SOP-Maze: Evaluating Large Language Models on Complicated Business Standard Operating Procedures

Anonymous ACL submission

Abstract

As large language models (LLMs) are widely deployed as domain-specific agents, many benchmarks have been proposed to evaluate their ability to follow instructions and make decisions in real-world scenarios. However, business scenarios often involve complex standard operating procedures (SOPs), and the evaluation of LLM capabilities in such contexts has not been fully explored. To bridge this gap, we propose SOP-Maze, a benchmark constructed from real-world business data and adapted into a collection of 397 tasks from 23 complex SOP scenarios. We further categorize SOP tasks into two broad classes: Lateral Root System (LRS), representing wide-option tasks that demand precise selection; and Heart Root System (HRS), which emphasizes deep logical reasoning with complex branches. Extensive experiments reveal that nearly all state-of-the-art models struggle with SOP-Maze. We conduct a comprehensive analysis and identify three key error categories: (i) route blindness: difficulty following procedures; (ii) conversational fragility: inability to handle real dialogue nuances; and (iii) calculation errors: mistakes in time or arithmetic reasoning under complex contexts. The systematic study explores LLM performance across SOP tasks that challenge both breadth and depth, offering new insights for improving model capabilities. We have open-sourced our work on `https://anonymous.4open.science/r/SOP-Maze-077B`.

1 Introduction

In recent years, LLMs have made significant progress in natural language understanding and generation (Kumar, 2024; Qin et al., 2024a; Wang et al., 2025). The milestone is the significant improvement in instruction-following ability (Wei et al., 2021; Ouyang et al., 2022; Wang et al., 2023; Zhang et al., 2025; Li et al., 2025a), which has driven their gradual deployment as professional agents in domain-specific applications.

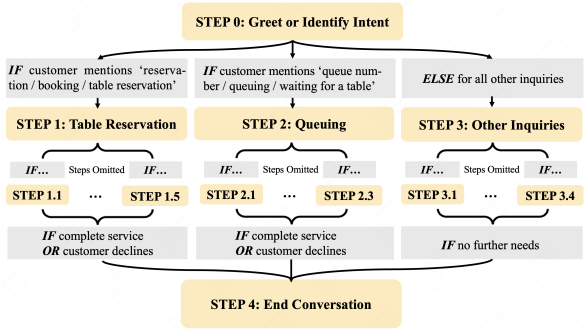

Figure 1: An example of business SOPs.

To evaluate this capability, considerable achievements have been made in generic and compositional instruction scenarios (Yao et al., 2023; Sun et al., 2024; Li et al., 2025b). However, many real-world applications of LLMs, such as business scenarios, are driven by more complex standard operating procedures (SOPs) (Grohs et al., 2023; Kourani et al., 2024a). SOPs define a standard route of thinking or task execution for models involving conditional branches or complex constraints, are more challenging than regular instruction-following tasks (Figure 1); meanwhile, the model is expected to comply with procedures and produce the required output robustly, even when given noisy input. This reveals a significant gap between existing benchmarks and requirements from business SOPs.

To this end, this paper introduces SOP-Maze, a benchmark constructed from real-world business data. SOP-Maze is developed through human-in-the-loop refinement, resulting in 397 instances across 23 distinct business scenarios; each SOP task incorporates a nested or intertwined execution logic route, with an average specification length of 5,040 tokens. We innovatively organize SOP tasks in SOP-Maze into two categories, based on the context and characteristics of the SOPs. As illustrated in Figure 2, **Lateral Root System (LRS)**

[1] represents wide SOP tasks with relatively shallow branching structure, requiring the model to make accurate choice among alternatives with procedure compliance. **Heart Root System (HRS)**[1] captures deep SOP tasks characterized by long and intricate logical chains, in this case, successful task completion requires the model to faithfully traverse a multi-step reasoning path and preserve contextual consistency until reaching the final decision. Together, LRS and HRS probe LLM capabilities across both breadth and depth dimensions of complex business SOPs under noisy conditions. To the best of our knowledge, SOP-Maze is the first benchmark evaluating the practical instruction-following capability of models in business SOP scenarios.

Based on SOP-Maze, we conduct a comprehensive evaluation of 18 prominent LLMs. To ensure reliability, we adopt a JSON schema based evaluation that is model-free, highly efficient, and capable of producing precise scores. Extensive experiments reveal that the benchmark poses substantial challenges, and prominent LLMs consistently fall short of following complex business SOPs, providing new insights into the practical capabilities of LLMs. Our contributions are as follows:

1. We propose SOP-Maze, the first SOP benchmark for real-world business scenarios, concluding 397 difficult tasks derived from 23 real-world business scenarios.

2. We categorize SOP tasks into two types: LRS and HRS, to evaluate LLM capabilities across both the breadth and depth dimensions of complex business SOP scenarios.

3. Extensive experiments on 18 SOTA models reveal the gap between current LLM capabilities and the requirements of business SOPs. Our analysis reveals three key limitations: (i) route blindness; (ii) conversational fragility; and (iii) calculation errors, providing new insights into model capabilities.

2 Related works

LLM Benchmarks for Instruction Following As LLMs are increasingly used in real-world tasks, instruction-following capability has become a key metric for evaluating model practicality (Zhou et al., 2023a; Qin et al., 2024b; He et al., 2024b). Early works focused on single-turn dialogues with simple constraints (e.g. semantic and format constraints) (Zhou et al., 2023b; Xia et al., 2024; Tang et al., 2024), which limited their applicability. Later works moved toward more real-world scenarios: CELLO (He et al., 2024a) generated complex instructions from task descriptions and users' text input, Complexbench (Wen et al., 2024) synthesized instructions using real-world data, adding constraints based on fixed patterns. Furthermore, Guidebench (Diao et al., 2025) raises task difficulty by incorporating domain-specific conditions and system feedback. However, these approaches lack a complex and realistic standard operation procedures in instructions, which is a critical factor in complex business tasks and pose challenges to evaluating LLMs in business scenarios.

