Talk, Evaluate, Diagnose: User-aware Agent Evaluation With Automated Error Analysis

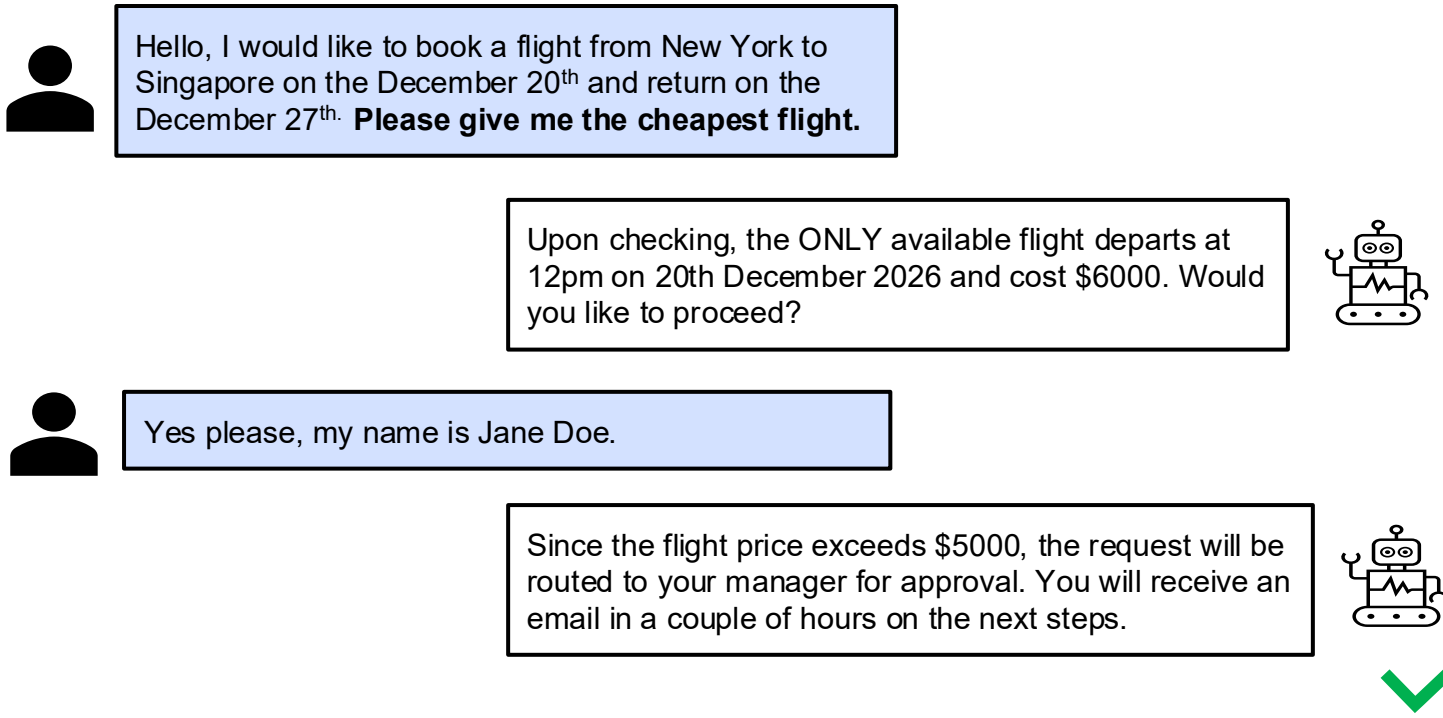
Penny Chong[1], Harshavardhan Abichandani[1], Jiyuan Shen[1], Atin Ghosh[1], Min Pyae Moe[1], Yifan Mai[2], Daniel Dahlmeier[1]

SAP[1], Stanford University[2]

1. Background

- Agent evaluation remains challenging due to the heterogenous domain they operate in. Existing works do not systematically account for the type of user interaction when evaluating agent performance.
- We argue that effective agent evaluation goes beyond correctness alone, incorporating conversation quality, efficiency and systematic diagnosis of agent errors.

