially less frequent than *Insert* and *Update*. This is because a *Delete* operation can occur at most once after an event is inserted, whereas insertions may occur without deletion, and multiple updates can be applied to the same event. To ensure this imbalance does not bias the results, we additionally evaluate models on a balanced subset, which yields consistent findings with the full dataset (Appendix A).

Figure 3 (a) demonstrates substantial diversity in event categories. In addition, to reflect real-world chat characteristics, PROEVENT includes scenarios with varying numbers of concurrent chats, different numbers of negotiation turns, and diverse types of noise. The proportions of these categories are illustrated in Figure 3 (c).

4.3 Quality

We perform human verification for each synthesized chat, focusing on whether the dialogue is consistent with the corresponding scheduling trajectory. Each case is independently reviewed by two annotators, and only cases approved by both are retained. As a result, 807 out of the initial 900 chats are kept. The inter-annotator agreement, measured by Cohen's κ, is 0.93, indicating strong consistency between annotators.

Furthermore, to assess alignment with human understanding, we compare human performance with state-of-the-art LLMs on a sampled subset of PROEVENT. This subset consists of 122 questions and preserves the overall distribution of the three operation types in PROEVENT. Human performance is reported as the average across two annotators, with the results presented in Table 1.

5 Experiment

5.1 Setup

We use PROEVENT to evaluate LLM-based proactive agents across three categories: (1) open-source LLMs: including Qwen-3 (Yang et al., 2025a), Deepseek-V3.2 (Liu et al., 2025), and Deepseek-R1 (Guo et al., 2025); (2) proprietary LLMs: GPT-5.1; and (3) proactive-agent pipelines: Proactive, which encourages LLMs to respond more proactively, and ProCoT, which explicitly requires the model to reason about whether a response is necessary (Deng et al., 2023).

Since we do not observe single-step ambiguity among operation types (e.g., an *Update* may equal an *Insert* combined with a *Delete*), we just report precision and recall for single-step performance. For multi-step evaluation, an event is no longer tracked once an incorrect prediction occurs. Moreover, while time and participant fields can be exactly matched to determine correctness, we use LLMs as auxiliary judges to evaluate the correctness of open-ended location descriptions(Appendix E).

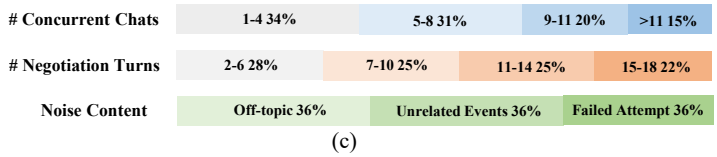
5.2 Results Analysis

Overall performance. GPT-5.1 achieves the best performance in almost all dimensions, with a False

(a)

# Test Cases	Chats		# Operations				# Events	# Timetables
	# Chats	Avg. #Turn	Insert	Update	Delete	No Response		
2049	807	8.26	1818	1962	202	579	1827	622

(b)



(c)

Figure 3: Dataset Statistics. (a) illustrates the diversity of events and their corresponding chat topics. (b) presents the benchmark statistics. (c) Shows the distribution of negotiation turns, concurrent chats, and noise content, reflecting realistic chat characteristics.

Models	**Timing**		**Single-Step**								**Multi-Step**	
			Insert		Update		Delete		Overall		Success Rate	
	FDR ↓	MNR ↓	R	P	R	P	R	P	R	P	Event	Table
Qwen-3 (8B)	65.6%	52.2%	31.1%	20.6%	16.3%	21.6%	3.5%	3.3%	22.4%	20.1%	15.0%	6.9%
Qwen-3 (32B)	79.9%	39.3%	51.7%	25.9%	26.7%	34.5%	21.8%	16.6%	37.9%	27.9%	28.9%	12.1%
Qwen-3 (235B)	54.1%	35.1%	58.5%	39.6%	39.6%	48.4%	25.7%	26.0%	47.5%	42.2%	35.9%	17.2%
DeepSeek-V3.2	93.4%	38.7%	39.3%	31.9%	23.5%	30.6%	**43.1%**	18.0%	31.7%	29.9%	20.8%	8.8%
DeepSeek-R1	66.5%	17.4%	65.2%	37.7%	50.6%	65.4%	22.8%	50.0%	55.9%	46.8%	43.6%	16.9%
GPT-5.1	**38.3%**	**17.0%**	**74.4%**	**62.0%**	**62.4%**	**69.4%**	26.7%	**90.0%**	**66.1%**	**65.6%**	**54.3%**	**30.7%**
Proactive (Deepseek-V3.2)	96.0%	41.0%	40.9%	32.8%	22.9%	28.0%	40.0%	13.7%	32.0%	28.6%	21.5%	9.8%
ProCoT (Deepseek-V3.2)	40.8%	31.0%	56.5%	46.8%	43.6%	67.9%	28.2%	70.4%	48.7%	54.9%	35.9%	19.5%
Human Performance	0%	1.6%	95.7%	94.7%	97.8%	99.2%	96.9%	100%	96.9%	97.5%	94.4%	90.5%

Table 1: Evaluation results on PROEVENT. ↓ indicates that lower values are better, while all other metrics are higher-is-better. We highlight problematic results (FDR and MNR $> 90\%$, while all other metrics are $< 10\%$) and the **best-performing** results except humans. P and R denote Precision and Recall, respectively.