LLM Benchmarks for Business Prevailing benchmarks primarily focus on Business Process Management (BPM), evaluating LLMs in tasks like modeling and optimizing complex business processes (Berti et al., 2024; Fahland et al., 2024; Kourani et al., 2024b). For instance, BP^{C} (Fournier et al., 2024) proposes a benchmark to assess LLM capabilities in causal reasoning and explaining decision points within business processes. SAPM (Rebmann et al., 2024) evaluates LLMs on semantic-aware tasks, such as anomaly detection and next activity prediction in process mining. Furthermore, SOP-Bench (Nandi et al., 2025) and SOPBench (Li et al., 2025c) evaluate language agents on their ability to execute precise tool invocations in compliance with processes. In contrast, SOP-Maze focuses on evaluating LLMs in answering questions or making decisions based on standard procedures, rather than process modeling, distinguishing it from these benchmarks.

3 The SOP-Maze Benchmark

3.1 SOP-Maze Overview

SOP-Maze is designed to evaluate LLM capabilities to effectively assist users in complex real-world business SOP scenarios. SOP-Maze encompasses 23 business scenarios (10 in HRS and 13 in LRS), comprising a total of 397 tasks. Comprehensive statistics are presented in Figure 3.

3.2 Data Curation Process

As shown in Figure 2, each task prompt in SOP-Maze comprises 4 key components:

1. **Objective** defines the background, model role, and task objectives.

2. **Standard Operating Procedures** specifies the execution logic and operation details.

[1] **Lateral Root System** and **Heart Root System** are terms borrowed from plant root morphology in botany.

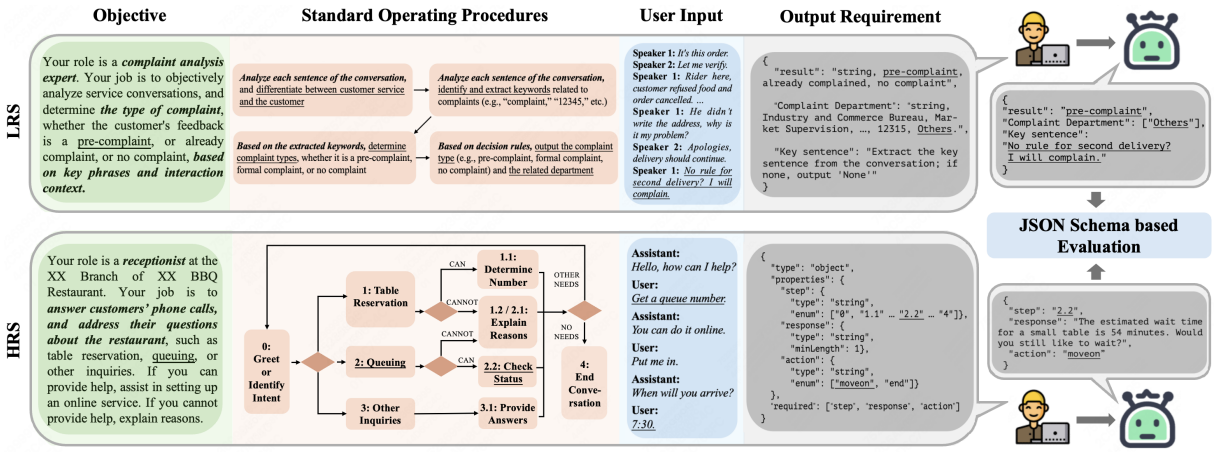

Figure 2: Illustration of SOP-Maze. Based on the context and characteristics of the SOPs, business SOP tasks are categorized into two types, LRS and HRS. Each task prompt comprises 4 key components: Objective, Standard Operating Procedures, User Input and Output Requirement. After the LLM generates an output, it is assessed using JSON Schema based Evaluation.

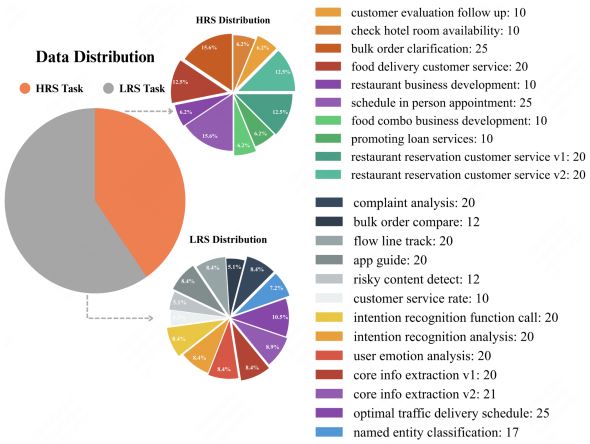

Figure 3: Task distribution

3. **User Input** represents the user-provided data requiring processing.

4. **Output Requirement** mandates adherence to a predetermined output format specification.

The following subsections demonstrate the four stages involved in the construction of the SOP-Maze dataset: (i) Data collection, (ii) SOP refinement, and (iii) User input choice, followed by (iv) the statement about benchmark quality control.

3.3 Data Collection

With user consent, we collect 300,000 data records from our own API router logs, capturing internal business department invocations of LLM for business task execution with user consent. Following rule-based filtering to control data volume, we perform semantic clustering on the dataset. Based on clustering results, we manually select the 23 largest clusters—representing the most frequently occurring business scenarios—to constitute the 23 scenarios of SOP-Maze.