Expert user with agent

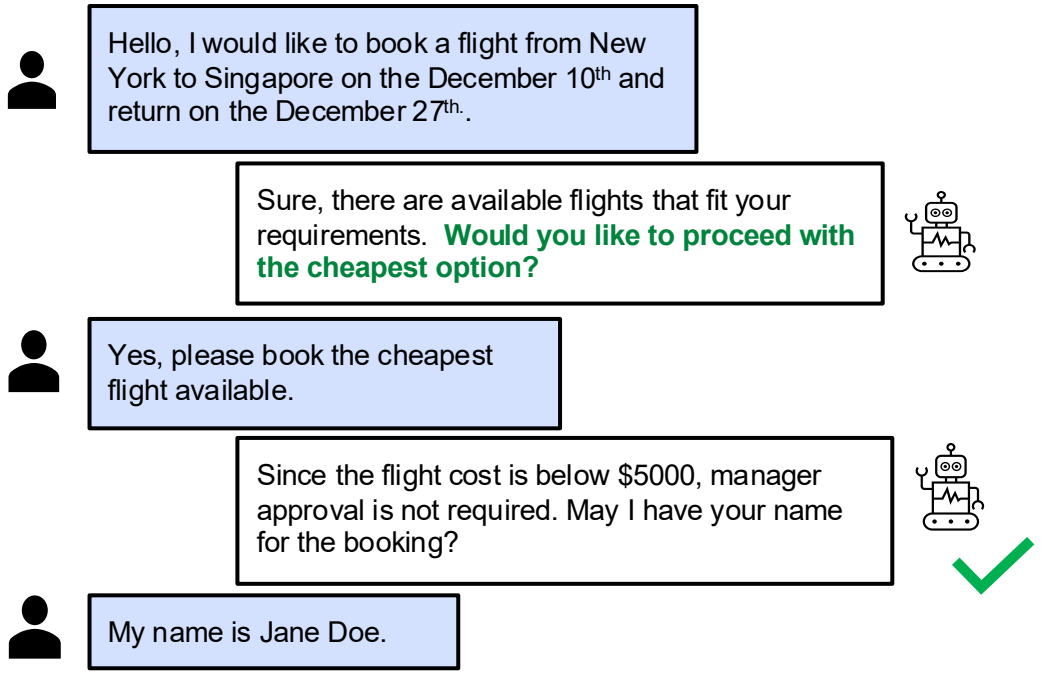


- Given clear user instructions as simulated by an expert user of the system, the agent can often complete the task.

1. Background

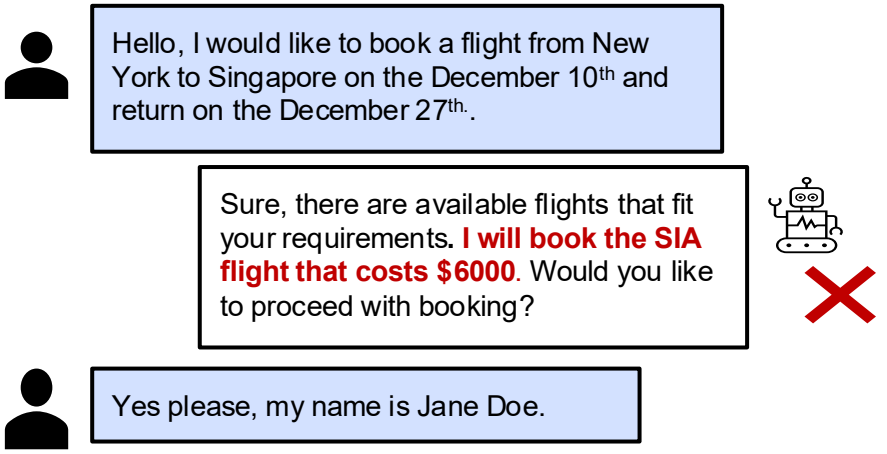
- When user instructions are *unclear* and *incomplete*, the task becomes more difficult, making testing across different user expertise essential.

Non-expert user with good agent



➢ A good agent will proactively ask user for her preference before proceeding to book flights to avoid triggering unnecessary approval workflow.

Non-expert user with poor agent



➢ Since the user forgot to specify a preference for the cheapest flight, a poorly designed agent proceeds without clarification and unnecessarily triggers the approval workflow.

2. Key Contributions

i. Propose an agent evaluation framework built on *reusable, generic* expert and non-expert user persona templates that assess the impact of users' role on agent performance.

ii. Introduce a benchmark by adapting existing datasets to grading notes—natural language checklists of subgoals.

iii. Introduce new metrics to accompany the user-aware agent evaluation setup to capture agent's progress with respect to the number of conversational turns.

iv. Propose an automated error analysis tool that discovers common errors and provides actionable feedback for agent improvement.