Detection Rate (FDR) of 38% for response timing, an overall recall of 66.1% for single-step response correctness (Table 1). Despite this advantage, the multi-step results indicate that GPT-5.1 can only correctly deal with about half of the events and provides trustworthy services in merely 30% of cases. Similarly, DeepSeek-V3.2 achieves an Event Success Rate of only around 20% and a Timetable Success Rate below 10%, whereas human annotators reach 94.4% and 90.5%. The results underscore that **current LLMs remain far from delivering dependable proactive assistance** and that **PROEVENT presents a challenging benchmark**.

LLMs tend to overreact. Across nearly all evaluated models, the FDR is significantly higher than the Missed Need Rate (MNR), suggesting a systemic bias toward over-responsiveness. For instance, DeepSeek-V3.2 exhibits an FDR more than 50% higher than its MNR (93.4% vs. 38.7%), while GPT-5.1's FDR is more than two times as its MNR (38.3% vs. 17.0%). These results indicate that LLMs frequently trigger proactive actions when no intervention is required, yet they are relatively more capable of avoiding omissions when assistance is actually needed. Furthermore, we observe that the ProCoT pipeline—which explicitly prompts the model to evaluate the necessity of an action before execution—reduces DeepSeek-V3.2's FDR by more than half (from 93.4% to 40.8%). This suggests that deliberative reasoning about response necessity mitigates LLMs' inherent tendency to over-anticipate user needs.

Stronger models are more prudent in deleting events. GPT-5.1, DeepSeek-R1, and the ProCoT strategy achieve the highest overall response correctness, along with the lowest FDR and MNR. Notably, these models also exhibit the lowest recall for *Delete* operations (26.7%, 22.8%, and 28.2%, respectively). In contrast, although the weaker DeepSeek-V3.2 attains a substantially higher recall for *Delete* operations, its precision is only 18.0%, compared with 90.0% for GPT-5.1. This discrep-

ancy indicates that models with stronger response timing judgment and higher overall correctness tend to adopt a more conservative strategy when deciding to delete events, avoiding deletions unless there is clear evidence. A similar bias has also been observed in prior studies, where LLMs tend to avoid commonsense reasoning (Li et al., 2024; Fu et al., 2025) and show a strong preference in binary Yes/No questions, In PROEVENT, such prudence is particularly relevant to the scenarios where users quit from a planned activity, or when cancellation cues are subtle (Section 5.3).

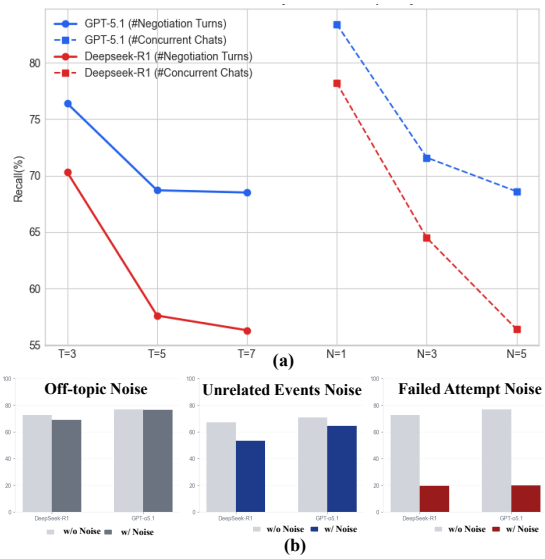


Figure 4: Effects of dynamic interactions ((a), left), concurrent chats ((a), right), and noise (b) on the performance of Deepseek-R1 and GPT-5.1. Here, T denotes the number of negotiation turns, and N denotes the number of concurrent chats.

Real-world chat complexities pose significant challenges for LLMs. Since PROEVENT incorporates dynamic user interactions, concurrent chat threads, and diverse forms of noise, we examine how these factors affect model performance. Our results show that recall consistently declines across all evaluated models as the number of negotiation turns increases and the volume of concurrent threads grows (Figure 4 (a)). With respect to noise, we find that **the semantic content of noise affects LLMs more than its positioning**. While injecting off-topic noise at varying levels does not significantly degrade performance (Appendix C), replacing such noise with discussions of events unrelated to the user or failed planning attempts leads to substantial drops in recall and precision, respectively (Figure 4(b)). These findings suggest that while LLMs can understand the semantics of chats, they still struggle to robustly infer underlying human intentions in complex, realistic chat settings.