3.4 SOP Refinement

In practical user scenarios, SOPs are incrementally developed through step-by-step completion. Consequently, certain instances exhibit logical chain inconsistencies and self-contradictions. We preserve the logical inconsistencies to evaluate model robustness under real-world data conditions, while resolving self-contradictions through user communication and requirement validation to ensure the uniqueness and accuracy of ideal solutions.

3.5 User Input Collection

We systematically analyze model performance under identical scenarios (same SOP) by validating with users in business department and confirm that the selected scenarios and user inputs authentically reflected the pain points they encounter when using LLMs in real-world applications. This process yielded a final dataset of 397 samples.

3.6 Benchmark Quality Control

Dataset Quality Control: To ensure data integrity, we recruit five professional annotators holding bachelor's degrees with over three years of annotation experience. These annotators conduct cross-validation to identify and rectify lethal logical inconsistencies in SOPs and user prompts, ensuring each task yields a single definitive answer. Each task undergoes quality inspection by a minimum of

three annotators, requiring unanimous consensus before inclusion in the final dataset.

Scoring Quality Control: Scoring validation employs a cross-verification framework where each task is evaluated by at least three annotators who assess responses generated by the three models Claude-4-Sonnet (Anthropic, 2025), GPT-4.1 (OpenAI, 2025a) and Deepseek-R1 (DeepSeek-AI et al., 2025). Only tasks achieving cross-validation consistency across all evaluators are incorporated into the final dataset.

4 Task Formulation

In this section, we present the task format of SOP-Maze and outline the evaluation criteria.

4.1 Task Format

SOP-Maze incorporates a comprehensive evaluation framework comprising two core components: task composition and evaluation protocol. Given that open-ended tasks such as agent-chat may require LLM-based evaluation, which introduces uncontrollable factors to our benchmark, we employ a reference-based evaluation approach to mitigate this challenge.

In authentic AI customer service scenarios, SOPs provide tested models with detailed guidance, and users expect models to generate responses directly based on these SOPs. However, such responses are inherently difficult to evaluate, as models' creative variations preclude simple string matching. Evaluation typically relies on semantic similarity computation (unreliable) or LLM-as-a-judge approaches (costly and unstable). To balance the trade-off among evaluation cost, reliability, and scenario authenticity, we assign indices to all reference outputs in SOP guidance. For tasks requiring free-form responses, we mandate that models simultaneously output both the user-expected response and the corresponding reference index from the SOP guidance. This design preserves the natural response generation that users require while enabling the benchmark to perform reliable and efficient evaluation through simple index matching.

4.2 Evaluation Criteria

As outlined in the preceding section, the model is required to generate responses conforming to the JSON schema specified within the prompt. The JSON schema typically encompasses various output format constraints (e.g., boolean, string types) and incorporates optional parameters for certain keys. A baseline score of 0.2 points is awarded when the model output adheres to the prescribed format requirements. Complete alignment between model output and the reference answer yields the maximum score of 1.0 points.

Total	397		
Score	0.2	0	1
DeepSeek-V3.1-Thinking	268	7	132
Claude-Opus-4-Thinking	227	39	131
Doubao-Seed-1.6-ThinkingOn	243	27	127

Table 1: Performance of Top Three Models

5 Experiments

The results are presented in Table 2. The experimental configuration is detailed below. We comprehensively assess 18 LLMs, including 11 API-based models and 7 open-source models (Anthropic, 2025; DeepSeek-AI, 2024; OpenAI, 2025a; Bytedance, 2025; Google DeepMind, 2025; OpenAI, 2025b; Team et al., 2025; Yang et al., 2025). All models are set to default hyper-parameters as demonstrated on API configuration page or huggingface.

As mentioned in Section 4.2, models are evaluated using a three-tier scoring system:

$$S = \begin{cases} 1.0 & \text{correct response} \\ 0.2 & \text{valid format, incorrect response} \\ 0 & \text{invalid format} \end{cases}$$

This scoring framework ensures that models are evaluated on both procedural compliance (format adherence) and substantive performance (content accuracy), with greater weight assigned to correctness of responses.

6 Analysis

To investigate why models struggle on SOP-Maze, we conduct an in-depth analysis across all evalu-

[2] **HRS Tasks (H1-H10):** food combo business development, restaurant reservation customer service v2, check hotel room availability, customer evaluation follow up, food delivery customer service, promoting loan services, schedule in person appointment, restaurant reservation customer service v1, restaurant business development, bulk order clarification. **LRS Tasks (L1-L13):** core info extraction v1, risky content detect, flow line track, customer service rate, bulk order compare, intention recognition analysis, optimal traffic delivery schedule, app guide, user emotion analysis, core info extraction v2, complaint analysis, intention recognition function call, named entity classification. All values are multiplied by 100 and rounded to integers.

(a) Part A: HRS Task Performance (values multiplied by 100)

Model	Overall	H1	H2	H3	H4	H5	H6	H7	H8	H9	H10	HRS OA
DeepSeek-V3.1-Thinking	**46**	60	76	28	44	96	50	49	84	76	74	**64**
Claude-Opus-4-Thinking	**44**	50	66	18	42	92	48	65	84	84	51	**60**
Doubao-Seed-1.6-ThinkingOn	**43**	68	61	18	20	96	50	58	63	82	68	**58**
Claude-Sonnet-4-Thinking	**42**	52	57	26	44	84	74	46	71	92	55	**60**
o3-mini (high)	**40**	36	55	20	44	88	52	26	76	92	49	**54**
Claude-Opus-4	**38**	60	67	16	20	79	66	52	84	68	58	**57**
GPT-4.1	**37**	68	52	20	36	88	50	33	64	58	42	**51**
Doubao-Seed-1.6-ThinkingOff	**37**	58	57	18	20	88	58	42	63	74	52	**53**
Claude-Sonnet-4	**36**	76	47	28	36	79	74	51	57	72	46	**57**
Kimi-K2-Instruct	**36**	44	55	20	20	84	50	39	76	76	62	**53**
Gemini-2.5-Flash-Preview	**36**	36	56	18	20	92	60	38	80	82	26	**51**
GPT-4o-0513	**36**	52	63	20	36	88	68	39	72	66	39	**54**
GPT-4o-2024-08-06	**35**	36	52	28	36	84	60	33	68	66	42	**51**
DeepSeek-V3.1	**35**	52	68	20	20	80	50	36	72	68	33	**50**
Qwen3-14B-Thinking	**31**	36	47	20	20	56	26	25	68	74	31	**40**
Qwen3-32B-Thinking	**30**	44	56	12	20	84	34	26	56	58	30	**42**
Qwen3-14B	**29**	36	55	20	20	72	26	25	64	66	21	**40**
Qwen3-32B	**27**	36	55	14	20	68	28	22	60	28	26	**36**

(b) Part B: LRS Task Performance (values multiplied by 100)

Model	L1	L2	L3	L4	L5	L6	L7	L8	L9	L10	L11	L12	L13	LRS OA
DeepSeek-V3.1-Thinking	19	20	24	20	52	24	24	36	20	30	68	52	25	**32**
Claude-Opus-4-Thinking	19	20	32	20	48	28	19	29	23	28	70	43	31	**32**
Doubao-Seed-1.6-ThinkingOn	20	3	23	20	47	32	22	43	18	31	76	52	19	**31**
Claude-Sonnet-4-Thinking	17	20	24	20	42	35	11	28	18	19	44	54	34	**28**
o3-mini (high)	18	20	24	20	40	32	22	31	17	19	48	52	29	**29**
Claude-Opus-4	9	12	24	18	15	32	0	24	19	33	56	42	22	**24**
GPT-4.1	20	20	20	20	33	24	20	23	20	28	40	44	20	**26**
Doubao-Seed-1.6-ThinkingOff	19	0	17	18	30	32	20	16	20	28	52	44	18	**24**
Claude-Sonnet-4	13	15	22	18	27	15	1	31	19	17	30	41	24	**21**
Kimi-K2-Instruct	20	20	20	20	8	36	20	17	19	22	48	36	25	**24**
Gemini-2.5-Flash-Preview	17	20	24	18	23	28	20	19	15	19	56	40	19	**24**
GPT-4o-0513	18	20	17	20	15	28	2	15	13	19	47	43	19	**21**
GPT-4o-2024-08-06	19	20	19	20	18	20	19	17	20	24	40	48	19	**23**
DeepSeek-V3.1	19	15	20	20	20	24	20	10	20	30	40	44	25	**24**
Qwen3-14B-Thinking	8	20	24	20	27	32	22	9	19	20	36	44	29	**24**
Qwen3-32B-Thinking	12	20	19	20	20	24	2	16	16	22	40	36	20	**21**
Qwen3-14B	18	20	20	18	17	24	20	11	19	20	31	20	20	**20**
Qwen3-32B	15	20	20	20	17	20	20	15	17	20	28	28	20	**20**

Table 2: Model Performance on HRS and LRS Tasks (Commercial & open-source models tested w/ default settings)[2]

Score	0.2		
Error Type	Route Blindness	Conversational Fragility	Calculation Error
DeepSeek-V3.1-Thinking	177	166	60
Claude-Opus-4-Thinking	151	149	60
Doubao-Seed-1.6-ThinkingOn	164	145	63

Table 3: Error Breakdown of Top Three Models. One model answer may cover more than one error types.

ated models. Our analysis reveals three main error types. To illustrate and quantify these issues, we present a detailed case study of the top three models: DeepSeek-V3.1-Thinking, Claude-Opus-4-Thinking, and Doubao-Seed-1.6-ThinkingOn. The frequency of each error type occurs for these three models is detailed in Table 1 and Table 3.

In the following subsections, we explain the models' suboptimal performance from the perspective of these three error categories.

6.1 Fail to Grasp the Full Context of SOPs

When reasoning over SOPs, models often fail to strictly adhere to the given procedure; they may deviate onto incorrect branches or jump to non-sequential steps. We refer to this common failure pattern as "route blindness." Specifically, it presents differently in LRS and HRS.

6.1.1 Route Blindness in LRS

In LRS, the underlying graph structure is relatively shallow, with at most three levels from the root to the leaves. However at each level, a parent node can have more than 10 child nodes, resulting in a wide branching structure. On average, an LRS scenario approximately contains 5 parent nodes and 58 leaf nodes in total. The substantial number of possible routes greatly increases the complexity of the reasoning process for models. Consequently, models cannot handle such extensive branching complexity in parallel, resulting in frequent errors.

6.1.2 Route Blindness in HRS

In HRS, the underlying graph structure is characterized by greater depth. We observe a distinct

phenomenon: models frequently bypass intermediate nodes instead of adhering to the procedure. To explore the causes, we investigate on reasoning models, and their CoT reveals that this is not a random occurrence. They often misinterpret the context as satisfying the preconditions for subsequent steps, prompting them to move forward without fulfilling necessary requirements. This misjudgment also demonstrates that even reasoning models lack effective self-correction.

6.1.3 Other Common Failure

Furthermore, both in LRS and HRS, we find that models have difficulty handling progressive or hierarchical constraints. For instance there is an instruction saying "*If user's query involves both Step 3 and Step 9, prioritize Step 9.*" In such cases, models often default to the general rule (i.e. Step 3) and overlook the more specific rule (i.e. Step 9). We also note that models struggle to distinguish between contexts that are syntactically similar yet semantically distinct.

6.2 Fail to Grasp the Nuances of Daily Conversation

The "input" part of SOP-Maze is mainly composed of conversations. These conversations mirror the complexity of daily dialogues, which are characterized by three key attributes: strong contextual reliance, disfluent phrasing, and subtle intent. These nuances reveal a core weakness in models, which we term "conversational fragility".

6.2.1 Strong Contextual Reliance

In real-world conversations, user intentions are not static but evolve dynamically over multiple turns. However, we find that models often fail at this point. They tend to lock on the initial instruction, and ignore later updates. A prime example is the "Intention Recognition Analysis" scenario. User first threatens, "*If you don't do anything about this, I'm going to file a complaint,*" but ends with "*I'll let it slide this time.*" This is a clear reversal of intent. Yet, models fail to recognize that the complaint has been withdrawn.

6.2.2 Disfluent Phrasing

The unpredictable and messy nature of daily conversation, particularly its irregular turn-taking, poses a major challenge for models. These disfluent inputs cause models to struggle with basic comprehension. The following example illustrates this vulnerability. In this scenario, the user is driving, and the car's navigation prompts are mixed with user's speech. "*Assistant: Hello, is this Ms. Zhang? User: Hello, make a U-turn... uh, yes. [...Omitting middle rounds...] Assistant: So, are you interested? User: Current speed is 126, yes, you are speeding.*"

While the underlined part is the user's actual speech, models fail to isolate user's simple but important affirmation, "yes". Instead, they incorrectly interpret the entire utterance as navigation instructions, leading to a misguided follow-up response.

6.2.3 Subtle Intent

Another noteworthy observation is that models are unable to comprehend non-literal language, such as sarcasm and irony. They process only the surface-level meaning of the user's statements, missing the true—often contradictory—intent behind the word. This limitation is evident in the "Food Combo Business Development" scenario, where users occasionally respond to the assistant's sales tactics with sarcasm. For example: "*Haha, yeah, right. Sounds amazing. Looks like that company is back at it again, trying to get their hands on our paychecks.*" Although this is a sarcastic and mocking response, models fail to detect the irony and instead interpret it as positive feedback.

6.3 Fail to Perform on Calculations

Another interesting finding is that real-world tasks involve calculating and counting, areas where models tend to perform poorly. Their difficulties are observed in time-related calculation and mathematical arithmetic. To further illustrate these challenges, we present two representative scenarios for each aspect below. In the "Customer Service Rate" scenario, we assess models' ability to calculate the reply latent—a metric defined as the period during which a customer service agent remains silent or unresponsive to the user. This metric affects the service agent's rating. Even when timestamps are explicitly stated within the conversation (e.g. *User (2025-02-09 19:01:57): [...] Assistant (2025-02-09 19:02:00): [...]*), we observe that models still make significant errors in the calculation. In the "Optimal Traffic Delivery Schedule" scenario, the goal is to identify the optimal date range for traffic allocation. One of the rules states: "*Select dates where the number of stores is less than or equal to the median number of stores within that date range.*" While models are often able to choose the correct date range, they frequently miscalculate the median.

6.4 Gap Between Reasoning models and Non-reasoning models

Reasoning models significantly outperform non-reasoning models in most scenarios within SOP-Maze (Figure 4), and this performance delta is especially noticeable in LRS scenarios. LRS scenarios require models to handle many different reasoning branches. Non-reasoning models struggle with this complexity. In contrast, reasoning models are better equipped to manage these tasks and can successfully navigate the majority of reasoning paths. In HRS scenarios, non-reasoning models exhibit a more severe form of "hopping" or logical incoherence. They might correctly identify the user's intent, but then unexpectedly inject an unrelated step in their answer. Sometimes, non-reasoning models also list repetitive or multiple possible candidate steps. Both behaviors indicate that they lack the ability to reason and organize information effectively. This helps explain why reasoning models consistently outperform them.

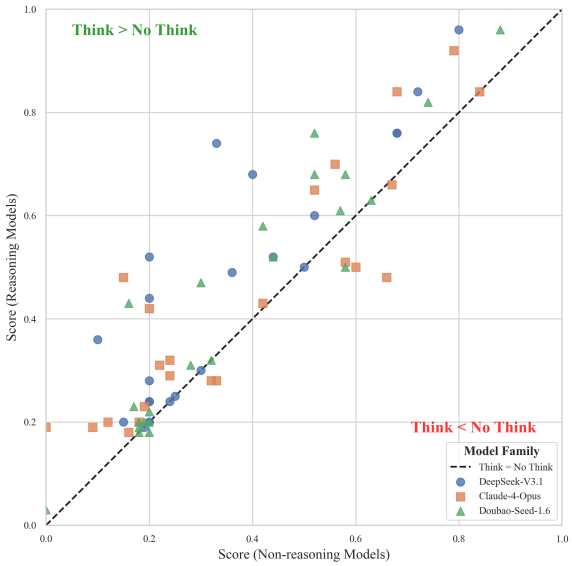

Figure 4: Reasoning models outperform non-reasoning models in most scenarios within SOP-Maze.

7 Ablation Studies

To quantitatively validate our previous analysis, we conduct a series of targeted ablation studies. Each study focuses on one of previous three key factors. These experiments are designed to break down scenario complexity and isolate how each component contributes to the model's performance degradation. For each of the three identified failure modes, we select its most representative scenario. Within each scenario, we apply a three-stage ablation process, progressively simplifying the input provided to the model. The three stages are defined as follows:

Stage 1 (S1): Full Context, Simplified Query
Context: The complete SOP is provided, including potentially irrelevant or distracting text.
Query: The query is simplified to only include the core constraint related to the failure mode. All other constraints are removed.

Stage 2 (S2): Partial Context, Simplified Query
Context: The SOP is simplified to only include the content related to the core constraint. All other parts are removed.
Query: The query remains identical to Stage 1.

Stage 3 (S3): No Context, Direct Query
Context: The SOP framework is entirely removed.
Query: A direct question to the core constraint.

We choose top three models, DeepSeek-V3.1-Thinking, Claude-Opus-4-Thinking, and Doubao-Seed-1.6-ThinkingOn. Their three-stage comparison helps to identify the sources of failure and measure the severity of each issue discussed earlier.

7.1 Influence of Route Blindness

We deliberately choose "pure" scenario—without conversational ambiguity or calculation. Despite the failure mode of LRS and HRS diverge in Route Blindness, the ablation study reveals a convergence: a sharp increase from the original task to Stage 1, after which performance quickly saturated (See Figure 5). This trend strongly indicates that the main obstacle lies not in individual rules but in their combinatorial complexity, highlighting the substantial challenge SOP-Maze poses even for state-of-the-art models.

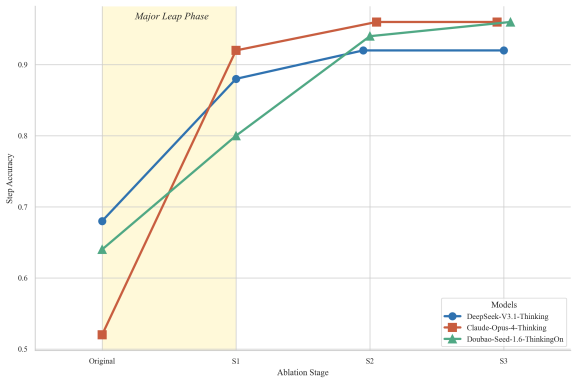

Figure 5: Experiment of Route Blindness Ablation on "Bulk Order Clarification"

However, a closer look at the remaining failures illustrates that not all individual constraints

are equally easy. The underlying reasons fundamentally differ between LRS and HRS. For LRS, models exhibit a weakness in handling fine-grained instructions. This is evident in the "Core Info Extraction" scenario. They either miss words when extracting tourist spots, or misidentify the number of afternoon tea sets. For HRS, the failure stems once the models become confused at two paths. They get "stuck" and are unable to recover, inevitably committing to the incorrect branch. We also uncover a side effect at Stage 2 and Stage 3: while attempting to resolve the initial ambiguity, models will sometimes introduce other entirely new, unpredictable errors.

7.2 Impact of Conversational Fragility

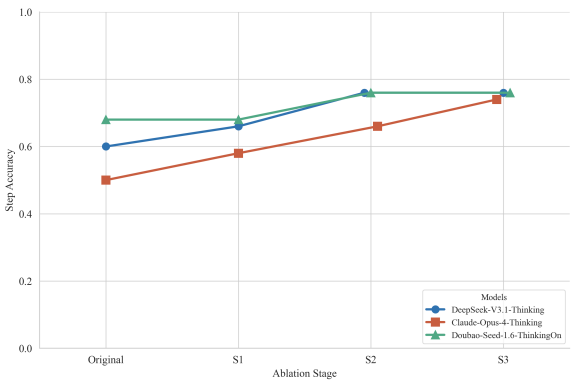

Figure 6: Experiment of Conversational Fragility Ablation on "Food Combo Business Development"

Figure 6 illustrates a distinct pattern from the Conversational Fragility ablation: models almost reach a performance ceiling even in the simplest form. We select the "Food Combo Business Development" scenario, which requires the detection of sarcasm and emotional cues. Unlike the dramatic improvements observed in previous experiments, all three models showed slight improvements, with accuracy stabilizing around 70%. These limited gains suggest that simplifying queries and cleaning up context provide only marginal benefits. Furthermore, the plateau in performance from Stage 2 to Stage 3 confirms that comprehension itself is the primary bottleneck. This core deficit underscores a critical barrier to the models' effective performance in the real-world conversation, where such nuanced, non-literal language is the norm.

7.3 Effect of Calculations

In the "Customer Service Rate" scenario, models are required to compute reply latency within a conversation. Surprisingly, we observe notably poor performance on the original task: Claude-4-Opus achieving only 40% accuracy and Seed-1.6-ThinkingOn at 30%, as shown in Figure 7. However, both models improve substantially in Stage 1, gaining over 20 percentage points. This upward trend continues through Stage 2, culminating in 90% accuracy for all tested models in Stage 3. This demonstrate that models possess the capability to perform accurate calculations, yet this ability is significantly hindered by contextual complexity of the full SOP-Maze task.

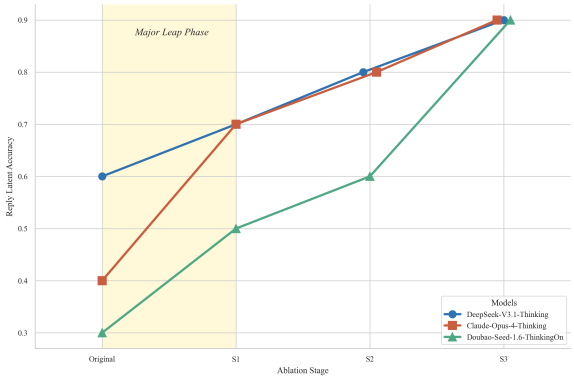

Figure 7: Experiment of Calculation Ablation on "Customer Service Rate"

A closer analysis of the remaining 10% of failures in Stage 3 reveal a persistent failure mode: models can identify the correct segment and relevant timestamps, but ultimately select incorrect ones. When provided with an explicit follow-up prompt—*Do you think this segment meets the reply latency?*—all three models consistently produce the correct answer.

8 Conclusion

We introduce SOP-Maze, the first benchmark evaluating the practical instruction-following capability of LLMs in business SOP scenarios, comprising 397 tasks across 23 real-world scenarios categorized into LRS and HRS. Evaluation of 18 state-of-the-art models reveals significant performance gaps, we identify three core failure modes: route blindness, conversational fragility, and calculation errors, stemming not from isolated skill deficits but from the complex interactions and noise inherent in real business SOPs. Consequently, we provide a rigorous analysis of the reasons for failure and offer new insights to develop more reliable instruction-following systems in future LLM research.

9 Limitations

Due to our stringent requirements for data quality, only 397 data samples were ultimately included in our final dataset. The relatively limited sample size may introduce variability in model evaluation scores. While the data presents considerable difficulty, our benchmark is designed with future advancements in mind, and we anticipate that superior models will emerge to tackle this benchmark in the future.

SOP-Maze inherently evaluates comprehensive LLM capabilities, making ablation studies particularly challenging. This difficulty arises because in real-world business scenarios, the problems and pain points encountered by models are highly abstract and complex, rendering it difficult to isolate these challenges into discrete capabilities for targeted optimization.

The allocation of 0.2 points for formatting in the metric system lacks a specific theoretical justification and serves primarily to introduce a quantitative component to the overall metric scoring framework. Alternative scoring methodologies may potentially offer more effective approaches to point distribution in this context.

References

Anthropic. 2025. Claude sonnet 4. Visited date: 2025-09-15.

Alessandro Berti, Hani Kourani, and Wil M. P. van der Aalst. 2024. PM-LLM-Benchmark: Evaluating Large Language Models on Process Mining Tasks. In *International Conference on Process Mining*, pages 610–623, Cham. Springer Nature Switzerland.

Bytedance. 2025. Doubao-ai. Visited date: 2025-09-15.

DeepSeek-AI. 2024. Deepseek-v3 technical report. *Preprint*, arXiv:2412.19437.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, and 181 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

Lingxiao Diao, Xinyue Xu, Wanxuan Sun, Cheng Yang, and Zhuosheng Zhang. 2025. Guidebench: Benchmarking domain-oriented guideline following for llm agents. *arXiv preprint arXiv:2505.11368*.

Dirk Fahland, Fabian Fournier, Lior Limonad, Inbal Skarbovsky, and Alexander J. Swevels. 2024. How well can large language models explain business processes? *arXiv preprint*.

Fabiana Fournier, Lior Limonad, and Inna Skarbovsky. 2024. Towards a benchmark for causal business process reasoning with llms. *Preprint*, arXiv:2406.05506.

Google DeepMind. 2025. Gemini flash. Visited date: 2025-09-15.

Michael Grohs, Luka Abb, Nourhan Elsayed, and Jana-Rebecca Rehse. 2023. Large language models can accomplish business process management tasks. *Preprint*, arXiv:2307.09923.

Qianyu He, Jie Zeng, Wenhao Huang, Lina Chen, Jin Xiao, Qianxi He, Xunzhe Zhou, Jiaqing Liang, and Yanghua Xiao. 2024a. Can large language models understand real-world complex instructions? In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 18188–18196.

Yun He, Di Jin, Chaoqi Wang, Chloe Bi, Karishma Mandyam, Hejia Zhang, Chen Zhu, Ning Li, Tengyu Xu, Hongjiang Lv, Shruti Bhosale, Chenguang Zhu, Karthik Abinav Sankararaman, Eryk Helenowski, Melanie Kambadur, Aditya Tayade, Hao Ma, Han Fang, and Sinong Wang. 2024b. Multi-if: Benchmarking llms on multi-turn and multilingual instructions following. *Preprint*, arXiv:2410.15553.

Humam Kourani, Alessandro Berti, Jasmin Hennrich, Wolfgang Kratsch, Robin Weidlich, Chiao-Yun Li, Ahmad Arslan, Daniel Schuster, and Wil M. P. van der Aalst. 2024a. Leveraging large language models for enhanced process model comprehension. *Preprint*, arXiv:2408.08892.

Humam Kourani, Alessandro Berti, Daniel Schuster, and Wil M. P. van der Aalst. 2024b. Evaluating large language models on business process modeling: Framework, benchmark, and self-improvement analysis. *Preprint*, arXiv:2412.00023.

Pranjal Kumar. 2024. Large language models (llms): survey, technical frameworks, and future challenges. *Artificial Intelligence Review*, 57(10):260.

Bo Li, Yuanhan Zhang, Liangyu Chen, Jinghao Wang, Fanyi Pu, Joshua Adrian Cahyono, Jingkang Yang, Chunyuan Li, and Ziwei Liu. 2025a. Otter: A multi-modal model with in-context instruction tuning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Jinnan Li, Jinzhe Li, Yue Wang, Yi Chang, and Yuan Wu. 2025b. Structflowbench: A structured flow benchmark for multi-turn instruction following. *Preprint*, arXiv:2502.14494.

Zekun Li, Shinda Huang, Jiangtian Wang, Nathan Zhang, Antonis Antoniades, Wenyue Hua, Kaijie Zhu, Sirui Zeng, Chi Wang, William Yang Wang, and Xifeng Yan. 2025c. Sopbench: Evaluating language agents at following standard operating procedures and constraints. *Preprint*, arXiv:2503.08669.

Subhrangshu Nandi, Arghya Datta, Nikhil Vichare, Indranil Bhattacharya, Huzefa Raja, Jing Xu, Shayan Ray, Giuseppe Carenini, Abhi Srivastava, Aaron Chan, Man Ho Woo, Amar Kandola, Brandon Theresa, and Francesco Carbone. 2025. Sop-bench: Complex industrial sops for evaluating llm agents. *Preprint*, arXiv:2506.08119.

OpenAI. 2025a. Gpt-4.1. Visited date: 2025-09-15.

OpenAI. 2025b. o3-mini. Visited date: 2025-09-15.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, and 1 others. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Libo Qin, Qiguang Chen, Xiachong Feng, Yang Wu, Yongheng Zhang, Yinghui Li, Min Li, Wanxiang Che, and Philip S Yu. 2024a. Large language models meet nlp: A survey. *arXiv preprint arXiv:2405.12819*.

Yiwei Qin, Kaiqiang Song, Yebowen Hu, Wenlin Yao, Sangwoo Cho, Xiaoyang Wang, Xuansheng Wu, Fei Liu, Pengfei Liu, and Dong Yu. 2024b. Infobench: Evaluating instruction following ability in large language models. *Preprint*, arXiv:2401.03601.

Adrian Rebmann, Fabian David Schmidt, Goran Glavaš, and Han van Der Aa. 2024. Evaluating the ability of llms to solve semantics-aware process mining tasks. In *2024 6th International Conference on Process Mining (ICPM)*, pages 9–16.

Yuchong Sun, Che Liu, Kun Zhou, Jinwen Huang, Ruihua Song, Wayne Xin Zhao, Fuzheng Zhang, Di Zhang, and Kun Gai. 2024. Parrot: Enhancing multi-turn instruction following for large language models. *Preprint*, arXiv:2310.07301.

Xiangru Tang, Yiming Zong, Jason Phang, Yilun Zhao, Wangchunshu Zhou, Arman Cohan, and Mark Gerstein. 2024. Struc-bench: Are large language models good at generating complex structured tabular data? In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 12–34.

Kimi Team, Yifan Bai, Yiping Bao, Guanduo Chen, Jiahao Chen, Ningxin Chen, Ruijue Chen, Yanru Chen, Yuankun Chen, Yutian Chen, Zhuofu Chen, Jialei Cui, Hao Ding, Mengnan Dong, Angang Du, Chenzhuang Du, Dikang Du, Yulun Du, Yu Fan, and 150 others. 2025. Kimi k2: Open agentic intelligence. *Preprint*, arXiv:2507.20534.

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

Zichong Wang, Zhibo Chu, Thang Viet Doan, Shiwen Ni, Min Yang, and Wenbin Zhang. 2025. History, development, and principles of large language models: an introductory survey. *AI and Ethics*, 5(3):1955–1971.

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 2021. Finetuned language models are zero-shot learners. *arXiv preprint*.

Bosi Wen, Pei Ke, Xiaotao Gu, Lindong Wu, Hao Huang, Jinfeng Zhou, Wenchuang Li, Binxin Hu, Wendy Gao, Jiaxing Xu, and 1 others. 2024. Benchmarking complex instruction-following with multiple constraints composition. *Advances in Neural Information Processing Systems*, 37:137610–137645.

Congying Xia, Chen Xing, Jiangshu Du, Xinyi Yang, Yihao Feng, Ran Xu, Wenpeng Yin, and Caiming Xiong. 2024. Fofo: A benchmark to evaluate llms' format-following capability. *arXiv preprint arXiv:2402.18667*.

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, and 41 others. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Shunyu Yao, Howard Chen, Austin W. Hanjie, Runzhe Yang, and Karthik Narasimhan. 2023. Collie: Systematic construction of constrained text generation tasks. *Preprint*, arXiv:2307.08689.

Junjie Zhang, Ruobing Xie, Yupeng Hou, Xin Zhao, Leyu Lin, and Ji-Rong Wen. 2025. Recommendation as instruction following: A large language model empowered recommendation approach. *ACM Transactions on Information Systems*, 43(5):1–37.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. 2023a. Instruction-following evaluation for large language models. *Preprint*, arXiv:2311.07911.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. 2023b. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*.

A Supplementary Materials

A.1 Ethics Statement

Our proposed method and algorithm do not incorporate any adversarial attack mechanisms and pose no threat to human safety. All experiments were conducted exclusively in simulated environments, thereby avoiding any ethical or fairness-related concerns.

A.2 The Use of LLM

We leverage a large language model as a general-purpose writing assistant to refine our paper, assisting with grammar correction, phrasing, tone adjustment, punctuation, and maintaining stylistic coherence. Additionally, we employ the LLM's auto-completion feature to expedite the development of our evaluation pipeline.

A.3 Reproducibility Statement

The source code associated with this work is publicly accessible at https://anonymous.4open.science/r/SOP-Maze-077B . The complete implementation details of our approach are in Section 5

A.4 Artifacts Use

This paper complies with all applicable licenses, whether open-source or commercial, for the large language models utilized. All artifacts employed in our study are used strictly in accordance with their intended purposes.

A.5 Data Source

Our dataset contains no personally identifiable information or offensive material. Professional annotators thoroughly reviewed the entire dataset and confirmed the absence of such content. The data were collected and curated by our internal business unit, which has explicitly consented to its open-source release and academic dissemination, adhering fully to the ACL Guidelines for Ethics Reviewing.

A.6 Recruitment and Payment

All annotators were engaged under formal contracts, and their compensation, handled confidentially, has been fully paid. The annotators are professional, Chinese individuals, each holding at least a bachelor's degree.