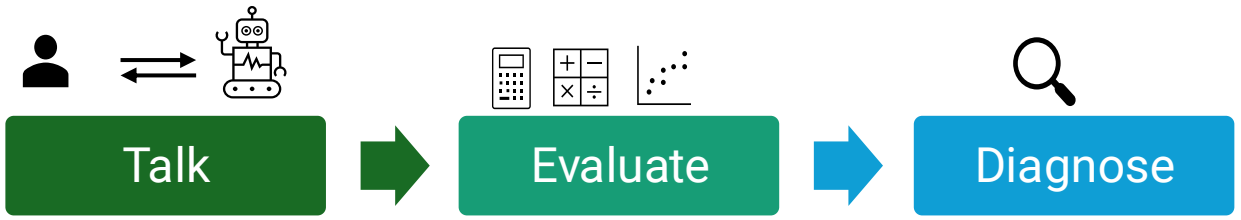


Figure 1: The *Talk*, *Evaluate*, *Diagnose* (TED) framework leverages *generic* user persona templates, introduces new metrics to track agent's progress, and an automated error analysis tool to provide actionable feedback for agent improvement.

3. Methodology

The Talking Stage

- This stage introduces *reusable*, *generic* user persona templates to simulate user interaction to assess impact of user's role on agent performance.

$$u = f(p, i), \quad \text{where} \quad p \in P, i \in I, u \in U.$$

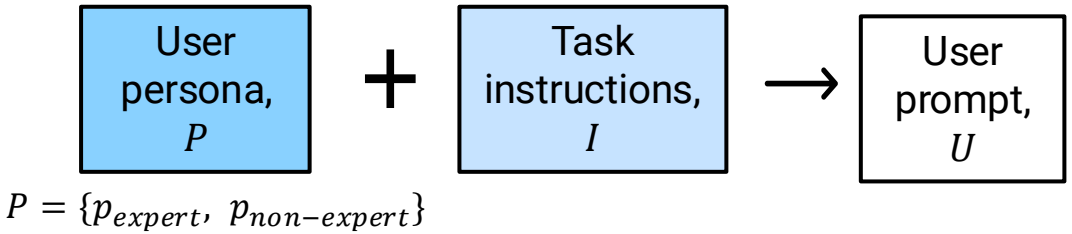


Figure 2: For each agent and task instruction $i \in I$, the user persona prompt $p \in P$ is varied to generate the user prompt $u_{expert} \in U$ and $u_{non-expert} \in U$.

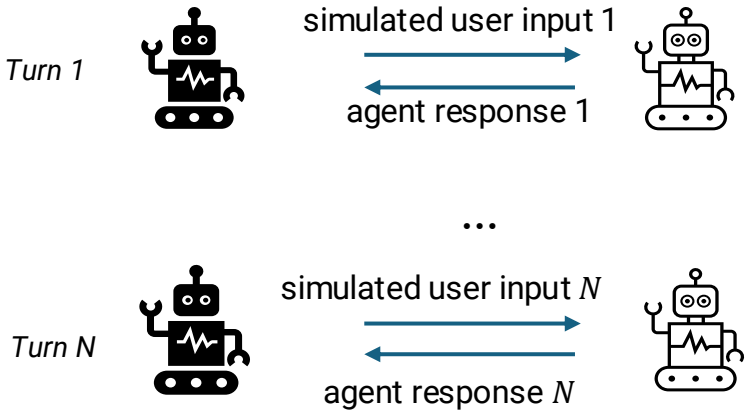


Figure 3: Dynamic user conversation with the agent is simulated using the full user prompt $u \in U$.

3. Methodology

The Evaluation Stage

- We follow Chang et al. (2024) to use a fine-grained progress metric to measure agent's best performance in the $n = k$ trials to differentiate between tasks that are near completion and those that are far from completion:

$$MaxProgressRate@k = \mathbb{E}_{(i,G_i)\sim P_D}\left[\max\left\{progress\left(i, G_i, \tau_i^l\right) \middle| \, l = 1, \ldots, k\right\}\right].$$

- We define the progress for a given task i as the *proportion of* grading notes $G_i = \{g_{i,1}, \ldots, g_{i,|G_i|}\}$, that are completed by the agent.

For task sample (i, G_i):

- *Agent should call get current location to retrieve the user's location* ✓
- *Agent should …* ✗
- *Agent should …* ✓

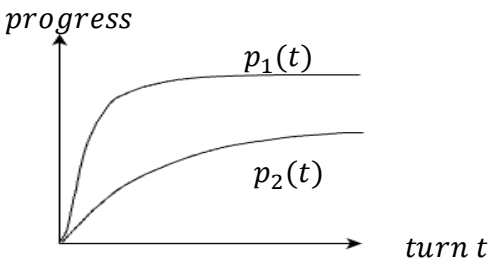
LLM-as-a-judge

$progress\left(i, G_i, \tau_i^l\right) = \frac{2}{3}$

Figure 4: Progress via LLM-as-a-judge and grading notes.