5.3 Discussion

In this section, we present qualitative analysis observations and discuss LLMs' deficiencies as proactive agents.

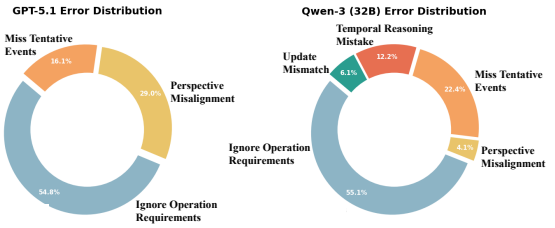


Figure 5: Error distributions of GPT-5.1 and Qwen-3.

GPT-5.1 is prone to overlooking events implicitly expressed in narrative styles. Despite its state-of-the-art reasoning capabilities, GPT-5.1 exhibits a notable failure rate when task instructions are embedded in natural, narrative dialogue. For instance, in Figure 6 (c), the model fails to trigger a participant update when a user provides the implicit instruction to "invite the same people". Given the model's high performance in structurally explicit tasks, we hypothesize that the cause is not a lack of reasoning capacity, but rather a perceptual bias where the model ignores operational requirements in the absence of explicit structural cues.

To verify this assumption, we conducted an ablation study on a curated set of these error cases by introducing two interventions: (1) substituting narrative instructions with direct expressions (e.g., specific names), and (2) appending an update prompt to the dialogue (see Appendix B). As shown in Table 2, both interventions markedly improve GPT-5.1's performance. Notably, the update prompt increases Recall by 32.0%, an effect not observed in other models. These results confirm that while GPT-5.1 possesses the underlying reasoning capabilities, its performance as a proactive agent is highly sensitive to the explicitness of information within the context.

Crucial Reasoning Abilities for Proactive Agents. By comparing the error distributions of GPT-5.1 and Qwen-3-32B (Figure 5), we identify two error categories that are present in Qwen-3 but absent in GPT-5.1: temporal reasoning mistakes and update mismatch. This contrast reveals two

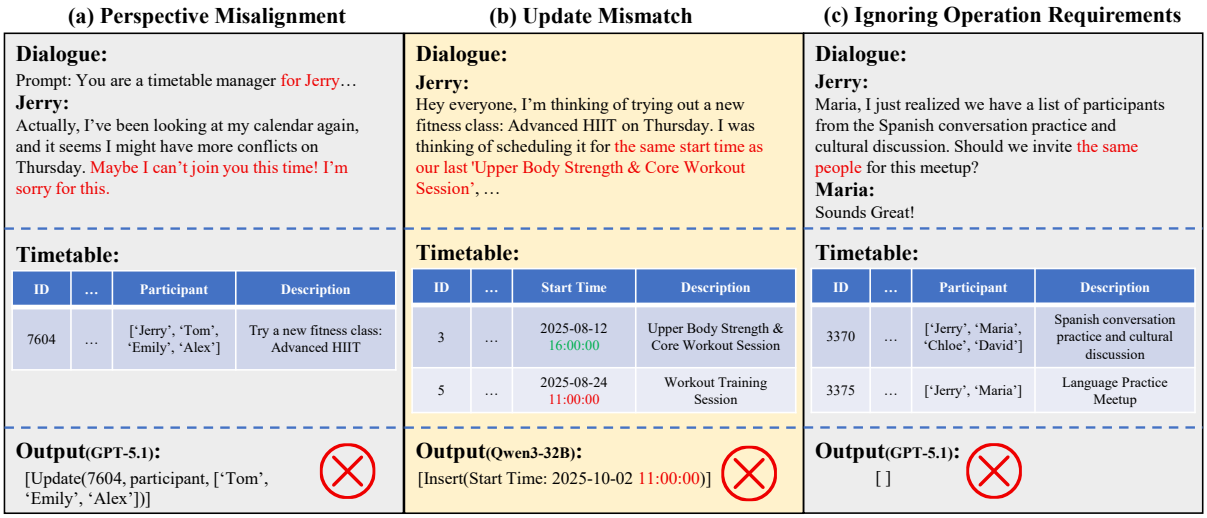


Figure 6: Error cases. LLMs make mistakes when reasoning from the user's first-person perspective, updating one of multiple events in the timetable, or missing required operations implicitly implied by the chats. Cases (a) and (c) are collected from GPT-5.1, while case (b) is collected from Qwen-3-32B. Additional cases involving temporal reasoning errors and missing tentative events are provided in Appendix 8.

	Recall	Precision
GPT-o5.1	56.0%	77.8%
GPT-o5.1 + direct expression	76.0%	90.5%
GPT-o5.1 + update prompt	88.0%	88%
Δ	**32%**	**10.2%**
Qwen-3 (32B)	44.0%	55.0%
Qwen-3 (32B) + direct expression	29.2%	70.0%
Qwen-3 (32B) + update prompt	44.0%	91.7%
Δ	0%	**36.7%**

Table 2: Effect of making implicit narratives explicit and adding an update prompt to the chats. Δ denotes the performance gain, and we highlight **improved results**.

critical reasoning abilities that distinguish stronger models from weaker ones: temporal reasoning and memory-enhanced reasoning. In Qwen-3, temporal reasoning errors primarily arise from incorrect transformations between dates and weekdays, leading to erroneous scheduling decisions. This suggests that models lacking robust temporal reasoning may benefit from auxiliary tools, such as calendar or calculator utilities, to support precise temporal computation (Li et al., 2023; Parisi et al., 2022). Furthermore, as illustrated in Figure 6 (b), update mismatch refers to cases where the agent modifies an incorrect timetable entry or retrieves information from an unrelated historical event. This error reflects a deficiency in memory-enhanced reasoning, which requires the agent to accurately retrieve and align relevant information across multiple events.

LLMs lack a first-person user perspective when making decisions. To provide effective assistance in dynamic and complex environments, proactive agents must consistently reason from the user's first-person perspective. However, our analysis reveals a systematic failure of LLMs to make decisions from the user-centric viewpoint. As illustrated in Figure 6 (a), when the user (Jerry) states that he will no longer attend an activity, the correct action is to perform a *Delete* operation that removes the event from Jerry's personal timetable. Instead, the model frequently performs an *Update* operation that merely removes "Jerry" from the participant list while retaining the event itself. This indicates that LLMs reason about events from a third-person perspective rather than adopting the user–centered viewpoint (Hou et al., 2024; Cheng et al., 2024). This limitation poses a challenge for deploying LLMs as reliable proactive agents.

6 Conclusion

We introduce PROEVENT, the first event-centric benchmark designed to evaluate proactive agents' ability to track users' upcoming events from instant messaging chats. Through a comprehensive evaluation of eight LLMs, we diagnose their systematic biases in both response timing and content. Our analyses further reveal fundamental deficiencies of current LLMs as proactive agents, particularly in real-world application scenarios. We hope that PROEVENT will facilitate future research on enhancing LLMs' capabilities for proactive event tracking in real-world settings.

Limitations

While constructing PROEVENT, we generate chats using LLMs. Although we promote diversity by varying event types and incorporating real-world dynamics such as negotiation complexity, concurrency, and noise, the generated dialogues still tend to be structured and repetitive, lacking the spontaneity and variability of human conversations. Moreover, real-world event planning often involves more complex coordination across multiple chats and participants, whereas we currently assume that all negotiations for an event occur within a single chat. In future work, we plan to refine and expand PROEVENT to further enhance linguistic diversity and interaction complexity.

References

Sijie Cheng, Zhicheng Guo, Jingwen Wu, Kechen Fang, Peng Li, Huaping Liu, and Yang Liu. 2024. Egothink: Evaluating first-person perspective thinking capability of vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14291–14302.

Yang Deng, Lizi Liao, Liang Chen, Hongru Wang, Wenqiang Lei, and Tat-Seng Chua. 2023. Prompting and evaluating large language models for proactive dialogues: Clarification, target-guided, and non-collaboration. *arXiv preprint arXiv:2305.13626*.

Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, and 1 others. 2025. Mme: A comprehensive evaluation benchmark for multimodal large language models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Ryuichiro Higashinaka, Masahiro Araki, Hiroshi Tsukahara, and Masahiro Mizukami. 2021. Integrated taxonomy of errors in chat-oriented dialogue systems. In *Proceedings of the 22nd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 89–98.

Guiyang Hou, Wenqi Zhang, Yongliang Shen, Zeqi Tan, Sihao Shen, and Weiming Lu. 2024. Egosocialarena: Benchmarking the social intelligence of large language models from a first-person perspective. *arXiv preprint arXiv:2410.06195*.

Guanzhen Li, Yuxi Xie, and Min-Yen Kan. 2024. Mvp-bench: Can large vision–language models conduct multi-level visual perception like humans? *arXiv preprint arXiv:2410.04345*.

Minghao Li, Yingxiu Zhao, Bowen Yu, Feifan Song, Hangyu Li, Haiyang Yu, Zhoujun Li, Fei Huang, and Yongbin Li. 2023. Api-bank: A comprehensive benchmark for tool-augmented llms. *arXiv preprint arXiv:2304.08244*.

Zuoou Li, Weitong Zhang, Jingyuan Wang, Shuyuan Zhang, Wenjia Bai, Bernhard Kainz, and Mengyun Qiao. 2025. Towards effective mllm jailbreaking through balanced on-topicness and ood-intensity. *arXiv preprint arXiv:2508.09218*.

Aixin Liu, Aoxue Mei, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, and 1 others. 2025. Deepseek-v3. 2: Pushing the frontier of open large language models. *arXiv preprint arXiv:2512.02556*.

Yaxi Lu, Shenzhi Yang, Cheng Qian, Guirong Chen, Qinyu Luo, Yesai Wu, Huadong Wang, Xin Cong, Zhong Zhang, Yankai Lin, and 1 others. 2024. Proactive agent: Shifting llm agents from reactive responses to active assistance. *arXiv preprint arXiv:2410.12361*.

Niharika Mathur, Tamara Zubatiy, Agata Rozga, Jodi Forlizzi, and Elizabeth Mynatt. 2025. " sometimes you need facts, and sometimes a hug": Understanding older adults' preferences for explanations in llm-based conversational ai systems. *arXiv preprint arXiv:2510.06697*.

Aaron Parisi, Yao Zhao, and Noah Fiedel. 2022. Talm: Tool augmented language models. *arXiv preprint arXiv:2205.12255*.

Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. 2023. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th annual acm symposium on user interface software and technology*, pages 1–22.

Gil Pasternak, Dheeraj Rajagopal, Julia White, Dhruv Atreja, Matthew Thomas, George Hurn-Maloney, and Ash Lewis. 2025. Beyond reactivity: Measuring proactive problem solving in llm agents. *arXiv preprint arXiv:2510.19771*.

Baolin Peng, Xiujun Li, Jianfeng Gao, Jingjing Liu, and Kam-Fai Wong. 2018. Deep dyna-q: Integrating planning for task-completion dialogue policy learning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2182–2192.

Huachuan Qiu and Zhenzhong Lan. 2024. Interactive agents: Simulating counselor-client psychological counseling via role-playing llm-to-llm interactions. *arXiv preprint arXiv:2408.15787*.

Sagar Sapkota, Mohammad Saqib Hasan, Mubarak Shah, and Santu Karmaker. 2025. Multi-party conversational agents: A survey. *arXiv preprint arXiv:2505.18845*.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. 2023. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291*.

Cheng-Kuang Wu, Wei-Lin Chen, and Hsin-Hsi Chen. 2023. Large language models perform diagnostic reasoning. *arXiv preprint arXiv:2307.08922*.

Yuexin Wu, Xiujun Li, Jingjing Liu, Jianfeng Gao, and Yiming Yang. 2019. Switch-based active deep dyna-q: Efficient adaptive planning for task-completion dialogue policy learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 7289–7296.

Jian Xie, Kai Zhang, Jiangjie Chen, Tinghui Zhu, Renze Lou, Yuandong Tian, Yanghua Xiao, and Yu Su. 2024. Travelplanner: A benchmark for real-world planning with language agents. *arXiv preprint arXiv:2402.01622*.

Zhenyu Xu, Hailin Xu, Zhouyang Lu, Yingying Zhao, Rui Zhu, Yujiang Wang, Mingzhi Dong, Yuhu Chang, Qin Lv, Robert P Dick, and 1 others. 2024. Can large language models be good companions? an llm-based eyewear system with conversational common ground. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 8(2):1–41.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025a. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Bufang Yang, Lilin Xu, Liekang Zeng, Kaiwei Liu, Siyang Jiang, Wenrui Lu, Hongkai Chen, Xiaofan Jiang, Guoliang Xing, and Zhenyu Yan. 2025b. Contextagent: Context-aware proactive llm agents with open-world sensory perceptions. *arXiv preprint arXiv:2505.14668*.

Qinglong Yang, Haoming Li, Haotian Zhao, Xiaokai Yan, Jingtao Ding, Fengli Xu, and Yong Li. 2025c. Fingertip 20k: A benchmark for proactive and personalized mobile llm agents. *arXiv preprint arXiv:2507.21071*.

Tianqing Yang, Tao Wu, Song Gao, and Jingzong Yang. 2023. Dialogue logic aware and key utterance decoupling model for multi-party dialogue reading comprehension. *IEEE Access*, 11:10985–10994.

Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. 2018. Personalizing dialogue agents: I have a dog, do you have pets too? *arXiv preprint arXiv:1801.07243*.

Yao Zhang, Zijian Ma, Yunpu Ma, Zhen Han, Yu Wu, and Volker Tresp. 2025. Webpilot: A versatile and autonomous multi-agent system for web task execution with strategic exploration. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 23378–23386.

Zheng Zhang, Ryuichi Takanobu, Qi Zhu, MinLie Huang, and XiaoYan Zhu. 2020. Recent advances and challenges in task-oriented dialog systems. *Science China Technological Sciences*, 63(10):2011–2027.

Yuheng Zhao, Xueli Shu, Liwen Fan, Lin Gao, Yu Zhang, and Siming Chen. 2025. Proactiveva: Proactive visual analytics with llm-based ui agent. *arXiv preprint arXiv:2507.18165*.

A Evaluation on a balanced subset

We sample a subset with a balanced distribution among *Insert*, *Update*, *Delete*. Specifically, we totally sample 142 questions, including 130 *Insert*, 135 *Update*, and 104 *Delete*. The result is shown in Table **??**. The results demonstrate similar patterns to the results of the full dataset: LLMs achieve a significantly lower recall on *Delete* compared with the other two operations. The results also indicate that although *Delete* takes up a relatively small proportion of the entire dataset, it still provides a reliable evaluation.

Models	Single-Step							
	Insert		Update		Delete		Overall	
	R	P	R	P	R	P	R	P
GPT	72.3%	65.7%	54.1%	47.4%	20.2%	100%	50.9%	59.1%
Deepseek-R1	50.4%	32.2%	45.9%	47.0%	22.1%	100%	40.8%	41.9%
Qwen3 (235B)	55.7%	44.8%	34.8%	40.2%	18.3%	82.6%	37.6%	45.9%

Table 3: Evaluation results on a sampled subset, with a balanced distribution among *Insert*, *Update*, and *Delete*. The results show the same trend as the results of full PROEVENT: The recall of *Delete* is significantly lower than the other two operations, while the precision of *Delete* remains high.

B A case for the explicit expression and the update prompt

Figure 7 demonstrates a case for explicit expression and the update prompt. The case shows that after explicitly mentioning the names of participants or adding a prompt to remind LLMs to update both correct the original wrong answers.

C LLMs' robustness on real-world chat complexities

The results in Table 3 demonstrate that almost all models' performance degrades when the number of negotiation turns and the number of concurrent chats grow. As for noise, the effect of off-topic noise is negligible, while the discussions on an event unrelated to the user and a failed attempt to plan an event lead to a significantly lower recall and precision, respectively.

Models	Negotiation Turn			Concurrent Chat			Noise							
	T=3	T=5	T=7	N=1	N=3	N=5	w/o		w/ OT		w/ UE		w/ FA	
							R	P	R	P	R	P	R	P
DeepSeek-V3.2	46.5%	34.2%	27.6%	50.0%	44.4%	30.0%	53.2%	58.3%	58.9%	64.7%	53.2%	46.7%	54.4%	19.9%
DeepSeek-R1	70.3%	57.6%	56.3%	78.2%	64.5%	56.4%	67.1%	72.6%	60.8%	69.1%	53.2%	54.5%	50.6%	19.5%
GPT-5.1	**76.4%**	**68.7%**	**68.5%**	**83.4%**	**71.6%**	**68.6%**	**70.9%**	**76.7%**	**69.6%**	**76.4%**	**64.6%**	**73.9%**	**60.8%**	19.8%
Proactive (Deepseek-V3.2)	45.9%	35.3%	28.5%	48.3%	43.2%	32.8%	51.9%	55.4%	61.4%	68.5%	49.4%	34.8%	49.4%	17.3%
ProCoT (Deepseek-V3.2)	63.8%	55.1%	51.8%	53.6%	46.6%	44.6%	63.3%	72.5%	65.2%	75.8%	59.5%	65.3%	63.3%	**23.7%**

Table 4: LLMs' performance with different negotiation turn numbers, concurrent chat numbers, and diversifying noise. R refers to Recall, while P refers to Precision. OT refers to the off-topic messages. UE means the discussions about an event unrelated to the user. FA means a failed attempt to plan an event. We highlight the models' **best performance**.

D Cases on temporal reasoning mistakes and tentative events

Figure 8 (a) demonstrates a case for temporal reasoning mistakes, when Qwen3-32B fails to transform the weekday into a date. As for the tentative events (Figure 8 (b)), LLMs fail to insert an event when the date is set and the time is to be determined. This reflects that LLMs may lack an ability to handle uncertainty.

E Judging the Correctness of Location with LLMs

The description of a location can be diverse and linguistically varied even when referring to the same physical entity. For example, a meeting might be described as occurring in "Room 305," "the third-floor

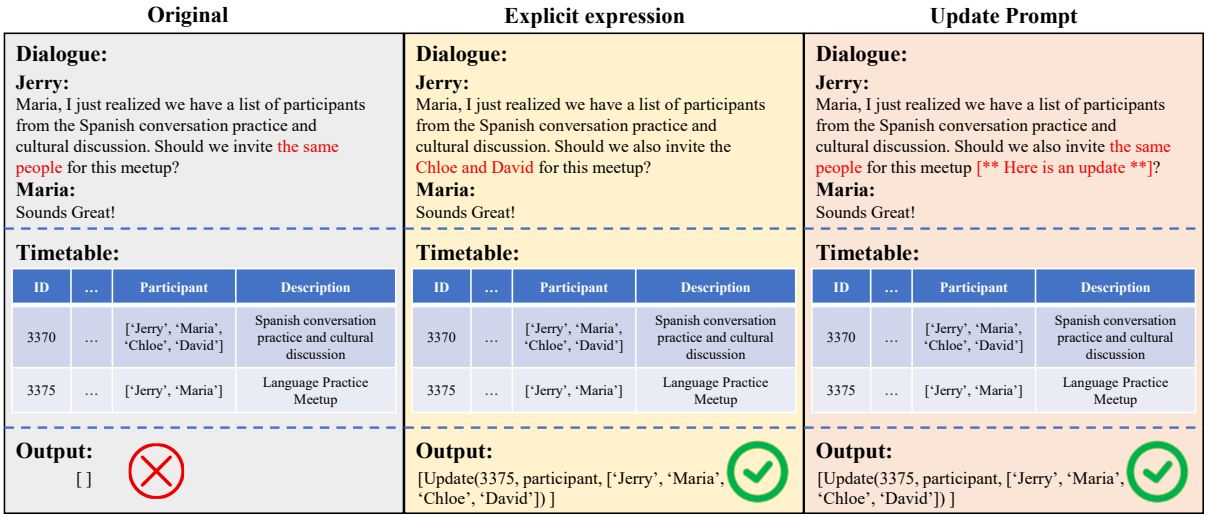


Figure 7: A case for demonstrating the effect of explicit expression and the update prompt.

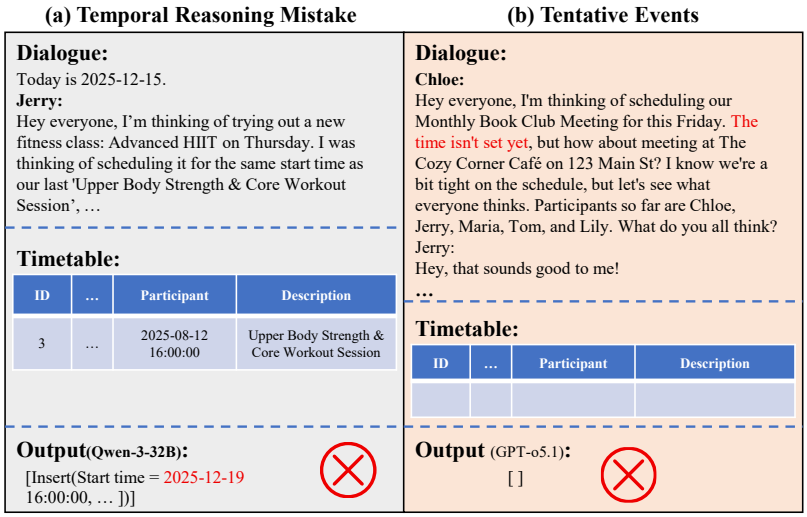


Figure 8: The case for temporal reasoning mistakes and tentative events.

conference room," or "305 Meeting Hall." Such variations make exact string matching an unreliable metric for evaluation. Hence, we use GPT-5.1 with a 4-shot setting to handle the location's open-ended nature and evaluate the semantic equivalence between the predicted location and the ground truth.

The prompt is designed to provide the model with the task instruction and chat context. The specific prompt structure is detailed as follows. Besides, we compare the judgment consistency between GPT-5.1 and humans on 100 randomly sampled cases; the results show that they are **100%** consistent, validating the reliability of using an LLM as a proxy evaluator for this task.

```
You are a meticulous assistant tasked with evaluating the correctness of a proactive agent's location extraction.

Input:
- Ground Truth Location: [ground truth location]
- Predicted Location: [predicted location]
- Chat Context: [chat context related to the location]

Task:
```

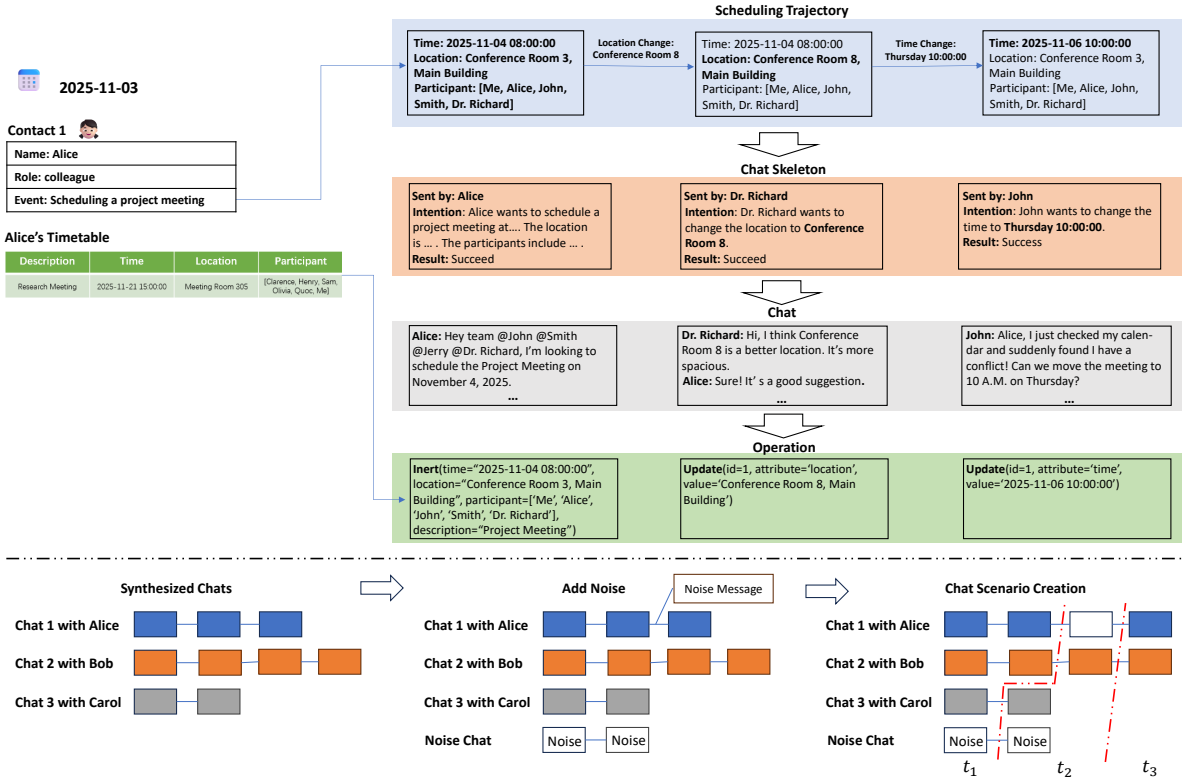


Figure 9: A specific case of our chat synthesis pipeline.

```
Determine whether the predicted location refers to the same specific place as the
ground-truth location. Consider abbreviations, synonyms, hierarchical descriptions,
and cases where the location is left implicit.

Output Format:
Return **CORRECT** if the two locations refer to the same place; otherwise, return
**INCORRECT**.
```

F A case of our chat synthesis pipeline

Figure 9 illustrates a concrete example of our chat synthesis pipeline. We first select a contact from the contact pool and generate scheduling trajectories to simulate the event planning process. Based on these trajectories, we construct a chat skeleton by specifying the event update proposer and organizing updates in a narrative form. Guided by the skeleton, we generate the chat and derive ground-truth operations by tracking changes in the scheduling trajectory. Finally, the lower part of Figure 9 shows how multiple chats are combined to form a concurrent chat scenario.

G The prompts for evaluating LLMs

We use the following prompts to evaluate LLMs and provide four examples (*Insert*, *Update*, *Delete* and no response) for further clarifying the definition of each operation. The prompt can also be found in the questions of our dataset.

```
You are a scheduling manager for Jerry. Given a dialogue and the current timetable,
identify the required schedule operations.

Rules:
- INSERT a new event when a previously unscheduled event is confirmed.
```

- UPDATE an existing event when its start time, end time, location, or participants change.
- DELETE an event when it is explicitly cancelled.
- Output [] if no operation is required.

The current timetable contains confirmed and historical events and serves as contextual memory.
If any information is unspecified, leave the corresponding field empty.
If the date is fixed but the time is unknown, output the date only.
Return only the operation results without any explanation.

Operation formats:
1. INSERT: (INSERT, {"start_time": "%Y-%m-%d %H:%M:%S",
"end_time": "%Y-%m-%d %H:%M:%S",
"location": str,
"participant": list,
"description": str})

2. UPDATE: (UPDATE, {"id": int,
"attribute": str,
"value": str})

3. DELETE: (DELETE, {"id": int})