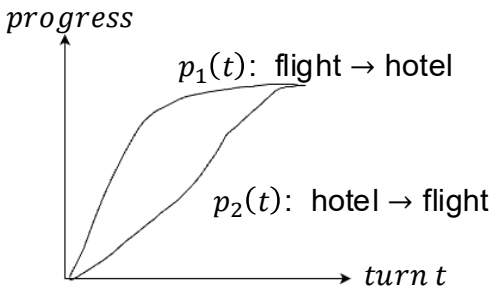
- Measuring final-turn progress is **insufficient**, especially when user behaviour affects the agent; we must assess both overall progress and turn-level efficiency across the following scenarios:

a) Early progress matter:



$$AUC_1 = \int_0^{\mathrm{T}} p_1(t)\, dt > \int_0^{\mathrm{T}} p_2(t)\, dt = AUC_2.$$

b) Early progress does not matter:



Progress-per-turn (PPT) = $\frac{1}{T}\sum_{t=0}^{T-1} p(t+1) - p(t) = \frac{p(T)}{T}$.

3. Methodology

The Diagnosis Stage

- Analyze the agent and judge consistencies by plotting the agent's expected progress and variance for each of the k trials.
- Introduce a two-step automated error discovery approach that automatically identifies common errors of the agent based on judge and agent inconsistencies.

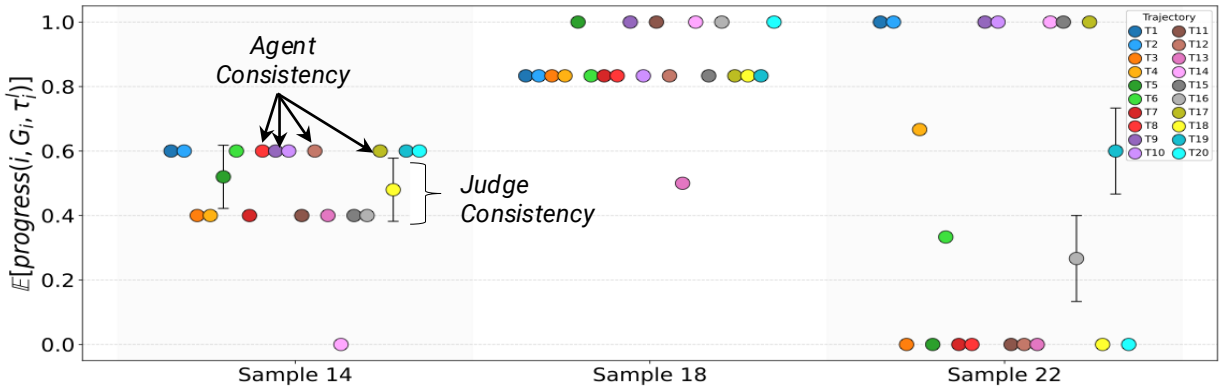


Figure 5: The $\mathbb{E}[progress(i, G_i, \tau_i^l)]$ *(dot)* and $Var[progress(i, G_i, \tau_i^l)]$ *(error bar)* values on the τ^2-bench gpt-4o-mini agent using non-expert user proxy gpt-4.1. Each dot is an agent trajectory from a single trial. Differences in the $\mathbb{E}[progress(i, G_i, \tau_i^l)]$ values indicate that the agent is inconsistent, while non-zero $Var[progress(i, G_i, \tau_i^l)]$ values indicate that the LLM-as-a-judge is inconsistent.

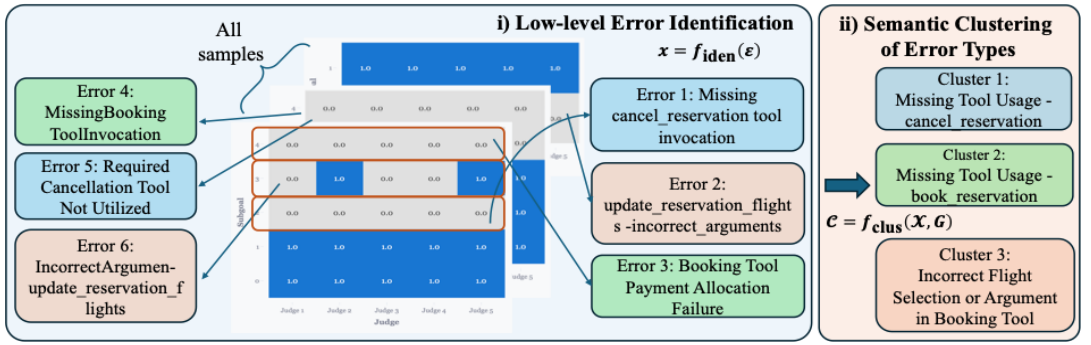


Figure 6: Two-step automated error discovery approach. Identical error colors indicate that similar low-level errors are clustered into the same high-level category.

4. Results and Discussion

Table 1: Overall performance of different agent models on τ^2-bench airline and ToolSandbox, using gpt-4.1 as user proxy and LLM-as-a-judge. Results are displayed with scores for Expert Persona | Non-expert Persona. For metrics with $@k$, the number of trials is n = k = 20 for τ^2 bench and n= k= 8 for ToolSandbox. $MaxProgressRate@k$ is abbreviated as $MaxProg@k$.

			Ours	Ours	
Agent Model	$MeanProg@k$	$MaxProg@k$	$MaxAUC@k$	$MaxPPT@k$	$pass@k$
τ^2-bench Airline Domain (Easy)					
gpt-4.1	**0.95** | 0.82	1.00 | 1.00	**0.99** | 0.81	**0.80** | 0.50	1.00 | 1.00
gpt-4o	0.79 | 0.86	1.00 | 1.00	0.96 | 0.86	0.70 | 0.53	1.00 | 1.00
gpt-4o-mini	0.70 | 0.61	0.90 | 0.90	0.85 | 0.73	0.60 | 0.37	0.80 | 0.80
gpt-5	0.92 | **0.92**	1.00 | 1.00	0.97 | **0.88**	0.67 | **0.54**	1.00 | 1.00
mistral-nemo	0.87 | 0.49	1.00 | 0.80	0.97 | 0.67	0.67 | 0.48	1.00 | 0.60
mistral-large	0.65 | 0.53	1.00 | 1.00	0.96 | 0.79	0.60 | 0.42	1.00 | 1.00
ToolSandbox Dataset					
gpt-4.1	0.91 | 0.87	0.98 | 0.97	0.96 | 0.92	0.84 | 0.73	0.92 | 0.92
gpt-4o	**0.95** | **0.94**	**0.99** | **1.00**	**0.98** | **0.96**	**0.94** | **0.81**	**0.95** | **0.97**
gpt-4o-mini	0.91 | 0.85	0.95 | 0.93	0.94 | 0.90	0.89 | 0.77	0.89 | 0.84
gpt-5	0.78 | 0.78	0.97 | 0.91	0.95 | 0.84	0.83 | 0.66	**0.95** | 0.84
mistral-nemo	0.72 | 0.71	0.92 | 0.96	0.88 | 0.87	0.76 | 0.65	0.84 | 0.92
mistral-large	0.82 | 0.79	0.94 | 0.95	0.93 | 0.91	0.87 | 0.75	0.89 | 0.89

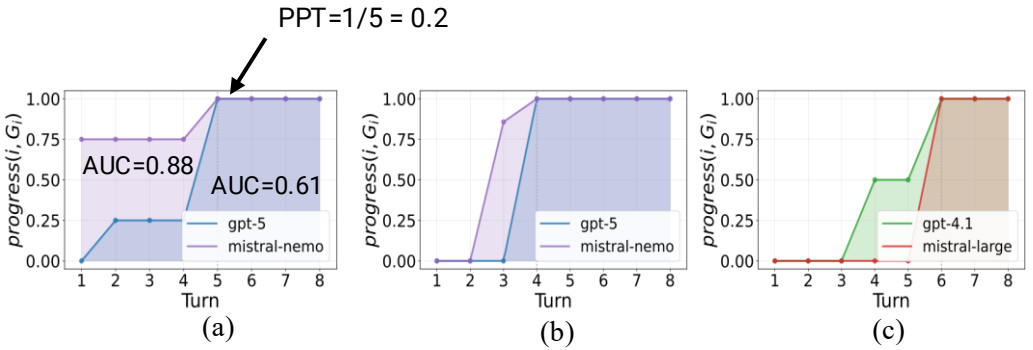


Figure 7: Progress curves for selected ToolSandbox samples.
(a) *search_reminder_with_recency_upcoming*: mistral-nemo (non-expert, purple; AUC=0.88, PPT=0.20) vs. gpt-5 (non-expert, blue; AUC=0.61, PPT=0.20).
(b) *find_current_city_low_battery_mode*: mistral-nemo (expert, purple; AUC=0.77) vs. gpt-5 (non-expert, blue; AUC=0.64).
(c) *add_reminder_content_and_date_and_time*: gpt-4.1 (non-expert, green; AUC=0.50) vs. mistral-large (non-expert, red; AUC=0.34).

4. Results and Discussion

LLM-summarized error category	Sample 14
1. Missing Communication - Amount/Payment/Breakdown	Trace 18
2. Missing Tool Usage: cancel_reservation	Trace 6, 18
3. Incorrect Output Communication - Amount/Value/Payment	Trace 6
4. Missing Tool Usage - book_reservation	Trace 18

Figure 8: Examples of the identified TED errors on the τ^2-bench airline *sample_14*.

Table 2: Agent improvement results using gpt-4.1 as user proxy and LLM-as-a-judge. Results are displayed for Expert Persona | Non-expert Persona. The improvement strategies that are used are *i) Errors Insert (EI):* add the list of TED errors into the agent instruction without manual refinement; *ii) Human Notes (HN):* insert manually refined TED error notes into the agent instruction.

Agent Model	$MeanProg@k$	$MaxProg@k$	$MaxAUC@k$	$MaxPPT@k$
τ^2-bench Airline Domain (Special Split: Samples 7, 14, 21, 23, and 29)				
gpt-4o-mini + EI	**0.37** ↑ 0.09 | 0.31 ↓ 0.01	**0.66** ↑ 0.06 | 0.61 ±0.00	**0.59** ↑ 0.05 | 0.39 ↓ 0.02	**0.27** ↑ 0.03 | 0.11 ↓ 0.01
gpt-4o-mini + HN	0.30 ↑ 0.02 | **0.33** ↑ 0.01	0.63 ↑ 0.02 | 0.61 ±0.00	0.53 ↓ 0.01 | **0.45** ↑ 0.04	0.24 ±0.00 | **0.14** ↑ 0.02
gpt-4.1 + EI	0.52 ±0.00 | 0.38 ±0.00	0.78 ↑ 0.04 | 0.77 ↓ 0.08	0.66 ↑ 0.02 | 0.41 ↓ 0.06	0.26 ↓ 0.01 | 0.10 ↓ 0.02
gpt-4.1 + HN	**0.53** ↑ 0.01 | **0.41** ↑ 0.03	0.78 ↑ 0.04 | 0.85 ±0.00	0.66 ↑ 0.02 | **0.55** ↑ 0.08	0.27 ±0.00 | **0.14** ↑ 0.02
ToolSandbox Dataset				
gpt-4o-mini + EI	0.87 ↓ 0.04 | 0.89 ↑ 0.04	**0.97** ↑ 0.02 | **0.98** ↑ 0.05	0.94 ±0.00 | 0.91 ↑ 0.01	0.85 ↓ 0.04 | 0.66 ↓ 0.11
gpt-4o-mini + HN	0.88 ↓ 0.03 | **0.91** ↑ 0.06	0.96 ↑ 0.01 | 0.96 ↑ 0.03	**0.95** ↑ 0.01 | **0.92** ↑ 0.02	**0.90** ↑ 0.01 | 0.74 ↓ 0.03
gpt-4.1 + EI	0.95 ↑ 0.03 | 0.93 ↑ 0.06	**0.99** ↑ 0.01 | 0.99 ↑ 0.02	0.97 ↑ 0.01 | 0.93 ↑ 0.01	0.87 ↑ 0.03 | 0.76 ↑ 0.03
gpt-4.1 + HN	0.95 ↑ 0.03 | **0.97** ↑ 0.10	0.98 ±0.00 | 0.99 ↑ 0.02	0.97 ↑ 0.01 | **0.95** ↑ 0.03	**0.91** ↑ 0.07 | **0.83** ↑ 0.10

5. Conclusion

- We introduced the *Talk*, *Evaluate*, *Diagnose* (TED) framework that redefines agent evaluation based on three stages.

- We also showed potential gains in agent performance with peaks of 8-10% on our proposed metrics after incorporating the identified error remedies into the agent's design.

- In the future, we plan to explore the applicability of our metric to non-task-oriented domains, such as open-ended dialogue with conversational agents.

Thank you.

QR code to paper:

QR code to github: