

SEATauBench: Progressively Adapting Tool-Agent-User Evaluation Into Low-Resource Southeast Asian Languages

My Chiffon Nguyen¹, Aulia Adila^{1,2}, Saksorn Ruangtanusak^{1,3},
Kittiphat Leesombatwathana^{1,4,5*}, Vissuta Gunawan Lim^{1,6},
Patomporn Payoungkhamdee^{1,5}, Samuel Cahyawijaya^{1,7}

¹SEACrowd, ²AI Singapore, Nanyang Technological University, ³SCB DataX, SCBX Group,
⁴Chulalongkorn University, ⁵VISTEC, ⁶University of Indonesia, ⁷Cohere
chiffonng136@gmail.com, aulia.adila@aisingapore.org,
saksorn.ruangtanusak@data-x.ai, vissuta.gunawan@ui.ac.id
{kittiphat.l_s26, patomporn.p_s21}@vistec.ac.th, samuelcahyawijaya@cohere.com

Abstract

While AI development and evaluation for Southeast Asia (SEA) has grown rapidly, agent capabilities in regional languages are still poorly understood despite its importance to sovereign AI. To fill this gap, we introduce SEATauBench, the first agent-focused evaluation framework for SEA sovereign AI. It adapts τ^2 -Bench (Barres et al., 2025) to five languages—Mandarin, Vietnamese, Thai, Indonesian, and Filipino—and evaluates agents across progressively localized settings that vary the language of user-agent interaction, tool specifications, and task domains. Across three models, we find that English agent capabilities transfer reasonably well when only the conversation language changes, but quality and robustness degrade sharply as more task contexts are localized, with the largest losses in full domain adaptation. We also highlight the limits of English-only agent assessment for predicting agent capabilities in SEA languages. More broadly, SEATauBench provides a diagnostic benchmark and reusable adaptation pipeline for building reliable multilingual agents for linguistically diverse regions. Data and code can be accessed at <https://github.com/SEACrowd/SEATauBench>¹.

1 Introduction

Sovereign AI seeks autonomy over digital futures while keeping systems relevant to the languages, institutions, and cultural settings of the people they serve (Chae et al., 2025; Mushkani et al., 2025; Putra, 2024). Southeast Asia (SEA) is home to more than 700 million people, yet English-centric development and evaluation provide an incomplete picture of agent capability across the region (Bhandari and Modi, 2026; Lovenia et al., 2024).

*Most of this work was conducted while Kittiphat Leesombatwathana was at Chulalongkorn University.

¹SEATauBench is pronounced "si-tau-bench", similar to the Filipino word for string beans, "sitaw".

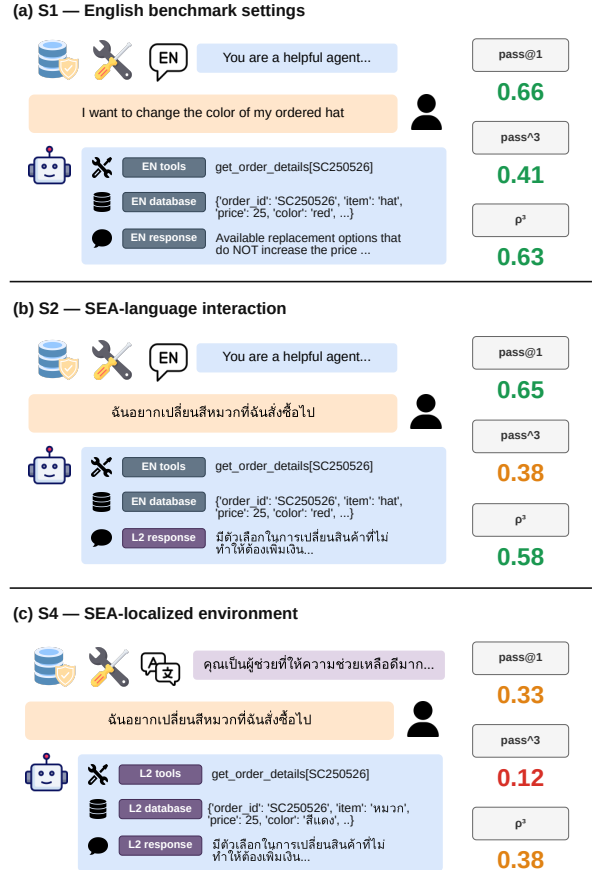


Figure 1: Illustration of SEATauBench’s progressive localization design. The same task is evaluated while dialogue, tool interfaces, and domain context are localized one surface at a time.

Existing benchmarks often isolate these capabilities. A tool-using agent must understand a user’s language, follow a domain policy, ground decisions in structured state, invoke tools, and complete tasks over multiple turns. Localizing every surface at once, however, obscures the source of failure.

We introduce SEATauBench, a multilingual extension of τ^2 -Bench (Barres et al., 2025) for five SEA languages: Vietnamese, Indonesian, Thai, Filipino, and Mandarin Chinese. Four controlled scenarios progressively localize the dialogue, tool schemas, and domain context: S1 is the English

Benchmark	Language coverage	Interaction	Simulated user	Action/state grounding	Localization depth	Multiple runs
MASSIVE-Agents (Kulkarni et al., 2025)	52	Single-turn	✗	Function calls; AST	User utterances; English names	✗
Ticket-Bench (Almeida et al., 2025)	6	Single-query	✗	Ticket functions; final state	Queries, entities, functions	✓
X-WebAgentBench (Wang et al., 2025)	14	Interactive web steps	✗	Search/click; product catalog	Instructions, products, web UI	✗
WebMMU (Awal et al., 2025)	4	Offline task instances	✗	UI grounding; code edits	Web content, prompts	✗
MAPS (Hofman et al., 2026)	12	Mixed agent loops	✗	Native tools; task executors	Instructions only	✓
macOSWorld (Yang et al., 2025)	5	Interactive GUI steps	✗	Mouse/keyboard; OS state	Instructions, OS UI	✗
AgentClinic (Schmidgall et al., 2025)	7	Multi-turn	✓	Exam/imaging tools; diagnosis	Patient cases, dialogue	✗
SEATauBench (Ours)	6	Multi-turn	✓	API tools; final DB	Dialogue, tools, policies, DB	✓

Table 1: Multilingual agent benchmarks compared with SEATauBench. All benchmarks with multiple run evidence report three trials per task. SEATauBench is the only benchmark here that progressively localizes user dialogue, tool interfaces, and domain context in multi-turn tasks.

baseline, S2 localizes the interaction, S3 localizes the tool schemas, and S4 localizes the full task domain. The progression separates language generation from tool grounding and domain reasoning while keeping the underlying tasks and scoring interface aligned.

The benchmark contributes three components: (1) a **progressive localization benchmark** with four scenarios that isolate dialogue, tool schemas, and domain context; (2) a **pipeline that separates localized agent-facing context from canonical execution** with automated and human audits; and (3) a **reliability analysis** showing how quality and robustness across repeated runs vary by language, domain, and localization depth.

Across three agent models, English task success transfers reasonably well when only dialogue is localized, but quality and robustness decline as tools and domain context are localized, with the largest losses in L2 Domain (Section 5.1). The decline varies by language, model, and domain, and English performance does not reliably predict repeated-run robustness in SEA languages (Sections 5.2 and 6.3). SEATauBench therefore serves as a diagnostic benchmark for multilingual, tool-using agents, especially for the SEA region.

2 Related Work

2.1 Agent evaluation

Task-oriented dialogue benchmarks such as MultiWOZ (Budzianowski et al., 2018) and MASSIVE (FitzGerald et al., 2023) establish evaluation for goal-directed dialogue and multilingual intent and slot understanding. ToolLLM (Qin et al., 2024) and the Berkeley Function Calling Leaderboard (Patil et al., 2025) measure tool selection and argument generation, while ToolSandbox (Lu et al., 2025) adds conversational context, mutable state, and action dependencies.

These benchmarks isolate useful capabilities, but

they do not fully test an agent that infers a user’s goal, updates external state, and repeats a specialized task across multiple turns. τ -Bench (Yao et al., 2024) evaluates this setting with a simulated user, domain-specific tools, and final-state metrics. τ^2 -Bench (Barres et al., 2025) adds dual control for telecom domain. These English-centered benchmarks provide the interaction structure that SEATauBench extends with controlled multilingual conditions and progressive localization.

2.2 Multilingual evaluation

Multilingual resources broaden evaluation beyond English. mMMLU (Hendrycks et al., 2021) and GlotEval (Luo et al., 2025) measure multilingual knowledge and language understanding. For Southeast Asian languages, NusaX (Winata et al., 2023) and NusaWrites (Cahyawijaya et al., 2023b) provide low-resource language resources, while NusaCrowd (Cahyawijaya et al., 2023a) and SEACrowd (Lovenia et al., 2024) provide curated multilingual data and benchmark infrastructure. SeaExam and SeaBench (Liu et al., 2025) cover local questions and open-ended tasks; SEA-HELM (Susanto et al., 2025) covers holistic linguistic, cultural, and safety evaluation; and SEA-VL (Cahyawijaya et al., 2025) and SEA-VQA (Urailertprasert et al., 2024) extend coverage to multimodal data. These resources are essential, but they largely assess static or closed-ended behavior rather than tool-mediated task completion with a simulated user and structured state.

2.3 Multilingual agent evaluation

Recent work brings multilingual evaluation closer to interactive agents. Table 1 summarizes the design choices that matter for this setting. MASSIVE-Agents (Kulkarni et al., 2025) scales function calling to 52 languages, but evaluates single-turn selection and argument prediction. Ticket-

Bench (Almeida et al., 2025) localizes entities and functions in a state-based ticket service, but remains single-query. X-WebAgentBench (Wang et al., 2025) evaluates multilingual planning in an interactive web environment, and WebMMU (Awal et al., 2025) evaluates multilingual web question answering, code editing, and mockup-to-code generation. MAPS (Hofman et al., 2026) translates instructions for established agent benchmarks and evaluates performance and security. macOSWorld (Yang et al., 2025) evaluates multilingual GUI agents on macOS, while Agent-Clinic (Schmidgall et al., 2025) combines multilingual clinical cases with simulated doctor-patient interaction and tools. These benchmarks are complementary, but each focuses on a particular interaction environment or localized surface.

SEATauBench fills this gap by placing a simulated user, tools, policies, and structured state in one multilingual loop. It localizes one agent-visible surface at a time while retaining a canonical execution layer, separating conversational transfer, tool grounding, and domain adaptation. Within this comparison, SEATauBench is, to our knowledge, the first multilingual benchmark to preserve the task-oriented methodology of τ^2 -Bench while controlling localization across dialogue, tools, and domain context. This design also supports repeated-run reliability analysis, which reveals failures that a single successful trial can hide.

3 Constructing SEATauBench

τ^2 -Bench is an English benchmark for tool, agent, and user interaction in three service domains: retail, airline, and telecom (Yao et al., 2024; Barres et al., 2025). Each task exposes the agent to three types of language: dialogue with a simulated user, tool schemas, and domain context such as task instructions, policies, workflows, and database content.

SEATauBench adapts this setting to four controlled scenarios (Section 3.1) and five target languages (L2): Vietnamese (vi), Indonesian (id), Thai (th), Filipino (tl), and Mandarin Chinese (zh) (Section 3.2).² L2 henceforth denotes a non-English target language. Rather than translating the benchmark as a single block, we localize each surface visible to the agent separately. This design isolates the effects of multilingual dialogue, tool grounding, and domain reasoning while preserving a common

²Due to limited native speaker review capacity, we do not include other SEA languages, such as Malay and Lao.

Scenario / L2	Tools	Dialogue	DB+Policy
(S1) English Baseline	✗	✗	✗
(S2) L2 Interaction	✗	✓	✗
(S3) L2 Tool	✓	✗	✗
(S4) L2 Domain	✓	✓	✓

Table 2: SEATauBench’s four scenarios. Checkmarks show which agent-visible surfaces are presented in L2; S2 and S3 isolate dialogue and tool schemas, while S4 localizes the task environment.

task definition and scoring procedure.

3.1 Progressive localization scenarios

S1. English Baseline. S1 is the English baseline. Dialogue, tool schemas, task instructions, policies, and database content remain in English.

S2. L2 Interaction. The simulated user and agent communicate in L2, while tools, policies, databases, and task context remain in English. This scenario isolates multilingual dialogue from localized tool use and domain reasoning. Changes relative to S1 can reflect both agent behavior and changes in the simulated user’s behavior.

S3. L2 Tool. The tool schemas shown to the agent are localized into L2. The conversation and domain context remain in English, and tool names and executable implementations remain unchanged. This scenario tests whether the agent can ground tool semantics, parameter descriptions, and allowed values in another language. We test a variant with one L2 for every schema and a mixed variant in which schemas use multiple L2 languages within the same environment during each run.

S4. L2 Domain. The simulated user and agent communicate in L2, and the tool descriptions, policies, task descriptions, and database content shown to the agent are also in L2. The underlying task definitions, tool behavior, canonical values, and expected final states remain equivalent to the English benchmark. This preserves executable behavior and makes scores comparable across scenarios.

Together, the scenarios measure how each additional localized surface affects task completion. The progression also separates a dialogue language effect from failures that require the agent to interpret localized tools or domain state.

3.2 L2 adaptation pipeline

Comparable scores require localized surfaces to preserve task meaning and executable behavior. We

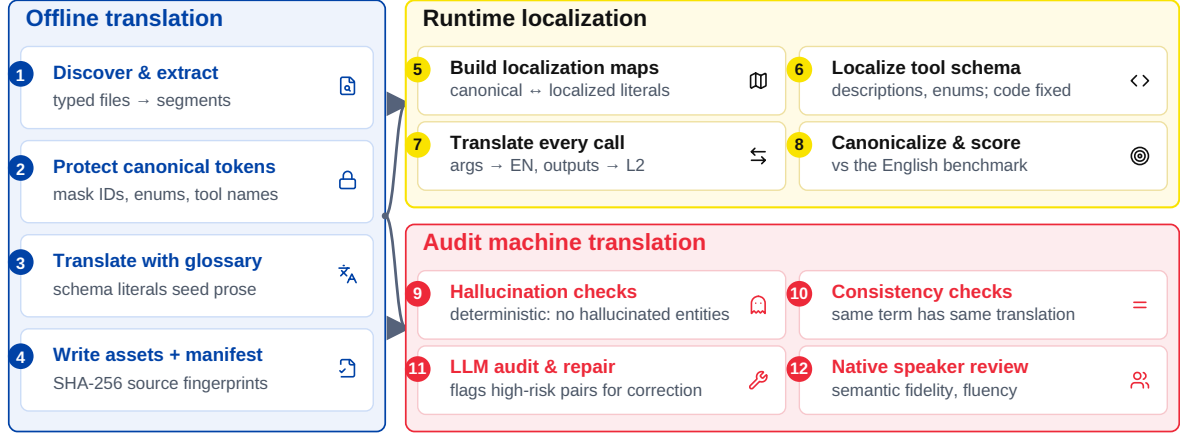


Figure 2: The localization pipeline for (S2) L2 Tools and (S4) L2 Domain, as defined in Table 2. Offline translation creates static localized assets, runtime localization preserves canonical execution, and automated and native speaker audits characterize translation quality. We use Gemini 3.1 Flash Lite for offline translation into five target languages.

separate the adaptation into an offline stage that creates translated assets and a runtime stage that connects those assets to the canonical benchmark implementation.

Offline translation. The offline stage uses Gemini 3.1 Flash Lite³ to build language-specific assets for the three τ^2 -Bench domains. It extracts natural language spans from tasks, policies, databases, tool docstrings, and schemas, while masking executable tokens such as IDs, status values, tool names, and structural markers. The pipeline translates schema literals first to build a glossary, then translates the remaining spans while preserving original format. It also records a per-language manifest with model metadata and source file SHA-256 fingerprints. Appendix A.5 reports the resulting artifact statistics.

Runtime localization. Runtime localization handles content that appears during inference. The wrapper localizes tool schema descriptions, enum choices, default values, and examples in L2 while leaving the implementation unchanged. Before each tool call, it maps localized arguments to canonical English values. After the call, it localizes tool responses and converts final payloads to their canonical forms before scoring. The agent therefore interacts in L2 while the evaluator compares equivalent canonical outcomes across languages.

3.3 L2 translation audit

Translation errors can make a task harder or impossible, alter values needed for execution, or reduce

target-language fidelity. We audit these risks in the frozen *experiment snapshot* through solvability analysis, preservation and consistency checks, and native speaker review; approved corrections enter a separate *reviewed release*. We first measure empirical achievability. A task is solvable when any of nine runs—three agent models with three trials each—is successful with reward 1. Table 3 shows near-complete solvability in Telecom but larger cross-language gaps in Airline and Retail. Inspection of the 16 tasks solved in English but not in L2 points mainly to incomplete multi-step execution, although translation may contribute (Appendix B.2).

Domain	EN	ID	TH	TL	VI	ZH
Airline	92.0%	82.0%	86.0%	84.0%	84.0%	96.0%
Retail	95.6%	82.5%	83.7%	85.1%	91.2%	90.4%
Telecom	100.0%	95.6%	96.5%	99.1%	94.7%	100.0%

Table 3: Empirical solvability. A task is counted when any of three agent models succeeds in one of three trials. Percentages use 50 Airline tasks and 114 Retail and Telecom tasks each.

Beyond solvability, deterministic checks cover 11,416 translation pairs, with targeted model review of 974; separate consistency checks cover 14,890 retail order items and 430 schema literals. Reviewers approved 2,894 corrections across 32 artifacts, after which the reviewed release passed the recorded checks. Each annotation sheet contains a required sample of 90 or 91 units and additional review; at least one native speaker per language completed 167–254 unique units in total. An ok translation is faithful and usable, a *minor_issue*

³We invoke this model through the LiteLLM route `vertex_ai/gemini-3.1-flash-lite`.

affects wording without changing meaning or use, and a `major_issue` can change the correct action or result. For each language, we recalibrate the finite-population acceptance number c to its population size and total completed review count. Table 4 reports all verdicts and the resulting threshold. The observed major-issue count does not exceed c for any language, so all five pass. Appendix B provides the calculation, selection caveat, and full evidence.

L2	Reviewed	OK	Minor	Major	c	Decision
ID	254	235	17	2	7	Pass
TH	200	189	11	0	5	Pass
TL	183	141	42	0	4	Pass
VI	167	142	24	1	3	Pass
ZH	200	161	38	1	5	Pass

Table 4: Native speaker review over all unique units with verdicts. For each language, c is calculated from its population size and reviewed count following the finite-population acceptance sampling procedure; Pass means that the observed major-issue count is at most c .

4 Experimental Setup

Metrics. Following τ^2 -Bench (Barres et al., 2025), we report two task metrics. First, `pass@1` measures quality as the mean success rate of independent trials reaching the expected final state, i.e., $\text{pass@1} = \text{pass}^1 = \mathbb{E}[r]$. Building on existing metrics, we also derive a robustness measure, $\rho \in [0, 1]$, defined as

$$\rho^q = \frac{\text{pass}^q}{\text{pass@1}} \quad (1)$$

where ρ^q denotes the probability that all q independent trials succeed, averaged across tasks. $\rho^q = 1$ indicates that the agent performs consistently across multiple runs, while lower ρ^q indicates that the agent may solve a task once but fail to do so reliably across repeated trials.

Task and Domain. For high-quality evaluation, we use latest versions of retail and airline domains (Cuadron et al., 2026), which incorporate several corrections and refinements over prior versions (Yao et al., 2024; Barres et al., 2025). For the telecom domain, we follow the original τ^2 -Bench, as its tasks were not updated in the most recent release. We leave exploration of the banking domain (Shi et al., 2026) and the voice modality (Ray et al., 2026) introduced in τ^3 -bench to future work.

Hyperparameters and Models. For all evaluations, we use $q = 3$, resulting in the robustness metric ρ^3 , and Qwen3-235B-A22B-Instruct (Qwen-Team, 2025) as the user LLM. For the natural-language assertion judge, we use GPT-4.1, following the implementations provided in τ^2 -Bench. We evaluate three recent LLM agents spanning three model families: GPT-5-Mini (Singh et al., 2026) as a proprietary model, Qwen3-235B-A22B-Instruct-2507 as a representative open-source model, and Kimi-K2.5 (Kimi Team et al., 2026) as a larger-scale open-source model.

For each model, we use the default hyperparameters defined in each provider (see Appendix C). We avoid any model-specific hyperparameter tuning to better reflect the out-of-the-box tool-user-agent interaction in the specified scenario and L2.

5 Results by localization setting

Across four scenarios, `pass@1` declines as more of the task environment moves from English to L2, while robustness, model rankings, and domain difficulty vary. We first compare aggregate setting means, then examine model and domain effects.

5.1 Task quality across settings

Average `pass@1` falls across settings. `Pass@1` declines as the benchmark localizes more of the task environment (Figure 3). The mean falls from 0.62 in the English Baseline to 0.61, 0.55, and 0.51 in L2 Interaction, L2 Tool, and L2 Domain, respectively. Because L2 Tool includes two agents and the other settings include three, this comparison is descriptive rather than fully matched.

L2 Domain quality varies by language. Across models and domains, L2 Domain `pass@1` is lowest for Thai (0.45), followed by Filipino (0.51), Indonesian (0.51), and Vietnamese (0.52), while Chinese is highest (0.58). Compared with the English Baseline (0.62), performance is lower in both Latin- and non-Latin-script languages, although the magnitude differs across languages. These results do not support reducing the gap to script type alone; they instead show language-specific differences in fully localized settings.

Quality and robustness diverge. ρ^3 measures repeated success relative to single-run success. The aggregate ρ^3 values are 0.66, 0.64, 0.60, and 0.62 for the English Baseline, L2 Interaction, L2 Tool, and L2 Domain, respectively. In L2 Domain, Chinese has the highest ρ^3 (0.68) and Thai the lowest

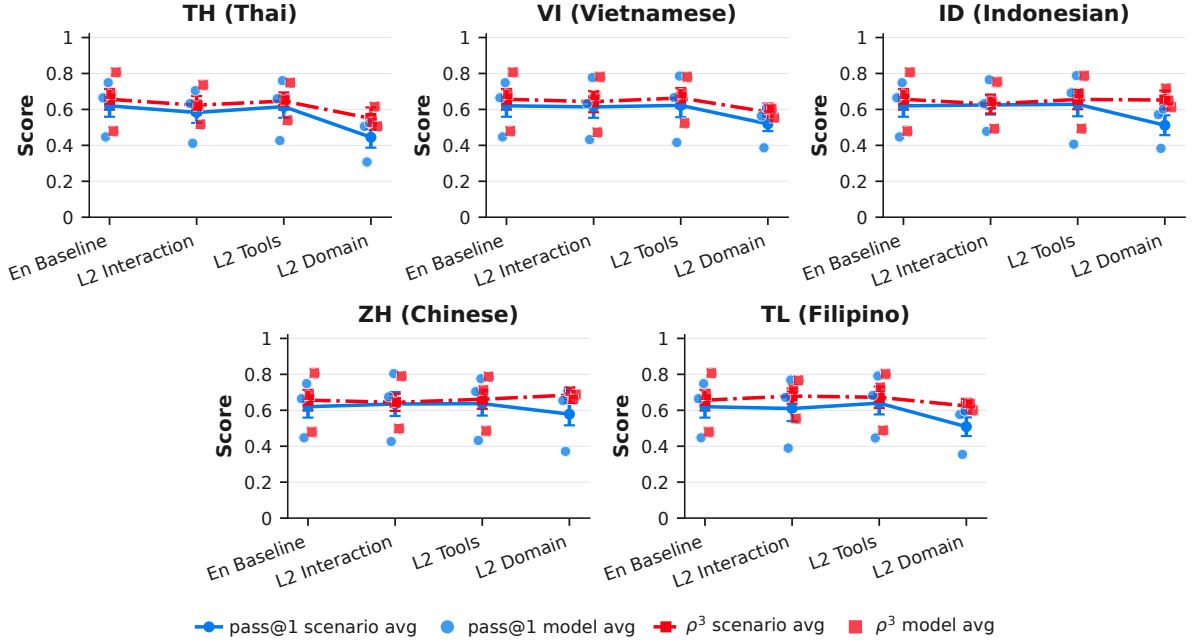


Figure 3: Evaluation results on SEATauBench. SEATauBench reveals consistent degradation as the setting becomes less English-centric on all languages in both pass@1 and ρ^3 metrics, with most severe drops in low-resource languages with non-Latin scripts.

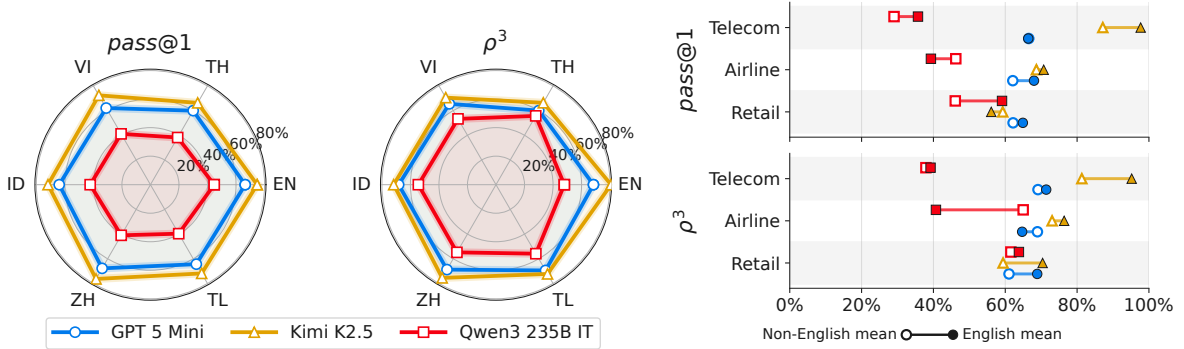


Figure 4: Quality and robustness of varying agent models across languages and domains. **(left)** All agents yield consistently higher pass@1 in English and Chinese indicating a quality bias toward high-resource languages. **(center)** A similar trend is observed on ρ^3 , with negative correlation of ρ^3 between English and L2 languages especially Thai. **(right)** pass@1 trend holds across most domains and models with some exceptions as described in Section 5.2.

(0.55), but several languages retain moderate robustness despite lower pass@1 . A high ρ^3 therefore does not indicate high task quality: it indicates that the tasks an agent solves are solved consistently. Together, the metrics distinguish task success from behavioral reliability.

5.2 Model and domain rankings shift

Model rankings vary by setting and metric. No single agent dominates every condition (Figure 4). In the English Baseline, Kimi-K2.5 has the highest mean pass@1 (0.75), followed by GPT-5-mini (0.66) and Qwen3-235B-A22B-Instruct-2507

(0.45). In L2 Domain, Kimi-K2.5 remains highest on pass@1 (0.61), but Qwen3-235B has higher mean ρ^3 (0.63) than GPT-5-mini (0.62) and Kimi-K2.5 (0.59). The model with the lowest probability of solving a task can therefore still be consistent on the subset of tasks it solves, reinforcing the need to evaluate quality and robustness separately.

Domain difficulty varies by setting. In L2 Domain, Retail has the lowest mean pass@1 for all three agents: 0.51 for GPT-5-mini, 0.48 for Kimi-K2.5, and 0.25 for Qwen3-235B. Telecom is easiest for GPT-5-mini and Kimi-K2.5, whereas Airline

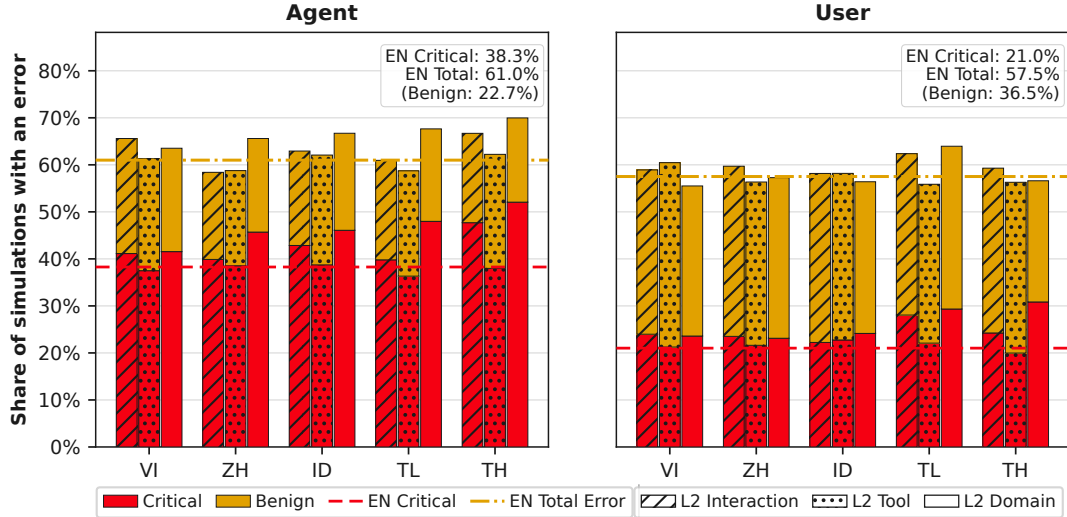


Figure 5: Critical, benign, and correct simulation outcomes for the simulated user and agent. The analysis uses DeepSeek-V4-Flash as the judge, Qwen3-235B-A22B-Instruct-2507 as the user, and GPT-5-mini as the agent.

is easiest for Qwen3-235B. Qwen3-235B shows the clearest reversal across settings: Retail is its strongest English domain (0.59) but its weakest L2 Domain (0.25). These results show that English domain rankings do not reliably transfer to a fully localized task environment.

6 Diagnosing the localization gap

We examine three signals of the localization gap: critical errors, language correctness versus robustness, and cross-language correlations. These analyses distinguish execution failures from patterns of language use and test whether English scores can serve as proxies for L2 performance. Mixed-language tool results and language drift diagnostics appear in Appendices E and F.

6.1 Critical errors in localized settings

We use the LLM-as-a-Judge framework proposed in Shi et al. (2026) to classify errors. Following Barres et al. (2025), a *Critical* error deviates from user intent or is irrecoverable, whereas a *Benign* error does not prevent task completion. For the severity comparison, we consider L2 Interaction and L2 Domain; L2 Tool translates only tool specifications, leaving tool, agent, and user interactions in English. The judge is DeepSeek-V4-Flash, with GPT-5-mini as the agent and Qwen3-235B-A22B-Instruct-2507 as the simulated user.

Agent critical errors rise in L2. Agent critical error rates are higher in L2 Domain than in the English reference, with Thai showing the largest

increase. In L2 Interaction, critical errors range from 39.80% to 47.70% across L2 languages, compared with 38.27% for the English reference. In L2 Domain, they range from 41.54% to 52.06%, with Thai at 52.06% and Filipino at 47.98% (Figure 5). The increase in critical rather than benign errors is consistent with failures in executing the intended task, not only with less fluent dialogue.

User critical errors vary by setting. User critical error rates are also higher in L2 than in the English reference. They range from 22.20% to 28.10% in L2 Interaction and from 23.10% to 30.80% in L2 Domain, compared with 21.02% for the English reference. These rates show that user-simulation reliability also differs across settings, consistent with prior findings for simulated users in English and dialectal English (Barres et al., 2025; Seshadri et al., 2026). The error analysis is therefore diagnostic rather than a causal decomposition of the end-to-end performance gap.

Agents omit actions and misread tasks. Figure 6 shows three recurring mistakes across all four scenarios: omitting a required action, violating a task guideline, and misreading the task or user intent. These leading errors occur at roughly 3 to 6 occurrences per 100 agent turns, against about 0.7 to 1.2 for the next tier and below 0.2 for the remainder. The ordering is stable across scenarios, but L2 Domain is consistently highest, with differences of about 1.2 to 1.5 occurrences per 100 turns relative to the English baseline. Because a

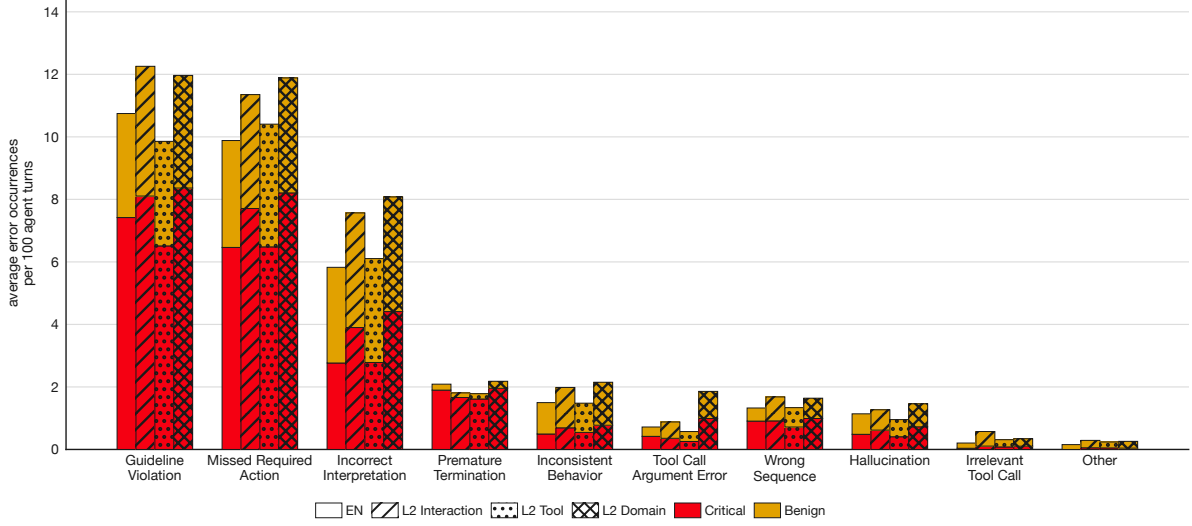


Figure 6: Breakdown of agent errors different across progressive localization scenarios. The trends are consistent across scenarios, with the most errors being Missed Required Action and Guideline Violation. Error taxonomy detailed in Appendix H.1.

single turn may carry several errors and each error several types, these are error type rates rather than a decomposition of failures.

The two tool schema categories are far smaller: *Tool Call Argument Error* reaches 0.91 and *Tool Call Schema Error* 0.12 per 100 agent turns under L2 Domain, against 6.3 and 6.2 for the two leading policy categories. They rise most steeply in relative terms from S1 to S4 ($2.8\times$ and $3.8\times$, against $1.3\times$ for the policy errors) yet remain an order of magnitude smaller in absolute terms, and the same ordering holds in S3, where only the schemas are localized. The dominant bottleneck is therefore multi-step policy and domain reasoning rather than reading L2 tool specifications.

6.2 Language correctness versus robustness

Language correctness is the fraction of eligible turns in the expected language; system error trajectories are excluded. Robustness is ρ^3 , the probability that all three independent trials succeed, averaged across tasks (Section 4). We use Kendall’s τ_b to summarize monotonic association because correctness contains tied values (Figure 7).

Correctness–robustness association is weak overall. Across 168 runs, the pooled Kendall association is weak ($\tau_b = 0.07$). The scenario-specific values are 0.36 in L2 Tool, 0.20 in L2 Interaction, 0.04 in L2 Domain, and -0.24 in the English Baseline. These values indicate at most a modest monotonic association, not a predictive or

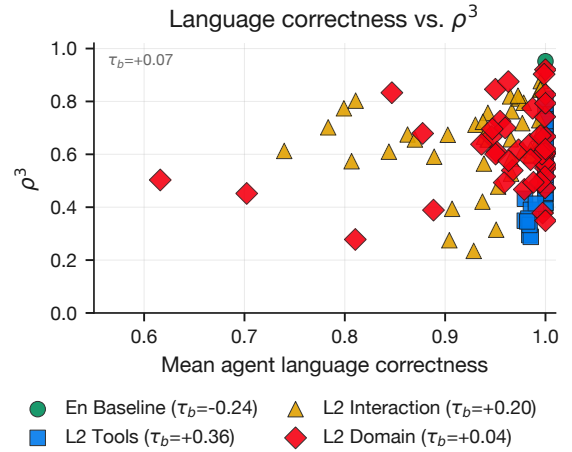


Figure 7: Agent language correctness versus robustness (ρ^3) across 168 runs. We report Kendall’s rank correlation τ_b for each scenario, and pooled association. The English Baseline is shown for context, but its language correctness has limited variation with 9 runs.

causal relationship.

The L2 Tool association is setting specific. The larger L2 Tool association does not make language correctness a general proxy for robustness. Correctness is concentrated near 1.00 in L2 Tool, while L2 Domain contains runs with near-perfect correctness and a broad range of ρ^3 values. L2 Interaction has more variation in correctness, but its association remains modest. The relationship therefore depends on the scenario and its available predictor variation; correctness alone does not account for the observed

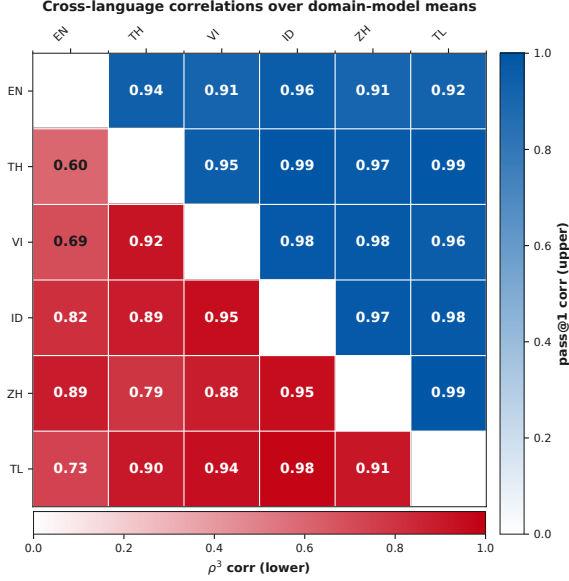


Figure 8: Cross-language Pearson correlations for pass@1 (upper triangle) and ρ^3 (lower triangle). Each correlation uses domain-model means after averaging over available scenarios, with three domains and three agent models.

robustness differences. Complementary drift diagnostics likewise show that off-target language is measurable and scenario dependent, without establishing that it explains the full performance gap (Appendix F).

6.3 English quality as an L2 proxy

English pass@1 tracks average L2 quality. English pass@1 is a useful coarse predictor of average L2 quality, but it is a poor predictor of repeated run reliability. Because the full benchmark is expensive to run, we correlate language-level pass@1 and ρ^3 over domain-model means (Figure 8). English correlations with L2 pass@1 range from 0.90 for Vietnamese to 0.96 for Indonesian, whereas English correlations with L2 ρ^3 range from only 0.49 for Thai to 0.88 for Chinese.

Filipino correlates across L2 languages. Filipino is a strong candidate among the L2 languages for a cross-language proxy in this analysis. Its correlations with the other four L2 languages range from 0.94 to 0.99 for pass@1 and from 0.85 to 0.94 for ρ^3 . A Filipino subset can therefore reduce evaluation cost when a proxy is necessary, but these correlations do not establish validity beyond the five languages, three domains, and three models studied here. For deployment, validate the proxy against the target language.

7 Conclusion

SEATauBench extends agent evaluation beyond translated dialogue by progressively localizing the surfaces on which task completion depends. Its four scenarios separate conversational language from tool schemas and domain context, while the adaptation pipeline preserves a canonical execution layer for comparable scoring. This design makes it possible to ask not only whether an agent can converse in a language, but whether it can reliably complete multi-step tasks when operational information is expressed in that language.

Across five widely spoken SEA languages, performance generally declines as localization reaches deeper into the task environment. The results also show why multilingual agent evaluation should report both quality and robustness. Model rankings shift across settings, and repeated-run consistency can remain moderate even when task success is low. English performance is therefore an incomplete proxy for multilingual readiness. By releasing controlled scenarios, localized assets, and a translation audit methodology, SEATauBench provides a foundation for diagnosing where multilingual agents fail and for evaluating whether future systems improve across languages rather than only on English-centric interfaces.

8 Limitations

We evaluate Vietnamese, Indonesian, Thai, Filipino, and Chinese in the retail, airline, and telecom domains. Although the framework can extend to other settings, the findings may not generalize beyond these languages, domains, task distributions, or agent models. Correlations reported over domain-model means are descriptive within this benchmark and should not be interpreted as evidence that one language will remain a reliable proxy in other deployment settings.

The adaptation is also specific to the τ -Bench paradigm (Yao et al., 2024; Barres et al., 2025; Ray et al., 2026; Shi et al., 2026). Its simulator, policies, and canonical tool layer provide experimental control but do not capture every ambiguity, cultural expectation, or interface constraint that arises in deployed systems. Future work should test whether the localization progression transfers to other agent benchmarks and real user environments.

The reported experiments use the immutable experiment snapshot rather than the reviewed release. Passing the native speaker audit and our acceptance

thresholds does not imply an error-free translation: the completed reviews contain two major issues in Indonesian and one each in Vietnamese and Chinese. Cross-language gaps may therefore include residual translation error. Future work should rerun the reviewed release and expand probability-based native-speaker review.

9 Ethical Statement

We conducted this research with careful attention to ethical considerations throughout the development process. In developing of the translation pipeline and the evaluation framework, we ensure that there is no external sensitive information is exposed throughout the process and the resulting artifacts. We recognize our responsibility to ensure these systems are fair, respectful of different cultures, and beneficial to all users. Our approach focuses on being transparent about what the agents can and cannot do, actively working to identify and address potential biases, and taking steps to prevent misuse. We handle all research data responsibly, protecting privacy while respecting linguistic diversity. This work aims to contribute to AI that serves people equitably across languages and cultures, and we are committed to being accountable for how these technologies impact society.

Acknowledgments

This work was supported by the National Research Foundation, Singapore, under its National Large Language Models Funding Initiative, and by the ThaiLLM collaboration through the Digital Economy and Society (DE) Development Fund of the Ministry of Digital Economy and Society, Thailand. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of any of the funding organizations.

We thank Kian Kyars for providing OpenAI API access, Dittaya Wanvarie for providing OpenRouter credits, and SCBX R&D for providing the Google Cloud Vertex AI resources used in our experiments. We thank Mark Joseph Pasciolco, Manuel Antonio Rufino, and Joseph Imperial for supporting the Filipino language review; and Melanie Bui for Mandarin Chinese language review.

References

Thales Sales Almeida, Joao Guilherme Alves Santos, Thiago Laitz, and Giovana Kerche Bonás. 2025.

[Ticket-bench: A kickoff for multilingual and regionalized agent evaluation](#). *Preprint*, arXiv:2509.14477.

Rabiul Awal, Mahsa Massoud, Aarash Feizi, Zichao Li, Suyuchen Wang, Christopher Pal, Aishwarya Agrawal, David Vazquez, Siva Reddy, Juan A. Rodriguez, Perouz Taslakian, Spandana Gella, and Sai Rajeswar. 2025. [WebMMU: A benchmark for multi-modal multilingual website understanding and code generation](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 25118–25145, Suzhou, China. Association for Computational Linguistics.

Victor Barres, Honghua Dong, Soham Ray, Xujie Si, and Karthik Narasimhan. 2025. [\$\tau^2\$ -bench: Evaluating conversational agents in a dual-control environment](#). *Preprint*, arXiv:2506.07982.

Manik Bhandari and Gaurav Modi. 2026. [Why sovereign artificial intelligence is imperative in south-east asia](#). Accessed: 22 May 2026.

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. 2018. [MultiWOZ - a large-scale multi-domain Wizard-of-Oz dataset for task-oriented dialogue modelling](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5016–5026, Brussels, Belgium. Association for Computational Linguistics.

Samuel Cahyawijaya, Holy Lovenia, Alham Fikri Aji, Genta Winata, Bryan Wilie, Fajri Koto, Rahmad Mahendra, Christian Wibisono, Ade Romadhony, Karissa Vincentio, Jennifer Santoso, David Moeljadi, Cahya Wirawan, Frederikus Hudi, Muhammad Satrio Wicaksono, Ivan Parmonangan, Ika Alfina, Ilham Firdausi Putra, Samsul Rahmadani, and 29 others. 2023a. [NusaCrowd: Open source initiative for Indonesian NLP resources](#). In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13745–13818, Toronto, Canada. Association for Computational Linguistics.

Samuel Cahyawijaya, Holy Lovenia, Fajri Koto, Dea Adhista, Emmanuel Dave, Sarah Oktavianti, Salsabil Akbar, Jhonson Lee, Nuur Shadieq, Tjeng Wawan Cenggoro, Hanung Linuwih, Bryan Wilie, Galih Muridan, Genta Winata, David Moeljadi, Alham Fikri Aji, Ayu Purwarianti, and Pascale Fung. 2023b. [NusaWrites: Constructing high-quality corpora for underrepresented and extremely low-resource languages](#). In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 921–945, Nusa Dua, Bali. Association for Computational Linguistics.

Samuel Cahyawijaya, Holy Lovenia, Joel Ruben Antony Moniz, Tack Hwa Wong, Mohammad Rifqi Farhan-syah, Thant Thiri Maung, Frederikus Hudi, David Anugraha, Muhammad Ravi Shulthan Habibi,

- Muhammad Reza Qorib, Amit Agarwal, Joseph Marvin Imperial, Hitesh Laxmichand Patel, Vicky Feliren, Bahrul Ilmi Nasution, Manuel Antonio Rufino, Genta Indra Winata, Rian Adam Rajagede, Carlos Rafael Catalan, and 73 others. 2025. [Crowdsource, crawl, or generate? creating SEA-VL, a multicultural vision-language dataset for Southeast Asia](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 18685–18717, Vienna, Austria. Association for Computational Linguistics.
- Kyubyeung Chae, Gihoon Kim, Gyuseong Lee, Taesup Kim, Jaemin Lee, and Heejin Kim. 2025. [Assessing socio-cultural alignment and technical safety of sovereign LLMs](#). In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 10579–10600, Suzhou, China. Association for Computational Linguistics.
- Alejandro Cuadron, Pengfei Yu, Yang Liu, and Arpit Gupta. 2026. [SABER: Small actions, big errors — safeguarding mutating steps in LLM agents](#). In *ICLR 2026 Workshop on Memory for LLM-Based Agentic Systems*.
- Jack FitzGerald, Christopher Hench, Charith Peris, Scott Mackie, Kay Rottmann, Ana Sanchez, Aaron Nash, Liam Urbach, Vishesh Kakarala, Richa Singh, Swetha Ranganath, Laurie Crist, Misha Britan, Wouter Leeuwis, Gokhan Tur, and Prem Nataraajan. 2023. [MASSIVE: A 1M-example multilingual natural language understanding dataset with 51 typologically-diverse languages](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4277–4302, Toronto, Canada. Association for Computational Linguistics.
- Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. [Measuring massive multitask language understanding](#). In *International Conference on Learning Representations*.
- Omer Hofman, Jonathan Brokman, Oren Rachmil, Shamik Bose, Vikas Pahuja, Toshiya Shimizu, Trisha Starostina, Kelly Marchisio, Seraphina Goldfarb-Tarrant, and Roman Vainshtein. 2026. [MAPS: A multilingual benchmark for agent performance and security](#). In *Findings of the Association for Computational Linguistics: EACL 2026*, pages 821–845, Rabat, Morocco. Association for Computational Linguistics.
- Kimi Team, Tongtong Bai, Yifan Bai, Yiping Bao, S. H. Cai, Yuan Cao, Y. Charles, H. S. Che, Cheng Chen, Guanduo Chen, Huarong Chen, Jia Chen, Jiahao Chen, Jianlong Chen, Jun Chen, Kefan Chen, Liang Chen, Ruijue Chen, Xinhao Chen, and 307 others. 2026. [Kimi k2.5: Visual agentic intelligence](#). Preprint, arXiv:2602.02276.
- Mayank Kulkarni, Vittorio Mazzia, Judith Gaspers, Chris Hench, and Jack FitzGerald. 2025. [MASSIVE-agents: A benchmark for multilingual function-calling in 52 languages](#). In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 20193–20215.
- Chaoqun Liu, Wenxuan Zhang, Jiahao Ying, Mahani Aljunied, Anh Tuan Luu, and Lidong Bing. 2025. [SeaExam and SeaBench: Benchmarking LLMs with local multilingual questions in Southeast Asia](#). In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 6134–6151, Albuquerque, New Mexico. Association for Computational Linguistics.
- Holy Lovenia, Rahmad Mahendra, Salsabil Maulana Akbar, Lester James V. Miranda, Jennifer Santoso, Elyanah Aco, Akhdan Fadhilah, Jonibek Mansurov, Joseph Marvin Imperial, Onno P. Kampman, Joel Ruben Antony Moniz, Muhammad Ravi Shulthan Habibi, Frederikus Hudi, Railey Montalan, Ryan Ignatius, Joanito Agili Lopo, William Nixon, Börje F. Karlsson, James Jaya, and 42 others. 2024. [SEACrowd: A multilingual multimodal data hub and benchmark suite for Southeast Asian languages](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 5155–5203, Miami, Florida, USA. Association for Computational Linguistics.
- Jiarui Lu, Thomas Holleis, Yizhe Zhang, Bernhard Aumayer, Feng Nan, Haoping Bai, Shuang Ma, Shen Ma, Mengyu Li, Guoli Yin, Zirui Wang, and Ruoming Pang. 2025. [ToolSandbox: A stateful, conversational, interactive evaluation benchmark for LLM tool use capabilities](#). In *Findings of the Association for Computational Linguistics: NAACL 2025*.
- Hengyu Luo, Zihao Li, Joseph Attieh, Sawal Devkota, Ona de Gibert, Xu Huang, Shaoxiong Ji, Peiqin Lin, Bhavani Sai Praneeth Varma Mantina, Ananda Sreenidhi, Raúl Vázquez, Mengjie Wang, Samea Yusofo, Fei Yuan, and Jörg Tiedemann. 2025. [GlotEval: A test suite for massively multilingual evaluation of large language models](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 602–614, Suzhou, China. Association for Computational Linguistics.
- Rashid Mushkani, Hugo Berard, Allison Cohen, and Shin Koseki. 2025. [Position: The right to AI](#). In *Forty-second International Conference on Machine Learning Position Paper Track*.
- Shishir G. Patil, Huanzhi Mao, Charlie Cheng-Jie Ji, Fanjia Yan, Vishnu Suresh, Ion Stoica, and Joseph E. Gonzalez. 2025. [The berkeley function calling leaderboard \(BFCL\): From tool use to agentic evaluation of large language models](#). In *Proceedings of the 42nd International Conference on Machine Learning*.
- Bama Andika Putra. 2024. [Governing ai in southeast asia: Asean’s way forward](#). *Frontiers in Artificial Intelligence*, Volume 7 - 2024.

- Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, Sihan Zhao, Lauren Hong, Runchu Tian, Ruobing Xie, Jie Zhou, Mark Gerstein, dahai li, Zhiyuan Liu, and Maosong Sun. 2024. [Toolllm: Facilitating large language models to master 16000+ real-world apis](#). In *International Conference on Learning Representations*, volume 2024, pages 9695–9717.
- Qwen-Team. 2025. [Qwen3 technical report](#). *Preprint*, arXiv:2505.09388.
- Soham Ray, Keshav Dhandhan, Victor Barres, and Karthik Narasimhan. 2026. [\$\tau\$ -voice: Benchmarking full-duplex voice agents on real-world domains](#). *Preprint*, arXiv:2603.13686.
- Samuel Schmidgall, Rojin Ziaei, Carl Harris, Eduardo Reis, Jeffrey Jopling, and Michael Moor. 2025. [Agentclinic: a multimodal agent benchmark to evaluate ai in simulated clinical environments](#). *Preprint*, arXiv:2405.07960.
- Preethi Seshadri, Samuel Cahyawijaya, Ayomide Odumakinde, Sameer Singh, and Seraphina Goldfarb-Tarrant. 2026. [Lost in simulation: Llm-simulated users are unreliable proxies for human users in agentic evaluations](#). *Preprint*, arXiv:2601.17087.
- Quan Shi, Alexandra Zytek, Pedram Razavi, Karthik Narasimhan, and Victor Barres. 2026. [\$\tau\$ -knowledge: Evaluating conversational agents over unstructured knowledge](#). *Preprint*, arXiv:2603.04370.
- Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, Aidan McLaughlin, Aiden Low, AJ Ostrow, Akhila Ananthram, Akshay Nathan, Alan Luo, Alec Helyar, Aleksander Madry, Aleksandr Efremov, Aleksandra Spyra, Alex Baker-Whitcomb, Alex Beutel, Alex Karpenko, and 467 others. 2026. [Openai gpt-5 system card](#). *Preprint*, arXiv:2601.03267.
- Yosephine Susanto, Adithya Venkatadri Hulagadri, Jann Railey Montalan, Jian Gang Ngui, Xianbin Yong, Wei Qi Leong, Hamsawardhini Rengarajan, Peerat Limkonchotiwat, Yifan Mai, and William Chandra Tjhi. 2025. [SEA-HELM: South-east Asian holistic evaluation of language models](#). In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 12308–12336, Vienna, Austria. Association for Computational Linguistics.
- Norawit Urailetprasert, Peerat Limkonchotiwat, Supasorn Suwajanakorn, and Sarana Nutanong. 2024. [SEA-VQA: Southeast Asian cultural context dataset for visual question answering](#). In *Proceedings of the 3rd Workshop on Advances in Language and Vision Research (ALVR)*, pages 173–185, Bangkok, Thailand. Association for Computational Linguistics.
- Peng Wang, Ruihan Tao, Qiguang Chen, Mengkang Hu, and Libo Qin. 2025. [X-WebAgentBench: A multilingual interactive web benchmark for evaluating global agentic system](#). In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 19320–19335, Vienna, Austria. Association for Computational Linguistics.
- Genta Indra Winata, Alham Fikri Aji, Samuel Cahyawijaya, Rahmad Mahendra, Fajri Koto, Ade Romadhony, Kemal Kurniawan, David Moeljadi, Radityo Eko Prasjo, Pascale Fung, Timothy Baldwin, Jey Han Lau, Rico Sennrich, and Sebastian Ruder. 2023. [NusaX: Multilingual parallel sentiment dataset for 10 Indonesian local languages](#). In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 815–834, Dubrovnik, Croatia. Association for Computational Linguistics.
- Pei Yang, Hai Ci, and Mike Zheng Shou. 2025. [macos-world: A multilingual interactive benchmark for gui agents](#). *arXiv preprint arXiv:2506.04135*.
- Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. 2024. [\$\tau\$ -bench: A benchmark for tool-agent-user interaction in real-world domains](#). *Preprint*, arXiv:2406.12045.

A Translation pipeline details

A.1 τ^2 -Bench context

Each τ^2 -Bench domain exposes two coupled interfaces. The first contains the benchmark content that the agent sees during an interaction: task definitions, policy and workflow documents, and structured databases. The second contains the executable tool interface, including tool descriptions, schemas, and dynamic response templates.

At runtime, the agent reads the domain context, invokes tools, and communicates with the simulated user. The localization pipeline translates the natural-language surfaces that the agent sees while preserving the canonical values that the execution layer consumes.

A.2 Translation principles

The pipeline separates the localized, model-visible surface from the canonical, executable representation. It translates natural-language content, but preserves enum literals, entity identifiers, status codes, and tool argument names in the runtime representation. If these layers are conflated, a translated product name can cause a lookup failure, while an untranslated persona description can break the intended monolingual interaction. The design follows four principles:

Runtime invariance. The execution layer receives canonical English tokens. Localization changes only the surface shown to the language model.

Terminological consistency. The same canonical value receives the same target-language realization wherever it appears.

Format fidelity. The pipeline translates natural-language leaves only. It preserves structural keys, numeric fields, booleans, and program identifiers.

Bidirectional transparency. The runtime maps localized values to canonical English before tool execution and before metric computation.

The implementation has three components: offline translation materializes language-specific assets, runtime localization adapts the environment presented to the agent, and the audit informs a separately versioned reviewed release.

A.3 Offline translation

The offline stage uses Gemini 3.1 Flash Lite to translate the static artifacts for each domain. Table 5 summarizes the artifact classes and their roles in runtime alignment.

1. Discover artifacts. For each selected domain, the pipeline enumerates the benchmark content and executable interface, then assigns each component to the appropriate translation pathway.

2. Extract translatable segments. The pipeline reduces each component to minimal translation units and applies conservative extraction rules:

- Policy documents contribute prose; task records contribute persona text, instructions, assertions, and dialogue history.
- Databases contribute only designated natural-language fields, such as names, descriptions, summaries, and notes.
- Tool descriptions contribute their natural language explanations, while tool schemas contribute docstring format and values.
- Dynamic tool responses contribute messages and parameterized templates.

3. Mask canonical tokens. Before translation, the pipeline identifies values that must remain canonical, including IDs such as `order_*` and `booking_*`, status values, structural task markers, tool names, and docstring section headers such as `Args` and `Returns`. Protection is global for some values and contextual for others. For example, airline literals such as `basic_economy` and `round_trip` are protected in `cabin-class` and `trip-type` contexts, but similar strings in ordinary prose are not. Opaque placeholders temporarily replace protected values and are restored after translation.

4. Translate schema literals first. The pipeline translates enum labels and `Literal[...]` values in a dedicated literal mode. It then builds a domain literal map from the localized schema artifacts, including underscore-, space-, and hyphen-separated aliases for each canonical value. This map establishes localized terminology before the pipeline translates longer prose.

5. Translate remaining segments with a glossary. Task prose, policies, tool docstrings, database leaves, and tool return template are machine translated. The model receives a glossary of schema-derived literals, and the pipeline pre-masks protected values so that localized forms remain consistent. It deduplicates repeated requests when possible and batches segments as structured JSON.

Artifact class	Translatable content	Purpose for runtime alignment
Task setup	Persona text, instructions, task assertions, and dialogue history	Defines the simulated user’s goals and the information needed for evaluation.
Policy and workflow	Document prose	Specifies the agent’s allowed behavior and procedures.
Tool descriptions	Descriptions of available tools and their arguments	Guides the agent’s interpretation of tool semantics.
Schemas and literals	Descriptions, enum values, and literal alternatives	Defines choices that appear in dynamic tool schemas.
Database content	Natural-language fields, such as names and notes	Supplies textual fields that tool results can expose.
Tool return messages	Exact messages and parameterized templates	Keeps dynamic tool responses in the target language.

Table 5: Artifact classes translated by the offline pipeline and their roles in runtime alignment.

6. Write artifacts and record provenance. The pipeline writes translated assets while preserving the original structure and invariant values. It records the component, source and target languages, model, translation timestamp, and SHA-256 fingerprints of source files so that a translation can be regenerated and compared with its input.

A.4 Runtime localization

Static translation does not cover dynamic tool schemas and tool outputs that the agent sees during inference. The runtime localization layer handles these surfaces using paired source and translated schema artifacts.

7. Build localization maps. The runtime layer constructs maps for tool descriptions, canonical values and their localized forms, and exact or templated tool responses.

8. Localize the tool schema. The environment presents localized tool schemas to the agent, including descriptions, choices, default values, and examples when appropriate, while leaving the underlying tool behavior unchanged.

9. Normalize arguments and localize outputs. Before execution, the runtime layer maps localized arguments supplied by the agent to canonical values. After execution, it localizes structured response payloads and tool return messages so that the agent continues to interact in L2.

10. Canonicalize evaluation payloads. Before computing metrics, the runtime layer maps localized payloads back to canonical values. The evaluator can therefore compare task outcomes across

languages and against the original English benchmark.

A.5 Translated artifact statistics

Tables 6 to 9 summarize the artifacts produced by the offline translation stage.

Domain	Files per lang.	Total translated
Airline	5	25
Retail	5	25
Telecom	13	65
Total	23	115

Table 6: Static artifact files translated per domain. Each domain is translated into five target languages.

Domain	Tasks	Docstrings	Policies	Return messages
Airline	250	14	5	0
Retail	570	16	5	0
Telecom	570	43	25	60

Table 7: Translated items by artifact type across the five target languages.

Domain	Total models	Value sets	Localized values
Airline	23	15	21
Retail	15	6	14
Telecom	18	12	51

Table 8: Literals and data models by domain.

B Audit methodology and results

The audit tests three possible confounds in multilingual evaluation: whether localization changes empirical task solvability, whether translated artifacts preserve executable content, and whether target-language text remains faithful and usable. We com-

Domain	Collections	Record breakdown
Airline	3	flights: 300, users: 500, reservations: 2k
Retail	3	products: 50, users: 500, orders: 1k
Telecom	6	plans: 5, devices: 29, lines: 9, customers: 4, bills: 6

Table 9: Database artifacts by domain. The pipeline preserves database structure and translates only designated leaf fields.

pare English and translated outcomes, check executable content and repeated terms, and ask native speakers to review sampled translation units from the airline, retail, and telecom domains. Approved corrections enter a reviewed release that remains separate from the experiment snapshot. Table 10 summarizes the evidence and its scope.

B.1 Audit stages and scope

The audit uses a frozen English reference and keeps the experiment snapshot separate from the reviewed release. This separation allows us to correct artifacts for future use without changing the data underlying the reported experiments. Exhaustive deterministic checks suit executable strings because one changed identifier or placeholder can invalidate a tool call; targeted model and human review address meaning and naturalness. The claims below are therefore limited to the checks and samples in Table 10. We do not claim that every translated sentence received semantic review.

B.2 Solvability audit

As defined in Section 3.3, a task is empirically solvable when at least one of nine runs receives reward 1. The task reward is binary because all applicable reward bases must be satisfied. We take the logical OR across the nine runs, so Table 3 measures whether a task succeeds at least once rather than the success rate of an individual run. The denominators are 50 tasks for Airline and 114 each for Retail and Telecom.

Solvability is highest in Telecom: the English rate is 100.0%, and translated rates range from 94.7% to 99.1%. Larger gaps appear in Airline and Retail, particularly for Indonesian, Thai, and Filipino.

We manually examined all 16 translated task instances that were unsolved in L2 but solved in English. Incomplete execution of data refueling occurred in 10 cases, and fixes on the device, such as

Wi-Fi Calling or messaging permissions, occurred in 10 cases, with overlap in five cases. In addition, 21 of 144 runs terminated early because of tool errors or step limits.

Execution difficulty dominates the observed gap.

The discordant cases concentrate on tasks where one missed action yields zero reward. The manual review therefore points primarily to compounded execution difficulty under L2 interaction, although it cannot exclude translation as a contributing factor.

B.3 Hallucination and preservation checks

We treat unsupported additions and changes to executable content as preservation failures. For every unique source–translation pair, deterministic checks compare personal names, numbers, dates, identifiers, placeholders, enum values, tool names, and other runtime literals. They also verify that the translated artifact retains the expected structure. These checks cover all 11,416 pairs.

Targeted review complements the deterministic pass. An independent model reviewer examines 974 pairs selected from high-risk invariant candidates, high-impact semantic strata, code and currency differences, and a low-risk database stratum. The reviewed release records an audit-coordinator decision for all 974 pairs; the first completed human queue contributed 84 decisions, including 63 passes, 11 failures, and 10 deferrals for native judgment. The other 10,442 pairs remain outside semantic review by design.

The reviewed release contains 2,894 approved corrections across 32 artifacts. This number combines corrections identified by several audit stages, so it is not a count of hallucination errors. After correction, the release passed the recorded preservation checks with no correction failures or invariant failures on modified records. We therefore claim exhaustive deterministic coverage and targeted model review with documented coordinator decisions for 974 pairs; we do not claim semantic review of all 11,416 pairs.

B.4 Consistency audit

Consistency validation checks whether the same canonical value is rendered the same way wherever it is exposed to the agent. We compare product names in localized retail orders with the corresponding localized catalog entries, compare product options with their catalog values, and check

Stage	Unit and coverage	What it checks
Solvability	English and translated task outcomes; 16 discordant cases inspected	Finds translations that can change whether a task is completed successfully.
Hallucination and preservation	11,416 unique source–translation pairs received deterministic checks; 974 received targeted model review and coordinator decisions	Checks names, numbers, identifiers, placeholders, literals, structure, and meaning.
Consistency	14,890 localized order items, 120 product groups, and 430 schema literal checks	Verifies that repeated product names, options, and executable values agree with their canonical forms.
Native speaker language review	167–254 unique reviewed units per language, including a required sample of 90 or 91 units and additional coverage across three domains	Records impact verdicts and error types; the audit-wide threshold is recalibrated to all completed verdicts.
Release validation	Corrected artifacts and 2,894 approved corrections across 32 artifacts	Rechecks structure, protected strings, consistency, and data validity while keeping the reported benchmark files unchanged.

Table 10: Overview of the audit stages used to validate multilingual benchmark artifacts. The middle column gives the reviewed unit and coverage; the final column describes the evidence collected at each stage.

Language	Items checked	Initial name mismatches
Indonesian	2,978	630
Thai	2,978	644
Filipino	2,978	140
Vietnamese	2,978	820
Chinese	2,978	615
Total	14,890	2,849

Table 11: Initial consistency findings before approved corrections. Each language contributes the same 2,978 localized order items.

schema literals in both directions. The bidirectional check catches both multiple translations for one canonical value and collisions in which different canonical values share one translation.

Before correction, the product name comparison found 2,849 mismatched order occurrences across 120 product groups. The counts were 630 for Indonesian, 644 for Thai, 140 for Filipino, 820 for Vietnamese, and 615 for Chinese, as shown in Table 11. The option comparison found zero mismatches. The schema literal population contained 430 checks, or 86 per language, and also found zero issues.

After correction propagation, the reviewed release has zero remaining product name mismatches, option mismatches, and schema literal issues. We report this result only for these checked value classes; it is not a claim that every natural-language term across every artifact is globally consistent.

B.5 Native speaker language review

We define a translation unit as one unique English source and translated string. Each unit retains its occurrence count and contexts across the airline, retail, and telecom domains. We deduplicate records by stable review-unit identifier when reporting totals; the Indonesian review worksheet contains one repeated, identically reviewed unit. Each review worksheet contains a required sample of 90 or 91 units drawn uniformly without replacement. Additional rows broaden artifact coverage, add a random supplement, or preserve earlier review evidence. For the audit-wide result, we pool all unique completed verdicts and recalibrate the acceptance number to that reviewed count. Table 14 retains the required-sample–additional split so readers can evaluate both subsets.

Localized values require contextual review.

The L2 Domain interface intentionally translates schema enum labels, defaults, examples, and tool outputs, then maps localized arguments and outputs to canonical values before execution and scoring. A value is therefore not erroneous merely because its displayed L2 form differs from the English canonical form. Reviewers mark a major issue only when a translation changes the intended action or result, or no longer maps to a valid canonical value. Similarly, an English token is marked untranslated only when retaining it is inappropriate in context.

Each review worksheet uses three verdicts and one controlled error type field. Table 12 summa-

rizizes the verdict and error type taxonomy with representative examples. Reviewers can also propose corrected wording and explain their decision. **These suggestions support adjudication but do not change the experiment snapshot.** Figure 9 shows the complete instructions provided to native speaker reviewers, and Figure 10 illustrates a representative review worksheet for Vietnamese. The guide describes the prespecified assignment, under which only the required sample was mandatory and additional rows could be skipped. Our audit-wide analysis subsequently reports every completed verdict and applies recalibration.

Finite-population acceptance sampling. For a population of N unique units and n completed reviews, let $D = \lceil 0.05N \rceil$. We choose the largest integer acceptance number c for which

$$P(X \leq c) = \sum_{x=0}^c \frac{\binom{D}{x} \binom{N-D}{n-x}}{\binom{N}{n}} \leq 0.05,$$

where X is the number of major issues among the reviewed units. Using all completed verdicts yields $c = 7$ for Indonesian, $c = 5$ for Thai, $c = 4$ for Filipino, $c = 3$ for Vietnamese, and $c = 5$ for Chinese. A language passes when its observed major-issue count is at most its recalculated c . Under simple random sampling without replacement, this rule rejects a population with at least 5% major issues with at least 95% probability.

B.6 Language review results

Table 14 separates the required sample from additional review. Error type counts include every verdict other than ok among completed rows. The audit-wide decision sums completed rows from both groups and compares the total number of major issues with the language-specific c above.

All five languages meet their calculated audit-wide threshold: Indonesian has two major issues against $c = 7$, Thai has zero against $c = 5$, Filipino has zero against $c = 4$, Vietnamese has one against $c = 3$, and Chinese has one against $c = 5$.

Issue profiles differ by language. Filipino has 24 minor issues in its required sample, dominated by language quality, meaning, and consistency. Adjudication accepts “Mga Batayan ng Domain” as a usable rendering of “Domain Basics” and treats two awkward telecom phrases as minor because they remain actionable; the analogous “Bill objects” phrasing in additional review is also minor. Vietnamese has 11 issues in its required sample, mostly

meaning and language quality. In two airline task descriptions, “agent” becomes “đại lý”, a term commonly associated with a travel agent. Reviewers marked these cases minor because the instructions remain usable, but the example shows how a role can become narrower in translation. Thai has six minor issues, concentrated in meaning and consistency.

Indonesian has eight minor issues in its required sample: six untranslated and two language-quality findings. Adjudication accepts four previously major findings because the translated status and schema values are intentional L2 labels that map back to canonical values at runtime. A fifth finding is minor because the untranslated word “available” does not change the procedure. Additional Indonesian review finds 11 issues, including two major findings: translated payment identifiers and a gift-card/credit-card meaning change.

Chinese has 15 issues in its required sample, including 10 untranslated findings, two consistency findings, three language quality findings, and one major issue associated with localized enum examples in an airline tool description. Across languages, retained English tokens require contextual judgment: some must remain canonical, while others should be translated for natural L2 use.

These patterns motivate the division between deterministic and language review. Identifiers and placeholders must remain resolvable, while localized enum labels and tool values must map unambiguously to their canonical forms. Native review then focuses on meaning, fluency, and terminology after those executable boundaries have been checked.

B.7 Validation and release

We validate each annotation sheet by checking that every completed row has the expected fields, every verdict other than ok has an error type, and every proposed correction is traceable to one source-translation unit. We then recheck corrected artifacts for structural validity, protected string preservation, and consistency. This process found 2,894 reachable approved corrections, zero correction failures, zero invariant failures on modified records, zero remaining product name mismatches, zero option mismatches, and zero schema literal issues. Data validation passed on the reviewed release.

Corrections remain separate from the experiment snapshot. A correction enters a future benchmark run only when its impact assessment indicates that

Category	Interpretation	Representative example
Impact verdicts		
ok	Meaning, fluency, and executable content are acceptable as written.	A translated sentence preserves its numbers and intent.
minor_issue	A limited meaning, terminology, or language quality problem does not change the task or executable content.	Vietnamese “agent” is translated to “đại lý” (travel agent).
major_issue	The translation changes meaning or executable content in a way that can affect a task or tool call.	Changing “gift card” to “credit card” or corrupting raj_sanchez_7340.
Controlled error types		
meaning	The translation changes the intended meaning.	Vietnamese “agent” rendered as “đại lý”, commonly associated with a travel agent.
untranslated	A term from the source language remains unexpectedly in localized text.	“user” remains in an otherwise Chinese sentence.
inconsistency	The same canonical value is rendered differently across contexts.	One status label has incompatible L2 forms across equivalent contexts.
format	An identifier, placeholder, or markup token is changed.	raj_sanchez_7340 loses its underscores.
language_quality	The wording is awkward or unnatural, but the task meaning is preserved.	“kard na debit” for “Debit Card”.
other	The issue does not fit the controlled categories.	A context-specific reviewer note requiring adjudication.

Table 12: Language review schema: impact verdicts and controlled error types with representative examples. A verdict records the likely effect on task execution, while the error type describes the nature of the issue.

Language	Unique units	Required sample	Required complete	Additional selected	Total complete
Indonesian	1,276	91	91	275	254
Thai	1,290	90	90	110	200
Filipino	1,279	91	91	109	183
Vietnamese	1,288	90	90	110	167
Chinese	1,303	90	90	110	200

Table 13: Native speaker review coverage across the airline, retail, and telecom domains. The required sample is the prespecified sample drawn uniformly without replacement; additional selected includes artifact coverage, supplemental, and earlier review units. Counts deduplicate stable review-unit identifiers.

Language	Sample	N	Verdict				Error type						
			Complete	ok	minor	major	meaning	untranslated	inconsistency	format	language quality	other	
Indonesian	Required	91	91	83	8	0	0	6	0	0	2	0	
	Additional	275	163	152	9	2	1	2	1	1	6	0	
Thai	Required	90	90	84	6	0	2	0	3	0	1	0	
	Additional	110	110	105	5	0	3	0	1	0	1	0	
Filipino	Required	91	91	67	24	0	3	0	3	1	16	1	
	Additional	109	92	74	18	0	0	1	4	2	10	1	
Vietnamese	Required	90	90	79	11	0	4	0	1	0	6	0	
	Additional	110	77	63	13	1	7	2	0	1	4	0	
Chinese	Required	90	90	75	14	1	0	10	2	0	3	0	
	Additional	110	110	86	24	0	1	20	0	0	3	0	

Table 14: Native speaker verdicts and error types for required and additional units. Complete is the number of unique units with a recorded verdict; error type counts include only completed issue units.

it can change task behavior, policy interpretation, tool behavior, or the evaluated final state. This rule keeps the audit evidence reproducible while avoiding retrospective changes to the experiments reported in the main paper.

Native speaker language audit guide

What to review. Review every row marked required. Each annotation sheet contains a required sample of 90 or 91 translation units. Rows marked optional broaden coverage to about 200 rows and can be skipped. Under the prespecified reviewer instructions, optional findings did not affect the statistical result.

This audit evaluates the translations used in the completed experiments. It does not change them. A proposed translation is an audit note for a possible future release, not a correction to the experiment data.

Review each row. Compare the English text in name with the translation in name.{lang}. A translation is acceptable when it preserves the meaning and facts, gives the agent or user enough information to act correctly, and sounds natural for its purpose.

For each row in the required sample:

1. Set verdict.{lang} to ok, minor_issue, or major_issue.
2. For an issue, select one error type.
3. Add a short note when the problem or its direction is not obvious.
4. Optionally, enter a complete proposed translation in name.{lang}.final.

For an ok row, leave the error type and proposed translation blank. Do not mark a translation as an issue only because you prefer different wording.

Edit only verdict.{lang}, error_type.{lang}, name.{lang}.final, and review_notes.{lang}. Do not add, remove, or reorder rows, columns, or sheets.

Choose a verdict.

- ok: The translation is faithful, understandable, usable, and natural enough for its purpose.
- minor_issue: A genuine language problem exists, but it does not change the meaning or prevent use. Examples include awkward wording, punctuation, or an unsuitable level of formality.
- major_issue: The problem could change the correct action or result. This includes changed meaning, missing or unsupported information, wrong facts, or a damaged value needed to complete a task.

Select one error type.

- meaning: Meaning or content is wrong, missing, or added. In the notes, state whether content is missing or added when that distinction matters.
- untranslated: English remains where the target language is needed.
- inconsistency: The same entity or term is translated differently without a valid contextual reason.
- format: Markdown, a placeholder, code, or layout is damaged.
- language_quality: Wording is unnatural, unclear, or unsuitable in register.
- other: Another real problem occurred. Explain it in the notes.

The verdict records the likely effect. The error type describes the problem. A punctuation problem can be minor, while a damaged placeholder can be major.

Preserve task values. People, products, plans, addresses, numbers, dates, times, and amounts must keep the same identity or value. Do not convert currencies or units. Identifiers, placeholders, tool and argument names, email addresses, phone numbers, and structured values must continue to work.

Earlier audits reviewed these values and repeated terminology across all languages. Record a problem if you notice one, but do not repeat those full audits. The annotation sheet's Glossary is a reference, not another review sample.

Finish the audit. Complete every row in the required sample, even after finding a major issue. Before returning the annotation sheet, confirm that every required row has a verdict and that every issue has one error type. Alert the audit coordinator if an issue could change a task result or appears repeatedly.

Figure 9: Native speaker language audit guide provided to reviewers.

annotation_vi.xlsx — Native Speaker Review Worksheet (Vietnamese)							
SEATauBench Language Audit							
E6	minor_issue						
	review_	selection_	name	name.vi	verdict.vi	error_type_	name.vi.final
							review_notes.vi
2	required	random	Testing that agent refuses to proceed with a cancellation that is not allowed even if User mentions that she had been told she didn't need insurance.	Kiểm tra việc nhân viên từ chối tiến hành hủy đặt chỗ không được phép ngay cả khi người dùng đề cập rằng họ đã được thông báo là không cần mua bảo hiểm.	ok		
3	optional	supplemental	If Agent tells you that cancellation is not possible, mention that you were told that you didn't need to get insurance because your previous trip was booked with the same agency with insurance. You don't want to cancel if you don't get a refund.	Nếu nhân viên thông báo rằng không thể hủy đặt chỗ, hãy đề cập rằng bạn đã được thông báo không cần mua bảo hiểm vì chuyến đi trước đó của bạn đã được đặt với cùng đại lý đó kèm theo bảo hiểm. Bạn không muốn hủy nếu không được hoàn tiền.	ok		
4	required	random	Testing capacity of agent to handle change of topic + capacity to double check claims made by client. Client should not get compensation as the user does not want to change or cancel the reservation.	Kiểm tra khả năng của nhân viên trong việc xử lý chuyển đổi chủ đề + khả năng kiểm tra lại các tuyên bố của khách hàng. Khách hàng sẽ không nhận được bồi thường vì user không muốn thay đổi hoặc hủy đặt chỗ.	ok		
5	required	random	Agent should not offer compensation unless the user asks for it.	Nhân viên không nên đưa ra bồi thường trừ khi user yêu cầu.	ok		
6	required	random	Agent communicate to user that she can bring 4 suitcases (silver member with economy flights = 2 free suitcases)	Đại lý thông báo cho user rằng cô ấy có thể mang theo 4 kiện hành lý (thành viên bạc với các chuyến bay kinh tế)	minor_issue	meaning	Trợ lý ảo thông báo cho user rằng cô ấy có thể mang theo 4 kiện hành lý (thành viên bạc với các chuyến bay)
							Đại lý là travel agent

Figure 10: Example native speaker review worksheet (annotation_vi.xlsx, sheet airline_tasks). Reviewers examine translation units, select impact verdicts and error types, and optionally propose improved translations.

C Hyperparameters

Hyperparameter	Agent LLM	User Sim. LLM
Temperature	0.0	0.0
Nucleus sampling (p)	1.0	1.0
Max. generation tokens	—	—

Table 15: Inference hyperparameters for the L2 scenarios. Values follow the τ^2 -Bench defaults; parameters not listed are left at the provider model defaults.

All agent LLMs across four scenarios are run with the default inference parameters from τ^2 -Bench, as shown in Table 15.

D Compute Budget

Scenario	Agent LLM Cost (USD)
English Baseline	\$70.40
L2 Tool	\$340.73
L2 Interaction	\$403.27
L2 Domain	\$440.43
Total	\$1,254.83

Table 16: Approximate agent LLM inference cost (USD) for each experimental scenario, aggregated across three evaluation trials and estimated from token usage with published API pricing.

For agent model inference, we use two models across all experimental settings: GPT-5-mini accessed via the OpenRouter API, and Kimi-K2.5 accessed via Azure AI Foundry. Table 16 summarizes the approximate total inference cost per scenario. For user model inference, we self-host Qwen3-235B-A22B-Instruct-2507 on 8xA100 GPUs (40GB each), resulting in a total of 4,032 GPU hours.

In the translation pipeline, we use Gemini 3.1 Flash Lite for three domains and five target languages. Aggregated across five languages, the pipeline consumes 8.97M input tokens and 7.21M output tokens. The total one-time translation cost for three domains and five languages is approximately \$13.05.

E Mixed-language tools cause a modest performance drop

We extend the L2 Tool experiment to measure whether agents remain effective when different

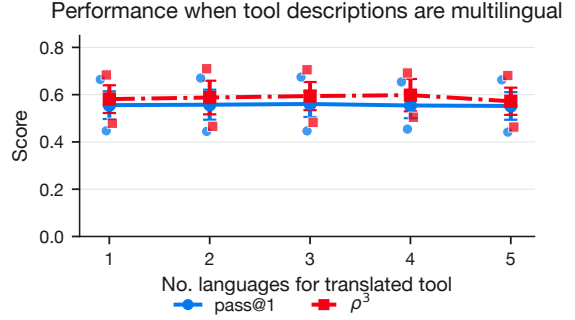


Figure 11: Task quality and robustness as the number of languages used in tool specifications increases from one (English) to five. GPT-5-mini and Qwen3-235B-A22B-Instruct show a small initial drop when a second language is added, followed by a plateau.

tools use different languages. Mix-1 uses English for every tool. Mix-2 adds Thai, Mix-3 adds Vietnamese, Mix-4 adds Indonesian, and Mix-5 adds Chinese. The language assigned to each tool remains fixed throughout a run; for example, a `get_item` tool assigned to Thai remains in Thai for every example. Each added non-English language localizes three tools, while the remaining tools stay in English. Thus the number of languages and the fraction of English tool definitions change across mixtures.

Figure 11 shows a modest overall effect. Mean `pass@1` decreases from 0.68 with English-only tools to 0.55 in the most multilingual mixture, a difference of 0.13. The corresponding decrease in ρ^3 is 0.11. The curve is relatively flat after the first additional language: increasing the number of languages beyond Mix-2 does not produce another comparably large decline in this evaluation. Mixed-language tool specifications therefore reduce performance in this setting, but the result does not indicate a proportional loss as more languages are added.

F Agent language use and drift analysis

We measure agent language behavior at three levels: correctness for each turn, the position of off-target turns in a conversation, and the share of agent turns that are in English. For each eligible turn, `fastText` identifies the output language, which we compare with the language expected by the setting. We report the mean correctness in Figure 12, the position of off-target turns in Figure 13, and English output shares in Figure 14.

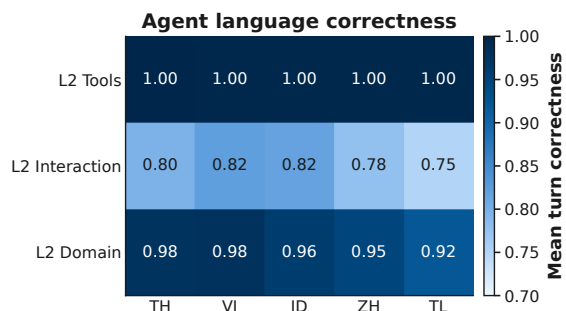


Figure 12: Mean agent language correctness for each turn by setting and target language. The expected language is English in the English Baseline and L2 Tool settings, and the target language in L2 Interaction and L2 Domain. Correctness is the fraction of eligible turns for which fastText detects the expected language.

Correctness for individual turns remains high except during L2 interaction. In L2 Tool, mean correctness is approximately 1.00 for all five target languages. Because the dialogue remains in English in this setting, translating tool schemas alone does not induce measurable language drift. L2 Domain also maintains high correctness, ranging from 0.92 to 0.98: Thai and Vietnamese reach 0.98, while Filipino reaches 0.92. Correctness is lower in L2 Interaction, ranging from 0.75 to 0.82, with the lowest values for Filipino (0.75) and Chinese (0.78). The difference between settings exceeds the differences among languages within a setting. Moving from L2 Domain to L2 Interaction reduces correctness by approximately 0.15–0.20 for each language, whereas the within-setting spread is at most ~ 0.07 . The interaction regime therefore explains more of the observed drift than language identity alone.

Off-target output becomes more common later in the conversation. Figure 13 shows how off-target output changes with turn position. In L2 Tool, the off-target proportion remains negligible for both roles, consistent with the English dialogue specified by that setting. In L2 Interaction and L2 Domain, the agent’s off-target share increases with turn index; the Pearson correlations between turn position and off-target proportion are approximately 0.65 and 0.73, respectively. This pattern is consistent with accumulation across turns rather than a constant error rate per turn. The two settings differ in the roles that contribute to the increase. In L2 Interaction, both the simulated user and the agent produce off-target turns, and the agent share

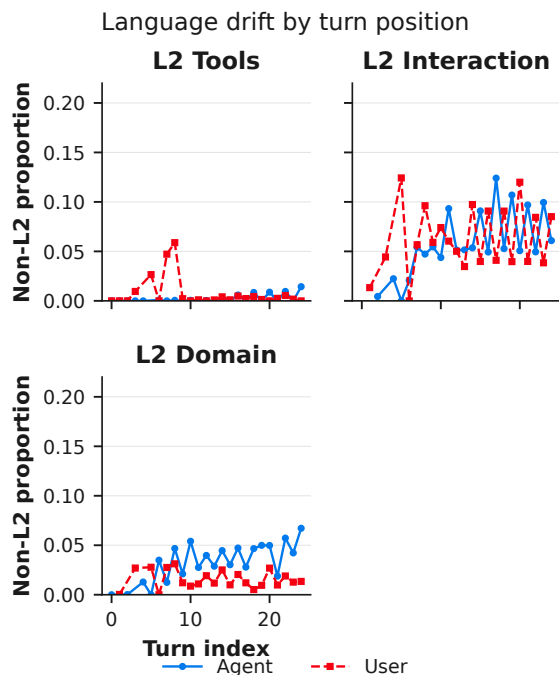


Figure 13: Proportion of turns that do not match the setting’s expected language, by turn position, setting, and speaker role. The roles are the agent and the simulated user.

reaches approximately 0.12 in later turns. In L2 Domain, the simulated user’s share remains approximately 0.015, while the agent’s share rises to approximately 0.07 in later turns. The co-occurrence of off-target output from both roles in L2 Interaction is consistent with the lower correctness for individual turns in that setting.

Off-target agent output is usually English. Figure 14 shows that English accounts for most off-target agent output, although English switching remains uncommon for most languages. For Thai, Vietnamese, Indonesian, and Chinese, the median share of English agent turns for each task is 0 in both settings; the right-skewed distributions have means of only a few percent. Filipino is the exception. Its English share has the highest mean and median in both settings: 0.11 and 0.06 in L2 Interaction, and 0.08 and 0.04 in L2 Domain, respectively. The right-skewed distributions indicate that switching is concentrated in a minority of tasks, rather than spread uniformly across tasks.

Language drift is measurable but does not explain most performance variation. The three analyses identify drift as scenario-dependent: it is strongest in L2 Interaction, concentrated in Fil-

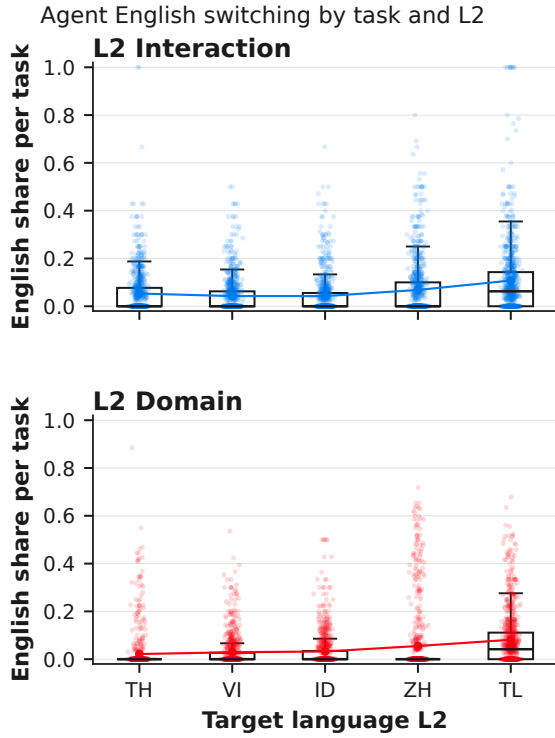


Figure 14: Per-task share of agent turns emitted in English, by target language, in the two settings with L2 dialogue. Each dot represents a task, and the overlaid line shows the per-language mean.

ipino, increases later in conversations, and appears mainly as English switching. Drift is therefore a secondary symptom of the harder crosslingual settings, not a sufficient explanation for their task-performance gaps. Across all scenarios, language correctness has a weak pooled association with ρ^3 (Pearson $r = 0.10$, $R^2 = 0.009$, $n = 153$). L2 Tool is the only setting with a strong positive correlation, but correctness is saturated near 1.00 there and provides little usable variation. The correlations in L2 Interaction and L2 Domain are weak ($|r| \lesssim 0.25$). These observational correlations do not establish that language drift causes the differences in ρ^3 .

G All results

Domain	Model	EN			
		p@1	p ²	p ³	ρ ³
Airline	GPT-5-mini	0.693	0.593	0.540	0.779
	Qwen3-235B	0.500	0.387	0.340	0.680
	Kimi-K2.5	0.707	0.600	0.540	0.764
Retail	GPT-5-mini	0.655	0.497	0.412	0.629
	Qwen3-235B	0.579	0.415	0.342	0.591
	Kimi-K2.5	0.561	0.442	0.395	0.704
Telecom	GPT-5-mini	0.713	0.588	0.509	0.714
	Qwen3-235B	0.357	0.205	0.132	0.370
	Kimi-K2.5	0.997	0.953	0.930	0.933

Table 17: English Baseline (S1) results. Where EN denotes English, p@1 denotes pass@1, p² denotes pass², and p³ denotes pass³. Bold values indicate the best score within each domain–metric group. Qwen3-235B refers to Qwen3-235B-A22B-Instruct-2507.

Domain	Model	VI				TH				ID				ZH				TL			
		p@1	p ²	p ³	ρ ³	p@1	p ²	p ³	ρ ³	p@1	p ²	p ³	ρ ³	p@1	p ²	p ³	ρ ³	p@1	p ²	p ³	ρ ³
Airline	GPT-5-mini	0.640	0.500	0.420	0.656	0.593	0.473	0.400	0.674	0.627	0.447	0.360	0.574	0.620	0.460	0.380	0.613	0.620	0.527	0.480	0.774
	Qwen3-235B	0.480	0.347	0.280	0.583	0.487	0.373	0.320	0.657	0.493	0.347	0.260	0.527	0.460	0.300	0.220	0.478	0.440	0.313	0.260	0.591
	Kimi-K2.5	0.733	0.660	0.600	0.818	0.687	0.580	0.520	0.757	0.740	0.653	0.600	0.811	0.760	0.640	0.580	0.763	0.747	0.653	0.600	0.804
Retail	GPT-5-mini	0.696	0.570	0.500	0.718	0.699	0.538	0.430	0.615	0.684	0.547	0.456	0.667	0.675	0.506	0.412	0.610	0.655	0.518	0.430	0.656
	Qwen3-235B	0.556	0.418	0.333	0.600	0.526	0.383	0.325	0.617	0.608	0.462	0.386	0.635	0.591	0.442	0.351	0.594	0.573	0.444	0.386	0.674
	Kimi-K2.5	0.687	0.550	0.465	0.677	0.573	0.430	0.360	0.627	0.640	0.483	0.386	0.603	0.722	0.597	0.526	0.729	0.675	0.541	0.474	0.701
Telecom	GPT-5-mini	0.561	0.430	0.368	0.656	0.605	0.436	0.342	0.565	0.582	0.453	0.404	0.693	0.728	0.608	0.518	0.711	0.740	0.617	0.535	0.723
	Qwen3-235B	0.260	0.129	0.061	0.236	0.222	0.108	0.061	0.276	0.333	0.170	0.105	0.316	0.228	0.132	0.097	0.423	0.155	0.088	0.061	0.396
	Kimi-K2.5	0.912	0.836	0.772	0.846	0.854	0.763	0.702	0.822	0.915	0.839	0.772	0.843	0.930	0.868	0.816	0.877	0.883	0.784	0.702	0.795

Table 18: L2 Interaction (S2) results across Vietnamese (VI), Thai (TH), Indonesian (ID), Chinese (ZH), and Filipino (TL). Here, p@1 denotes pass@1, p² denotes pass², and p³ denotes pass³. Bold values indicate the best score within each domain–metric group. Qwen3-235B refers to Qwen3-235B-A22B-Instruct-2507.

Domain	Model	VI				TH				ID				ZH				TL			
		p@1	p ²	p ³	ρ ³	p@1	p ²	p ³	ρ ³	p@1	p ²	p ³	ρ ³	p@1	p ²	p ³	ρ ³	p@1	p ²	p ³	ρ ³
Airline	GPT-5-mini	0.653	0.527	0.460	0.704	0.620	0.473	0.400	0.645	0.687	0.567	0.480	0.699	0.687	0.593	0.520	0.757	0.667	0.580	0.540	0.810
	Kimi-K2.5	0.727	0.620	0.560	0.770	0.713	0.587	0.500	0.701	0.753	0.640	0.560	0.744	0.700	0.593	0.540	0.771	0.773	0.693	0.640	0.828
	Qwen3-235B	0.447	0.313	0.260	0.582	0.380	0.253	0.220	0.579	0.400	0.233	0.180	0.450	0.480	0.313	0.200	0.417	0.467	0.320	0.240	0.514
Retail	GPT-5-mini	0.681	0.532	0.430	0.631	0.646	0.477	0.377	0.584	0.675	0.526	0.430	0.637	0.719	0.579	0.482	0.670	0.661	0.515	0.439	0.664
	Kimi-K2.5	0.658	0.497	0.404	0.614	0.602	0.456	0.360	0.598	0.658	0.523	0.439	0.667	0.673	0.532	0.456	0.678	0.629	0.482	0.395	0.628
	Qwen3-235B	0.547	0.412	0.351	0.642	0.561	0.421	0.351	0.626	0.550	0.418	0.351	0.638	0.594	0.433	0.360	0.606	0.570	0.421	0.342	0.600
Telecom	GPT-5-mini	0.664	0.550	0.482	0.726	0.713	0.588	0.518	0.727	0.713	0.596	0.518	0.727	0.705	0.576	0.500	0.709	0.719	0.594	0.509	0.708
	Kimi-K2.5	0.971	0.947	0.930	0.958	0.965	0.936	0.912	0.945	0.953	0.927	0.904	0.949	0.953	0.909	0.868	0.911	0.971	0.944	0.921	0.949
	Qwen3-235B	0.254	0.135	0.088	0.346	0.339	0.208	0.140	0.413	0.269	0.155	0.105	0.390	0.222	0.135	0.096	0.432	0.301	0.167	0.105	0.349

Table 19: L2 Tool (S3) results in monolingual settings across Vietnamese (VI), Thai (TH), Indonesian (ID), Chinese (ZH), and Filipino (TL). Here, p@1 denotes pass@1, p² denotes pass², and p³ denotes pass³. Bold values indicate the best score within each domain–metric group. Qwen3-235B refers to Qwen3-235B-A22B-Instruct-2507.

Domain	Model	Bi				Tri				Quad				Multi			
		p@1	p ²	p ³	ρ^3	p@1	p ²	p ³	ρ^3	p@1	p ²	p ³	ρ^3	p@1	p ²	p ³	ρ^3
Airline	GPT-5-mini	0.680	0.560	0.480	0.706	0.660	0.560	0.480	0.727	0.640	0.520	0.480	0.750	0.640	0.513	0.440	0.688
	Qwen3-235B	0.440	0.280	0.200	0.455	0.413	0.267	0.200	0.484	0.413	0.287	0.240	0.581	0.413	0.267	0.220	0.533
Retail	GPT-5-mini	0.667	0.541	0.465	0.697	0.667	0.538	0.456	0.684	0.652	0.485	0.386	0.592	0.696	0.547	0.447	0.642
	Qwen3-235B	0.596	0.456	0.386	0.648	0.535	0.398	0.325	0.607	0.585	0.447	0.377	0.644	0.570	0.401	0.298	0.523
Telecom	GPT-5-mini	0.664	0.541	0.482	0.726	0.696	0.558	0.491	0.705	0.670	0.550	0.491	0.733	0.652	0.520	0.465	0.713
	Qwen3-235B	0.298	0.155	0.088	0.295	0.392	0.213	0.140	0.357	0.365	0.181	0.105	0.288	0.342	0.181	0.114	0.333

Table 20: L2 Tool (S3) results for multilingual settings, including bilingual (Bi), trilingual (Tri), quadlingual (Quad), and multilingual (Multi) configurations. Here, p@1 denotes pass@1, p² denotes pass², and p³ denotes pass³. Bold values indicate the best score within each domain–metric group. Qwen3-235B refers to Qwen3-235B-A22B-Instruct-2507.

Domain	Model	VI				TH				ID				ZH				TL			
		p@1	p ²	p ³	ρ^3	p@1	p ²	p ³	ρ^3	p@1	p ²	p ³	ρ^3	p@1	p ²	p ³	ρ^3	p@1	p ²	p ³	ρ^3
Airline	GPT-5-mini	0.540	0.400	0.320	0.593	0.593	0.453	0.360	0.607	0.600	0.540	0.500	0.833	0.607	0.493	0.440	0.725	0.560	0.433	0.380	0.679
	Qwen3-235B	0.487	0.460	0.440	0.903	0.460	0.413	0.380	0.826	0.480	0.440	0.420	0.875	0.500	0.473	0.460	0.920	0.473	0.420	0.400	0.846
	Kimi-K2.5	0.560	0.427	0.340	0.607	0.547	0.387	0.300	0.548	0.600	0.487	0.420	0.700	0.660	0.507	0.440	0.667	0.600	0.487	0.400	0.667
Retail	GPT-5-mini	0.585	0.427	0.333	0.569	0.325	0.175	0.123	0.378	0.541	0.392	0.325	0.601	0.596	0.450	0.368	0.617	0.488	0.333	0.263	0.539
	Qwen3-235B	0.289	0.205	0.149	0.516	0.178	0.143	0.132	0.742	0.243	0.170	0.140	0.576	0.266	0.181	0.149	0.560	0.260	0.187	0.158	0.608
	Kimi-K2.5	0.567	0.392	0.268	0.473	0.327	0.190	0.114	0.349	0.433	0.287	0.202	0.467	0.607	0.450	0.366	0.603	0.444	0.292	0.219	0.493
Telecom	GPT-5-mini	0.564	0.427	0.360	0.638	0.599	0.447	0.360	0.601	0.497	0.325	0.246	0.495	0.760	0.652	0.588	0.774	0.681	0.547	0.474	0.696
	Qwen3-235B	0.383	0.213	0.149	0.389	0.284	0.143	0.079	0.278	0.427	0.295	0.211	0.494	0.330	0.205	0.149	0.452	0.330	0.205	0.149	0.452
	Kimi-K2.5	0.699	0.529	0.404	0.578	0.693	0.523	0.43	0.620	0.798	0.681	0.588	0.737	0.863	0.763	0.684	0.793	0.743	0.582	0.474	0.638

Table 21: L2 Domain (S4) results across Vietnamese (VI), Thai (TH), Indonesian (ID), Chinese (ZH), and Filipino (TL). Here, p@1 denotes pass@1, p² denotes pass², and p³ denotes pass³. Bold values indicate the best score within each domain–metric group. Qwen3-235B refers to Qwen3-235B-A22B-Instruct-2507.

H Detailed Error Analysis Results

H.1 Error taxonomy

Errors are annotated by an LLM judge following the review prompt of [Shi et al. \(2026\)](#), as implemented in τ^2 -Bench. For each error the judge records the responsible party, one or more tags, a severity level, and the turn at which it occurred. Table 23 lists the tags and Table 22 the severity levels, both quoted from the prompt. Because one error instance may carry several tags, tag counts sum to more than the number of errors.

H.2 Per-tag error rates

H.3 Per-domain error counts

Tables 24 to 26 report simulation-level severity per domain; the tables that follow give per-tag counts split by severity for the agent and the simulated user. Both sets cover all four scenarios, the five single-language tool variants, and the mixed-language tool variants (Mix-2–Mix-5).

Level	Definition given to the judge
<i>Agent errors</i>	
critical	The error directly caused task failure or violated important security/policy requirements, even if the task succeeded.
minor	The error was suboptimal but did not affect the final outcome or violate critical policies.
<i>Simulated-user errors</i>	
critical_helped	The user error helped the agent succeed inappropriately, e.g. the user supplied information it should not have, making the task too easy.
critical_hindered	The user error hindered the agent or made the task harder or impossible, e.g. the user supplied incorrect information or contradicted its instructions.
minor	The user made an error but it did not influence the simulation outcome.

Table 22: Severity levels assigned by the LLM judge. We report minor as *Benign* throughout. Figures pool critical_helped and critical_hindered into a single *Critical* category for the simulated user. Agent and user severities follow separate rubrics and are not directly comparable: the user is additionally fact-checked on every factual detail it supplies, with any detail not derivable from its instructions counted as a hallucination.

Tag	Identifier	Definition given to the judge
<i>Policy and procedure adherence</i>		
Missed Required Action	missed_required_action	Did not take a required action that was expected.
Guideline Violation	guideline_violation	Message or action not consistent with the provided guidelines or policy.
Wrong Sequence	wrong_sequence	Performed actions out of the expected order or sequence.
Premature Termination	premature_termination	Ended the conversation or accepted an incomplete outcome while the other participant was actively working and making progress. Not applied when the user ended the conversation because the agent was unresponsive or repeatedly failing.
<i>Comprehension</i>		
Incorrect Interpretation	incorrect_interpretation	Misinterpreted available information (e.g., misread a tool result or misunderstood a message).
Hallucination	hallucination	Made up information not grounded in guidelines, instructions, or tool call results. For user errors specifically: provided factual details (e.g., zip codes, sizes, product descriptions) not present in or derivable from the user instructions.
Inconsistent Behavior	inconsistent_behavior	Action or statement contradicts earlier statements or actions in the conversation.
<i>Tool-schema logic</i>		
Tool Call Argument Error	tool_call_argument_error	Made a tool call with correct schema but incorrect argument values.
Tool Call Schema Error	tool_call_schema_error	Made a tool call with invalid tool name, missing arguments, or wrong argument types.
Irrelevant Tool Call	irrelevant_tool_call	Made a tool call not relevant to the current task or conversation state.
<i>Other</i>		
Revealed Info Early	revealed_info_early	Shared information before it was appropriate or before proper verification.
Other	other	Used only when no other tag applies; the judge describes the error type in its reasoning.
Interruption Error [†]	interruption_error	Interrupted inappropriately or failed to interrupt when appropriate.

Table 23: Error tags available to the LLM judge, quoted from the review prompt. Groupings are ours and are used in Section 6.1 to relate tags to the capability a localized task stresses; the judge receives a flat list. A single error instance may carry several tags, and may span groups. [†]interruption_error is defined only in the full-duplex variant of the prompt; SEATauBench runs text-mode simulations exclusively, so this tag is never offered to the judge and its count is zero by construction rather than by rarity.

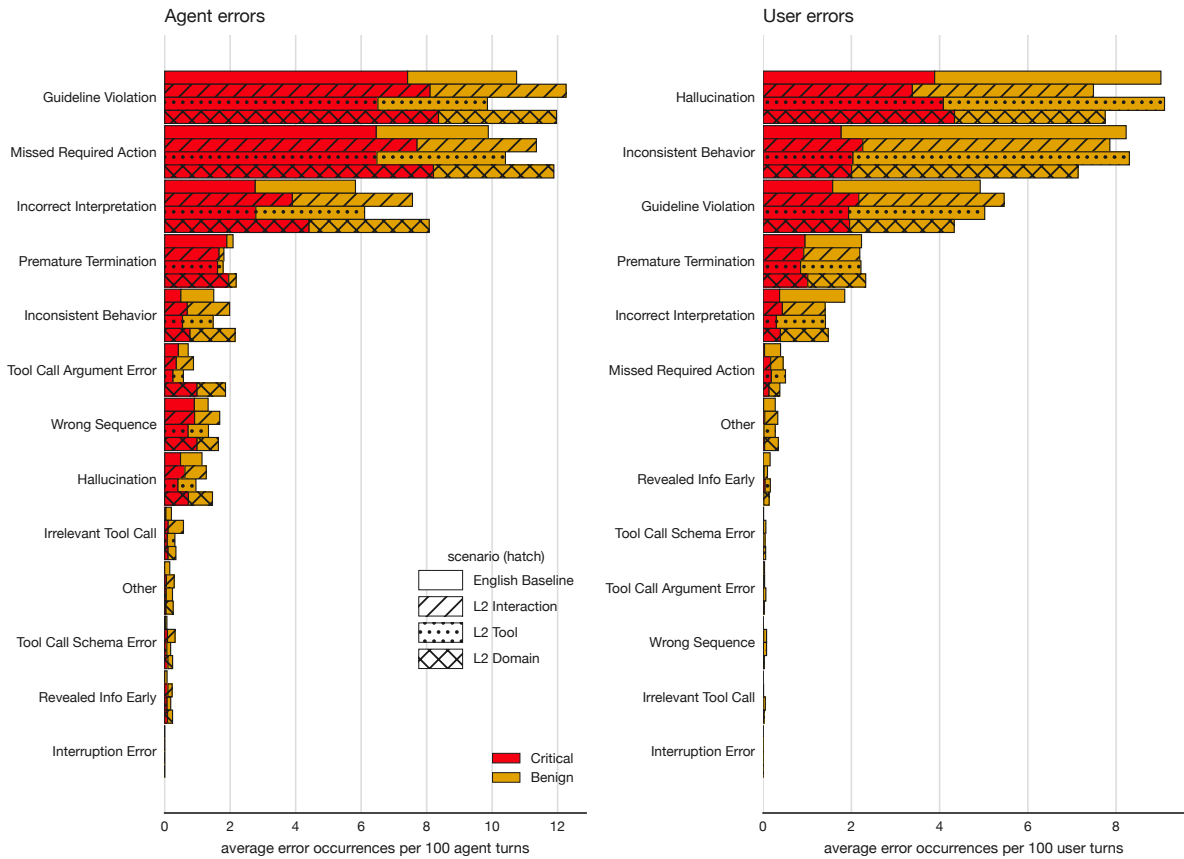


Figure 15: Agent and simulated-user error tags across the four scenarios. Bars give the average number of error occurrences per 100 turns of the relevant role — agent errors per assistant turn, user errors per user turn — computed within each scenario $\times$ language $\times$ domain cell and then averaged over the three domains and the five L2. Because a turn may be flagged more than once with the same tag, possibly at different severities, this counts occurrences rather than turns and sits slightly above the share of turns affected. For the simulated user, critical includes critical_helped and critical_hindered; agent and user severities follow role-specific rubrics and are not comparable across panels. All 13 tags are shown, with interruption_error at zero in both panels because it applies only to full-duplex simulations. L2 Tool averages the five single-language tool variants; the mixed-language variants (Mix-2–Mix-5) appear in Appendix H.3. Definitions are in Tables 22 and 23. Agent gpt-5-mini, simulated user qwen3-235b-a22b-instruct-2507, judge DeepSeek-V4-Flash.

Setting	Variant	Agent Errors			User Errors		
		Critical	Minor	Correct	Critical	Minor	Correct
English Baseline	English	81	22	47	38	48	64
L2 Interaction	Vietnamese	93	25	32	43	48	59
	Thai	105	14	31	42	45	63
	Indonesian	88	18	44	36	49	65
	Chinese	90	18	42	44	51	55
	Filipino	94	18	38	53	36	61
L2 Tool	VI Tool	81	23	46	43	51	56
	TH Tool	88	28	34	39	45	66
	ID Tool	94	20	36	41	41	68
	ZH Tool	84	23	43	35	44	71
	FIL Tools	92	17	41	45	45	60
	Mix-2 (EN+TH)	90	24	36	52	38	60
	Mix-3 (+VI)	86	17	47	33	54	63
	Mix-4 (+ID)	89	26	35	35	45	70
	Mix-5 (+ZH)	102	15	33	40	41	69
L2 Domain	Vietnamese	83	27	40	43	41	66
	Thai	97	17	36	55	30	65
	Indonesian	92	24	34	42	39	69
	Chinese	103	15	32	40	48	62
	Filipino	101	17	32	50	47	53

Table 24: Error analysis on the **Airline** domain (gpt-5-mini agent, qwen3-235b user simulator, DeepSeek-V4-Flash judge, $n=150$ per condition). Counts of simulations by maximum severity level.

Setting	Variant	Agent Errors			User Errors		
		Critical	Minor	Correct	Critical	Minor	Correct
English Baseline	English	163	44	135	68	136	138
L2 Interaction	Vietnamese	176	36	130	81	124	137
	Thai	259	21	62	73	108	161
	Indonesian	179	36	127	73	135	134
	Chinese	161	43	138	70	142	130
	Filipino	201	35	106	104	109	129
L2 Tool	VI Tools	172	39	131	62	147	133
	TH Tools	195	35	112	66	135	141
	ID Tools	199	29	114	68	131	143
	ZH Tools	171	44	127	76	129	137
	FIL Tools	191	40	111	74	135	133
	Mix-2 (EN+TH)	191	35	116	78	122	142
	Mix-3 (+VI)	174	37	131	75	135	132
	Mix-4 (+ID)	176	40	126	67	133	142
	Mix-5 (+ZH)	197	32	113	62	140	140
L2 Domain	Vietnamese	183	45	114	76	130	136
	Thai	250	28	64	103	106	133
	Indonesian	190	44	108	88	122	132
	Chinese	175	41	126	78	135	129
	Filipino	216	33	93	92	118	132

Table 25: Error analysis on the **Retail** domain (gpt-5-mini agent, qwen3-235b user simulator, DeepSeek-V4-Flash judge, $n=342$ per condition). Counts of simulations by maximum severity level.

Setting	Variant	Agent Errors			User Errors		
		Critical	Minor	Correct	Critical	Minor	Correct
English Baseline	English	123	36	183	78	116	148
L2 Interaction	Vietnamese	151	43	148	90	111	141
	Thai	158	40	144	88	118	136
	Indonesian	158	40	144	78	116	148
	Chinese	123	36	183	78	116	148
	Filipino	147	32	163	107	87	148
L2 Tool	VI Tools	132	40	170	75	124	143
	TH Tools	135	39	168	77	124	141
	ID Tools	157	31	154	77	122	143
	ZH Tools	124	37	181	72	130	140
	FIL Tools	143	31	168	85	103	154
	Mix-2 (EN+TH)	146	31	165	82	117	143
	Mix-3 (+VI)	147	35	160	81	120	141
	Mix-4 (+ID)	145	34	163	84	119	139
	Mix-5 (+ZH)	139	36	167	76	127	139
L2 Domain	Vietnamese	166	40	136	86	115	141
	Thai	170	36	136	84	109	149
	Indonesian	195	38	109	87	107	148
	Chinese	124	39	179	74	121	147
	Filipino	140	44	158	96	106	140

Table 26: Error analysis on the **Telecom** domain (gpt-5-mini agent, qwen3-235b user simulator, DeepSeek-V4-Flash judge, $n=342$ per condition). Counts of simulations by maximum severity level.

Error Tag	Eng. Baseline	L2 Interaction				
	EN	VI	TH	ID	ZH	FIL
Guideline Violation	C:100 M:20	C:116 M:38	C:113 M:23	C:89 M:31	C:85 M:27	C:110 M:28
Hallucination	C:6 M:7	C:3 M:6	C:4 M:8	C:6 M:5	C:9 M:1	C:8 M:5
Inconsistent Behavior	C:4 M:8	C:7 M:11	C:11 M:9	C:6 M:15	C:5 M:7	C:5 M:3
Incorrect Interpretation	C:33 M:25	C:67 M:29	C:56 M:30	C:47 M:33	C:46 M:38	C:46 M:36
Interruption Error	C:0 M:0	C:0 M:0	C:0 M:0	C:0 M:0	C:0 M:0	C:0 M:0
Irrelevant Tool Call	C:0 M:0	C:3 M:4	C:0 M:4	C:0 M:4	C:0 M:2	C:0 M:0
Missed Required Action	C:71 M:16	C:82 M:28	C:98 M:18	C:74 M:15	C:85 M:20	C:90 M:25
Other	C:0 M:1	C:2 M:4	C:0 M:0	C:0 M:1	C:1 M:3	C:0 M:0
Premature Termination	C:28 M:1	C:20 M:1	C:30 M:2	C:21 M:1	C:18 M:0	C:25 M:1
Revealed Info Early	C:0 M:1	C:0 M:2	C:0 M:1	C:1 M:0	C:1 M:0	C:2 M:2
Tool Call Argument Error	C:5 M:4	C:3 M:1	C:7 M:6	C:4 M:6	C:3 M:7	C:1 M:4
Tool Call Schema Error	C:0 M:0	C:1 M:0	C:2 M:2	C:0 M:2	C:0 M:1	C:0 M:4
Wrong Sequence	C:9 M:3	C:9 M:7	C:8 M:3	C:3 M:4	C:11 M:4	C:7 M:3

Table 27: Agent error tag counts — **Airline** domain, English baseline and L2 Interaction settings (gpt-5-mini agent, qwen3-235b user simulator, DeepSeek-V4-Flash judge, $n=150$ per condition). **C:** = critical-severity count; **M:** = minor-severity count. Counts are judge-annotated error instances pooled over all simulations; a simulation may contribute several instances and an instance may carry several tags, so column sums can exceed n .

Error Tag	Tool Adaptation — Single-language				
	VI	TH	ID	ZH	FIL
Guideline Violation	C:82 M:19	C:105 M:30	C:87 M:32	C:92 M:26	C:89 M:21
Hallucination	C:0 M:4	C:5 M:2	C:5 M:6	C:2 M:4	C:6 M:5
Inconsistent Behavior	C:12 M:5	C:11 M:3	C:5 M:6	C:5 M:10	C:3 M:5
Incorrect Interpretation	C:25 M:29	C:46 M:35	C:33 M:40	C:42 M:31	C:39 M:29
Interruption Error	C:0 M:0	C:0 M:0	C:0 M:0	C:0 M:0	C:0 M:0
Irrelevant Tool Call	C:4 M:0	C:0 M:1	C:0 M:0	C:0 M:2	C:0 M:1
Missed Required Action	C:79 M:22	C:75 M:22	C:78 M:17	C:75 M:27	C:85 M:22
Other	C:0 M:2	C:0 M:1	C:2 M:1	C:0 M:0	C:0 M:0
Premature Termination	C:22 M:1	C:21 M:4	C:19 M:1	C:23 M:3	C:34 M:1
Revealed Info Early	C:3 M:1	C:1 M:1	C:0 M:0	C:0 M:0	C:1 M:1
Tool Call Argument Error	C:2 M:6	C:2 M:6	C:1 M:4	C:1 M:0	C:5 M:0
Tool Call Schema Error	C:0 M:0	C:0 M:1	C:0 M:0	C:2 M:0	C:0 M:0
Wrong Sequence	C:12 M:6	C:3 M:3	C:5 M:3	C:12 M:6	C:5 M:2

Table 28: Agent error tag counts — **Airline** domain, Tool Adaptation (single-language) settings.

Error Tag	Tool Adaptation — Mixed-language			
	Mix-2	Mix-3	Mix-4	Mix-5
Guideline Violation	C:110 M:22	C:97 M:16	C:96 M:24	C:103 M:26
Hallucination	C:3 M:2	C:3 M:5	C:2 M:5	C:7 M:2
Inconsistent Behavior	C:3 M:5	C:13 M:4	C:8 M:9	C:6 M:5
Incorrect Interpretation	C:34 M:29	C:33 M:24	C:36 M:34	C:46 M:21
Interruption Error	C:0 M:0	C:0 M:0	C:0 M:0	C:0 M:0
Irrelevant Tool Call	C:0 M:1	C:1 M:1	C:1 M:3	C:0 M:2
Missed Required Action	C:78 M:20	C:78 M:14	C:87 M:33	C:84 M:21
Other	C:0 M:1	C:2 M:0	C:1 M:1	C:0 M:1
Premature Termination	C:29 M:1	C:35 M:2	C:26 M:1	C:45 M:2
Revealed Info Early	C:1 M:1	C:2 M:1	C:0 M:0	C:0 M:0
Tool Call Argument Error	C:3 M:3	C:2 M:5	C:1 M:4	C:1 M:2
Tool Call Schema Error	C:0 M:1	C:1 M:0	C:0 M:1	C:0 M:2
Wrong Sequence	C:5 M:1	C:11 M:1	C:5 M:1	C:7 M:1

Table 29: Agent error tag counts — **Airline** domain, Tool Adaptation (mixed-language) settings.

Error Tag	L2 Domain				
	VI	TH	ID	ZH	FIL
Guideline Violation	C:90 M:36	C:130 M:28	C:115 M:18	C:108 M:21	C:114 M:24
Hallucination	C:8 M:2	C:14 M:10	C:3 M:5	C:7 M:8	C:6 M:7
Inconsistent Behavior	C:6 M:13	C:9 M:10	C:6 M:8	C:7 M:10	C:6 M:5
Incorrect Interpretation	C:54 M:37	C:71 M:35	C:43 M:25	C:47 M:32	C:58 M:30
Interruption Error	C:0 M:0	C:0 M:0	C:0 M:0	C:0 M:0	C:0 M:0
Irrelevant Tool Call	C:2 M:0	C:1 M:4	C:1 M:0	C:0 M:1	C:0 M:1
Missed Required Action	C:77 M:15	C:86 M:16	C:96 M:25	C:94 M:19	C:110 M:23
Other	C:1 M:0	C:1 M:0	C:0 M:1	C:0 M:1	C:0 M:4
Premature Termination	C:21 M:9	C:25 M:2	C:28 M:0	C:27 M:2	C:30 M:1
Revealed Info Early	C:1 M:2	C:0 M:2	C:3 M:1	C:1 M:2	C:0 M:0
Tool Call Argument Error	C:5 M:3	C:5 M:3	C:7 M:8	C:6 M:4	C:6 M:2
Tool Call Schema Error	C:1 M:1	C:0 M:0	C:0 M:1	C:0 M:1	C:0 M:0
Wrong Sequence	C:8 M:3	C:7 M:3	C:13 M:5	C:12 M:2	C:11 M:5

Table 30: Agent error tag counts — **Airline** domain, L2 Domain settings.

Error Tag	Eng. Baseline	L2 Interaction				
	EN	VI	TH	ID	ZH	FIL
Guideline Violation	C:13 M:26	C:18 M:22	C:19 M:25	C:15 M:30	C:28 M:23	C:26 M:31
Hallucination	C:26 M:27	C:22 M:24	C:31 M:24	C:19 M:37	C:23 M:31	C:39 M:23
Inconsistent Behavior	C:15 M:34	C:22 M:34	C:15 M:28	C:17 M:42	C:25 M:35	C:25 M:31
Incorrect Interpretation	C:4 M:7	C:3 M:8	C:2 M:9	C:2 M:4	C:3 M:9	C:11 M:7
Interruption Error	C:0 M:0	C:0 M:0	C:0 M:0	C:0 M:0	C:0 M:0	C:0 M:0
Irrelevant Tool Call	C:0 M:0	C:0 M:0	C:0 M:0	C:0 M:0	C:0 M:0	C:0 M:0
Missed Required Action	C:0 M:1	C:3 M:1	C:2 M:0	C:0 M:1	C:1 M:2	C:2 M:1
Other	C:0 M:1	C:0 M:1	C:0 M:0	C:0 M:0	C:0 M:2	C:1 M:0
Premature Termination	C:9 M:10	C:6 M:12	C:12 M:15	C:11 M:9	C:11 M:18	C:8 M:10
Revealed Info Early	C:0 M:2	C:1 M:0	C:0 M:2	C:0 M:0	C:0 M:1	C:0 M:2
Tool Call Argument Error	C:0 M:0	C:0 M:0	C:0 M:0	C:0 M:0	C:0 M:0	C:0 M:0
Tool Call Schema Error	C:0 M:0	C:0 M:0	C:0 M:0	C:0 M:0	C:0 M:0	C:0 M:0
Wrong Sequence	C:0 M:0	C:0 M:0	C:0 M:0	C:0 M:1	C:0 M:0	C:0 M:0

Table 31: User error tag counts — **Airline** domain, English baseline and L2 Interaction settings (gpt-5-mini agent, qwen3-235b user simulator, DeepSeek-V4-Flash judge, $n=150$ per condition). **C:** = critical-severity count; **M:** = minor-severity count. For the simulated user, **C:** pools critical_helped and critical_hindered. Counts are judge-annotated error instances pooled over all simulations; a simulation may contribute several instances and an instance may carry several tags, so column sums can exceed n .

Error Tag	Tool Adaptation — Single-language				
	VI	TH	ID	ZH	FIL
Guideline Violation	C:19 M:36	C:9 M:17	C:16 M:13	C:14 M:25	C:22 M:18
Hallucination	C:39 M:28	C:39 M:20	C:25 M:21	C:32 M:28	C:35 M:29
Inconsistent Behavior	C:15 M:45	C:19 M:27	C:13 M:40	C:14 M:37	C:24 M:33
Incorrect Interpretation	C:2 M:11	C:3 M:2	C:3 M:5	C:1 M:6	C:1 M:4
Interruption Error	C:0 M:0	C:0 M:0	C:0 M:0	C:0 M:0	C:0 M:0
Irrelevant Tool Call	C:0 M:0	C:0 M:0	C:0 M:0	C:0 M:0	C:0 M:0
Missed Required Action	C:1 M:1	C:1 M:0	C:3 M:1	C:1 M:0	C:1 M:2
Other	C:0 M:0	C:0 M:2	C:0 M:0	C:1 M:0	C:0 M:1
Premature Termination	C:4 M:7	C:10 M:17	C:10 M:16	C:7 M:10	C:9 M:15
Revealed Info Early	C:0 M:2	C:0 M:0	C:0 M:0	C:1 M:1	C:0 M:1
Tool Call Argument Error	C:0 M:0	C:0 M:0	C:0 M:0	C:0 M:0	C:0 M:0
Tool Call Schema Error	C:0 M:0	C:0 M:0	C:0 M:0	C:0 M:0	C:0 M:0
Wrong Sequence	C:0 M:0	C:0 M:0	C:0 M:0	C:0 M:1	C:0 M:1

Table 32: User error tag counts — **Airline** domain, Tool Adaptation (single-language) settings.

Error Tag	Tool Adaptation — Mixed-language			
	Mix-2	Mix-3	Mix-4	Mix-5
Guideline Violation	C:23 M:19	C:16 M:22	C:19 M:20	C:18 M:14
Hallucination	C:46 M:19	C:30 M:26	C:25 M:21	C:33 M:20
Inconsistent Behavior	C:21 M:37	C:19 M:36	C:15 M:37	C:25 M:33
Incorrect Interpretation	C:6 M:8	C:4 M:9	C:3 M:4	C:1 M:7
Interruption Error	C:0 M:0	C:0 M:0	C:0 M:0	C:0 M:0
Irrelevant Tool Call	C:0 M:0	C:0 M:0	C:0 M:0	C:0 M:0
Missed Required Action	C:1 M:1	C:0 M:0	C:1 M:3	C:0 M:1
Other	C:0 M:0	C:0 M:3	C:1 M:0	C:0 M:0
Premature Termination	C:9 M:7	C:7 M:9	C:6 M:11	C:3 M:11
Revealed Info Early	C:0 M:1	C:0 M:0	C:0 M:0	C:0 M:1
Tool Call Argument Error	C:0 M:0	C:0 M:0	C:0 M:0	C:0 M:0
Tool Call Schema Error	C:0 M:0	C:0 M:0	C:0 M:0	C:0 M:0
Wrong Sequence	C:0 M:0	C:1 M:0	C:0 M:0	C:0 M:0

Table 33: User error tag counts — **Airline** domain, Tool Adaptation (mixed-language) settings.

Error Tag	L2 Domain				
	VI	TH	ID	ZH	FIL
Guideline Violation	C:16 M:20	C:29 M:14	C:20 M:13	C:18 M:19	C:21 M:24
Hallucination	C:35 M:22	C:48 M:13	C:24 M:13	C:33 M:18	C:29 M:21
Inconsistent Behavior	C:16 M:23	C:26 M:30	C:15 M:27	C:18 M:37	C:19 M:39
Incorrect Interpretation	C:3 M:8	C:3 M:2	C:3 M:6	C:0 M:13	C:5 M:8
Interruption Error	C:0 M:0	C:0 M:0	C:0 M:0	C:0 M:0	C:0 M:0
Irrelevant Tool Call	C:0 M:0	C:0 M:0	C:0 M:0	C:0 M:0	C:0 M:0
Missed Required Action	C:2 M:3	C:0 M:1	C:0 M:0	C:0 M:0	C:3 M:3
Other	C:0 M:2	C:0 M:0	C:0 M:1	C:0 M:0	C:2 M:1
Premature Termination	C:8 M:14	C:10 M:8	C:9 M:11	C:9 M:12	C:11 M:16
Revealed Info Early	C:0 M:3	C:0 M:0	C:0 M:0	C:0 M:3	C:0 M:1
Tool Call Argument Error	C:0 M:0	C:0 M:0	C:0 M:0	C:0 M:0	C:0 M:0
Tool Call Schema Error	C:0 M:0	C:0 M:0	C:0 M:0	C:0 M:0	C:0 M:0
Wrong Sequence	C:0 M:0	C:0 M:0	C:0 M:0	C:0 M:1	C:0 M:0

Table 34: User error tag counts — **Airline** domain, L2 Domain settings.

Error Tag	Eng. Baseline	L2 Interaction				
	EN	VI	TH	ID	ZH	FIL
Guideline Violation	C:70 M:74	C:99 M:80	C:105 M:81	C:88 M:83	C:71 M:59	C:86 M:94
Hallucination	C:7 M:11	C:9 M:12	C:15 M:13	C:17 M:17	C:13 M:11	C:10 M:10
Inconsistent Behavior	C:10 M:15	C:17 M:24	C:13 M:24	C:8 M:15	C:8 M:28	C:8 M:16
Incorrect Interpretation	C:34 M:39	C:32 M:54	C:42 M:59	C:38 M:54	C:26 M:44	C:37 M:39
Interruption Error	C:0 M:0	C:0 M:0	C:0 M:0	C:0 M:0	C:0 M:0	C:0 M:0
Irrelevant Tool Call	C:2 M:3	C:1 M:3	C:1 M:4	C:5 M:5	C:0 M:5	C:0 M:2
Missed Required Action	C:75 M:53	C:78 M:62	C:95 M:51	C:88 M:55	C:93 M:53	C:94 M:63
Other	C:0 M:2	C:0 M:3	C:1 M:7	C:2 M:1	C:0 M:3	C:0 M:3
Premature Termination	C:10 M:5	C:5 M:2	C:7 M:2	C:11 M:0	C:5 M:7	C:9 M:5
Revealed Info Early	C:0 M:0	C:1 M:5	C:4 M:4	C:4 M:2	C:1 M:2	C:2 M:3
Tool Call Argument Error	C:8 M:5	C:7 M:9	C:7 M:18	C:8 M:6	C:9 M:10	C:5 M:3
Tool Call Schema Error	C:1 M:2	C:3 M:5	C:4 M:6	C:1 M:5	C:0 M:4	C:4 M:9
Wrong Sequence	C:20 M:7	C:24 M:24	C:20 M:17	C:19 M:20	C:19 M:16	C:23 M:15

Table 35: Agent error tag counts — **Retail** domain, English baseline and L2 Interaction settings (gpt-5-mini agent, qwen3-235b user simulator, DeepSeek-V4-Flash judge, $n=342$ per condition). **C:** = critical-severity count; **M:** = minor-severity count. Counts are judge-annotated error instances pooled over all simulations; a simulation may contribute several instances and an instance may carry several tags, so column sums can exceed n .

Error Tag	Tool Adaptation — Single-language				
	VI	TH	ID	ZH	FIL
Guideline Violation	C:68 M:52	C:70 M:66	C:69 M:69	C:62 M:55	C:78 M:56
Hallucination	C:9 M:11	C:6 M:13	C:3 M:11	C:16 M:7	C:15 M:10
Inconsistent Behavior	C:10 M:15	C:6 M:14	C:7 M:17	C:8 M:10	C:5 M:21
Incorrect Interpretation	C:31 M:53	C:40 M:47	C:28 M:39	C:30 M:40	C:36 M:34
Interruption Error	C:0 M:0	C:0 M:0	C:0 M:0	C:0 M:0	C:0 M:0
Irrelevant Tool Call	C:1 M:3	C:1 M:5	C:0 M:0	C:0 M:2	C:2 M:2
Missed Required Action	C:62 M:59	C:79 M:66	C:74 M:64	C:69 M:63	C:69 M:69
Other	C:0 M:3	C:0 M:3	C:0 M:4	C:2 M:1	C:1 M:2
Premature Termination	C:6 M:3	C:7 M:3	C:8 M:3	C:12 M:3	C:6 M:1
Revealed Info Early	C:2 M:3	C:1 M:2	C:1 M:5	C:0 M:6	C:0 M:0
Tool Call Argument Error	C:9 M:7	C:5 M:7	C:8 M:9	C:6 M:3	C:8 M:6
Tool Call Schema Error	C:0 M:6	C:2 M:6	C:1 M:8	C:4 M:2	C:4 M:5
Wrong Sequence	C:16 M:14	C:12 M:21	C:18 M:18	C:8 M:13	C:11 M:9

Table 36: Agent error tag counts — **Retail** domain, Tool Adaptation (single-language) settings.

Error Tag	Tool Adaptation — Mixed-language			
	Mix-2	Mix-3	Mix-4	Mix-5
Guideline Violation	C:49 M:55	C:67 M:48	C:63 M:53	C:78 M:72
Hallucination	C:11 M:10	C:11 M:12	C:17 M:18	C:12 M:18
Inconsistent Behavior	C:10 M:15	C:10 M:13	C:7 M:16	C:8 M:26
Incorrect Interpretation	C:22 M:33	C:29 M:32	C:38 M:43	C:27 M:65
Interruption Error	C:0 M:0	C:0 M:0	C:0 M:0	C:0 M:0
Irrelevant Tool Call	C:0 M:1	C:1 M:3	C:0 M:3	C:1 M:5
Missed Required Action	C:58 M:56	C:63 M:71	C:73 M:49	C:59 M:67
Other	C:1 M:2	C:0 M:4	C:0 M:1	C:0 M:6
Premature Termination	C:12 M:4	C:17 M:2	C:5 M:1	C:6 M:2
Revealed Info Early	C:0 M:4	C:3 M:1	C:0 M:5	C:1 M:0
Tool Call Argument Error	C:5 M:14	C:7 M:4	C:8 M:9	C:5 M:8
Tool Call Schema Error	C:1 M:2	C:0 M:9	C:4 M:5	C:2 M:7
Wrong Sequence	C:14 M:14	C:7 M:10	C:11 M:13	C:11 M:18

Table 37: Agent error tag counts — **Retail** domain, Tool Adaptation (mixed-language) settings.

Error Tag	L2 Domain				
	VI	TH	ID	ZH	FIL
Guideline Violation	C:73 M:76	C:128 M:54	C:92 M:53	C:112 M:93	C:92 M:59
Hallucination	C:18 M:17	C:30 M:10	C:18 M:17	C:9 M:18	C:14 M:13
Inconsistent Behavior	C:13 M:27	C:29 M:25	C:12 M:22	C:24 M:27	C:22 M:22
Incorrect Interpretation	C:48 M:88	C:142 M:62	C:50 M:56	C:63 M:63	C:50 M:55
Interruption Error	C:0 M:0	C:0 M:0	C:0 M:0	C:0 M:0	C:0 M:0
Irrelevant Tool Call	C:0 M:1	C:11 M:1	C:1 M:4	C:1 M:5	C:1 M:1
Missed Required Action	C:73 M:66	C:160 M:62	C:75 M:65	C:98 M:83	C:101 M:81
Other	C:0 M:3	C:1 M:5	C:1 M:2	C:1 M:5	C:2 M:1
Premature Termination	C:12 M:4	C:28 M:4	C:15 M:4	C:14 M:1	C:7 M:3
Revealed Info Early	C:1 M:3	C:2 M:3	C:2 M:3	C:0 M:3	C:2 M:2
Tool Call Argument Error	C:35 M:49	C:48 M:39	C:39 M:28	C:48 M:25	C:37 M:34
Tool Call Schema Error	C:6 M:1	C:3 M:9	C:5 M:9	C:6 M:5	C:2 M:4
Wrong Sequence	C:14 M:11	C:24 M:8	C:19 M:21	C:12 M:15	C:13 M:12

Table 38: Agent error tag counts — **Retail** domain, L2 Domain settings.

Error Tag	Eng. Baseline	L2 Interaction				
	EN	VI	TH	ID	ZH	FIL
Guideline Violation	C:19 M:50	C:32 M:51	C:41 M:72	C:28 M:36	C:38 M:42	C:31 M:56
Hallucination	C:69 M:105	C:87 M:91	C:71 M:82	C:59 M:92	C:55 M:86	C:95 M:108
Inconsistent Behavior	C:22 M:93	C:38 M:85	C:40 M:101	C:38 M:96	C:32 M:81	C:44 M:101
Incorrect Interpretation	C:4 M:21	C:15 M:8	C:9 M:10	C:6 M:8	C:1 M:13	C:10 M:7
Interruption Error	C:0 M:0	C:0 M:0	C:0 M:0	C:0 M:0	C:0 M:0	C:0 M:0
Irrelevant Tool Call	C:0 M:0	C:0 M:0	C:0 M:0	C:0 M:0	C:0 M:0	C:0 M:0
Missed Required Action	C:0 M:2	C:1 M:1	C:2 M:4	C:4 M:2	C:3 M:1	C:3 M:3
Other	C:0 M:4	C:2 M:3	C:2 M:10	C:1 M:1	C:0 M:8	C:1 M:1
Premature Termination	C:9 M:19	C:11 M:17	C:17 M:18	C:12 M:19	C:16 M:22	C:10 M:15
Revealed Info Early	C:0 M:1	C:0 M:0	C:2 M:0	C:0 M:0	C:0 M:0	C:0 M:0
Tool Call Argument Error	C:0 M:0	C:0 M:0	C:0 M:0	C:0 M:0	C:0 M:0	C:0 M:0
Tool Call Schema Error	C:0 M:0	C:0 M:0	C:0 M:0	C:0 M:0	C:0 M:0	C:0 M:0
Wrong Sequence	C:0 M:0	C:0 M:0	C:0 M:0	C:0 M:0	C:0 M:0	C:0 M:0

Table 39: User error tag counts — **Retail** domain, English baseline and L2 Interaction settings (gpt-5-mini agent, qwen3-235b user simulator, DeepSeek-V4-Flash judge, $n=342$ per condition). **C:** = critical-severity count; **M:** = minor-severity count. For the simulated user, **C:** pools critical_helped and critical_hindered. Counts are judge-annotated error instances pooled over all simulations; a simulation may contribute several instances and an instance may carry several tags, so column sums can exceed n .

Error Tag	Tool Adaptation — Single-language				
	VI	TH	ID	ZH	FIL
Guideline Violation	C:19 M:46	C:22 M:34	C:41 M:43	C:29 M:57	C:24 M:47
Hallucination	C:71 M:115	C:55 M:90	C:61 M:132	C:74 M:104	C:54 M:123
Inconsistent Behavior	C:28 M:99	C:28 M:86	C:27 M:86	C:30 M:103	C:37 M:92
Incorrect Interpretation	C:2 M:10	C:5 M:16	C:4 M:8	C:8 M:16	C:7 M:15
Interruption Error	C:0 M:0	C:0 M:0	C:0 M:0	C:0 M:0	C:0 M:0
Irrelevant Tool Call	C:0 M:0	C:0 M:0	C:0 M:0	C:0 M:0	C:0 M:0
Missed Required Action	C:2 M:1	C:5 M:4	C:0 M:4	C:3 M:4	C:2 M:3
Other	C:0 M:2	C:0 M:4	C:0 M:2	C:0 M:2	C:1 M:3
Premature Termination	C:10 M:16	C:7 M:15	C:15 M:14	C:10 M:16	C:14 M:17
Revealed Info Early	C:2 M:2	C:1 M:0	C:3 M:0	C:1 M:2	C:0 M:1
Tool Call Argument Error	C:0 M:0	C:0 M:0	C:0 M:0	C:0 M:0	C:0 M:0
Tool Call Schema Error	C:0 M:0	C:0 M:0	C:0 M:0	C:0 M:0	C:0 M:0
Wrong Sequence	C:0 M:0	C:0 M:0	C:0 M:0	C:0 M:0	C:0 M:0

Table 40: User error tag counts — **Retail** domain, Tool Adaptation (single-language) settings.

Error Tag	Tool Adaptation — Mixed-language			
	Mix-2	Mix-3	Mix-4	Mix-5
Guideline Violation	C:17 M:37	C:30 M:39	C:23 M:29	C:27 M:40
Hallucination	C:53 M:123	C:82 M:106	C:69 M:94	C:64 M:129
Inconsistent Behavior	C:14 M:66	C:38 M:94	C:31 M:76	C:21 M:85
Incorrect Interpretation	C:7 M:12	C:5 M:12	C:5 M:12	C:12 M:14
Interruption Error	C:0 M:0	C:0 M:0	C:0 M:0	C:0 M:0
Irrelevant Tool Call	C:0 M:0	C:0 M:0	C:0 M:0	C:0 M:0
Missed Required Action	C:0 M:0	C:0 M:6	C:1 M:2	C:2 M:2
Other	C:1 M:5	C:0 M:7	C:0 M:2	C:0 M:6
Premature Termination	C:6 M:14	C:10 M:15	C:17 M:12	C:11 M:21
Revealed Info Early	C:1 M:1	C:1 M:1	C:0 M:0	C:0 M:0
Tool Call Argument Error	C:0 M:0	C:0 M:0	C:0 M:0	C:0 M:0
Tool Call Schema Error	C:0 M:0	C:0 M:0	C:0 M:0	C:0 M:0
Wrong Sequence	C:0 M:1	C:0 M:0	C:0 M:0	C:0 M:1

Table 41: User error tag counts — **Retail** domain, Tool Adaptation (mixed-language) settings.

Error Tag	L2 Domain				
	VI	TH	ID	ZH	FIL
Guideline Violation	C:25 M:31	C:29 M:23	C:29 M:32	C:33 M:31	C:23 M:33
Hallucination	C:75 M:76	C:111 M:74	C:76 M:105	C:119 M:110	C:115 M:106
Inconsistent Behavior	C:39 M:94	C:39 M:86	C:34 M:102	C:26 M:84	C:51 M:111
Incorrect Interpretation	C:5 M:20	C:11 M:16	C:7 M:12	C:5 M:17	C:13 M:15
Interruption Error	C:0 M:0	C:0 M:0	C:0 M:0	C:0 M:0	C:0 M:0
Irrelevant Tool Call	C:0 M:0	C:0 M:0	C:0 M:0	C:0 M:0	C:0 M:0
Missed Required Action	C:2 M:0	C:3 M:2	C:2 M:5	C:2 M:3	C:3 M:2
Other	C:0 M:4	C:0 M:3	C:0 M:4	C:1 M:3	C:0 M:4
Premature Termination	C:8 M:18	C:28 M:24	C:16 M:19	C:11 M:12	C:22 M:16
Revealed Info Early	C:0 M:1	C:3 M:2	C:0 M:0	C:0 M:1	C:0 M:2
Tool Call Argument Error	C:0 M:0	C:0 M:0	C:0 M:0	C:0 M:0	C:0 M:0
Tool Call Schema Error	C:0 M:0	C:0 M:0	C:0 M:0	C:0 M:0	C:0 M:0
Wrong Sequence	C:0 M:0	C:0 M:0	C:0 M:0	C:0 M:0	C:0 M:0

Table 42: User error tag counts — **Retail** domain, L2 Domain settings.

Error Tag	Eng. Baseline	L2 Interaction				
	EN	VI	TH	ID	ZH	FIL
Guideline Violation	C:88 M:81	C:77 M:128	C:99 M:105	C:103 M:116	C:71 M:46	C:61 M:68
Hallucination	C:4 M:8	C:6 M:16	C:7 M:15	C:6 M:14	C:3 M:5	C:4 M:11
Inconsistent Behavior	C:9 M:28	C:15 M:49	C:16 M:29	C:12 M:40	C:4 M:20	C:6 M:32
Incorrect Interpretation	C:39 M:94	C:38 M:84	C:47 M:100	C:43 M:100	C:36 M:53	C:29 M:60
Interruption Error	C:0 M:0	C:0 M:0	C:0 M:0	C:0 M:0	C:0 M:0	C:0 M:0
Irrelevant Tool Call	C:0 M:10	C:4 M:36	C:7 M:24	C:5 M:19	C:1 M:6	C:3 M:18
Missed Required Action	C:127 M:144	C:106 M:132	C:142 M:118	C:138 M:132	C:109 M:107	C:107 M:97
Other	C:0 M:6	C:1 M:6	C:0 M:10	C:1 M:8	C:1 M:3	C:1 M:8
Premature Termination	C:25 M:4	C:25 M:6	C:36 M:4	C:28 M:2	C:23 M:2	C:18 M:1
Revealed Info Early	C:0 M:2	C:2 M:2	C:2 M:2	C:0 M:1	C:1 M:1	C:0 M:0
Tool Call Argument Error	C:1 M:0	C:6 M:7	C:4 M:19	C:0 M:4	C:1 M:3	C:1 M:4
Tool Call Schema Error	C:0 M:1	C:1 M:3	C:0 M:6	C:0 M:1	C:1 M:0	C:1 M:2
Wrong Sequence	C:6 M:12	C:13 M:24	C:7 M:27	C:10 M:16	C:12 M:7	C:5 M:8

Table 43: Agent error tag counts — **Telecom** domain, English baseline and L2 Interaction settings (gpt-5-mini agent, qwen3-235b user simulator, DeepSeek-V4-Flash judge, $n=342$ per condition). **C:** = critical-severity count; **M:** = minor-severity count. Counts are judge-annotated error instances pooled over all simulations; a simulation may contribute several instances and an instance may carry several tags, so column sums can exceed n .

Error Tag	Tool Adaptation — Single-language				
	VI	TH	ID	ZH	FIL
Guideline Violation	C:73 M:90	C:69 M:77	C:57 M:94	C:77 M:61	C:49 M:88
Hallucination	C:4 M:18	C:4 M:11	C:6 M:8	C:2 M:14	C:2 M:11
Inconsistent Behavior	C:6 M:29	C:6 M:25	C:3 M:29	C:3 M:36	C:7 M:36
Incorrect Interpretation	C:41 M:80	C:17 M:91	C:30 M:80	C:37 M:74	C:26 M:66
Interruption Error	C:0 M:0	C:0 M:0	C:0 M:0	C:0 M:0	C:0 M:0
Irrelevant Tool Call	C:1 M:11	C:2 M:8	C:2 M:27	C:5 M:6	C:2 M:11
Missed Required Action	C:140 M:154	C:93 M:124	C:104 M:133	C:134 M:142	C:96 M:147
Other	C:0 M:5	C:1 M:14	C:0 M:17	C:2 M:5	C:5 M:2
Premature Termination	C:25 M:1	C:28 M:1	C:28 M:4	C:28 M:3	C:23 M:1
Revealed Info Early	C:1 M:1	C:1 M:3	C:1 M:3	C:1 M:1	C:1 M:0
Tool Call Argument Error	C:0 M:2	C:1 M:3	C:1 M:4	C:1 M:1	C:1 M:7
Tool Call Schema Error	C:0 M:0	C:0 M:0	C:0 M:0	C:0 M:1	C:2 M:0
Wrong Sequence	C:12 M:15	C:7 M:13	C:10 M:14	C:10 M:8	C:9 M:9

Table 44: Agent error tag counts — **Telecom** domain, Tool Adaptation (single-language) settings.

Error Tag	Tool Adaptation — Mixed-language			
	Mix-2	Mix-3	Mix-4	Mix-5
Guideline Violation	C:80 M:83	C:78 M:81	C:70 M:75	C:72 M:90
Hallucination	C:4 M:15	C:3 M:14	C:4 M:17	C:3 M:14
Inconsistent Behavior	C:8 M:27	C:9 M:29	C:3 M:19	C:7 M:26
Incorrect Interpretation	C:28 M:71	C:36 M:98	C:26 M:96	C:40 M:80
Interruption Error	C:0 M:0	C:0 M:0	C:0 M:0	C:0 M:0
Irrelevant Tool Call	C:2 M:14	C:1 M:36	C:0 M:6	C:1 M:20
Missed Required Action	C:107 M:153	C:100 M:189	C:106 M:135	C:118 M:118
Other	C:0 M:11	C:1 M:27	C:0 M:2	C:1 M:6
Premature Termination	C:36 M:4	C:28 M:4	C:33 M:0	C:37 M:2
Revealed Info Early	C:2 M:1	C:0 M:0	C:0 M:1	C:1 M:4
Tool Call Argument Error	C:3 M:1	C:2 M:3	C:0 M:0	C:0 M:6
Tool Call Schema Error	C:1 M:3	C:0 M:2	C:0 M:0	C:0 M:0
Wrong Sequence	C:8 M:13	C:7 M:16	C:7 M:9	C:12 M:18

Table 45: Agent error tag counts — **Telecom** domain, Tool Adaptation (mixed-language) settings.

Error Tag	L2 Domain				
	VI	TH	ID	ZH	FIL
Guideline Violation	C:117 M:104	C:106 M:97	C:98 M:91	C:78 M:81	C:106 M:115
Hallucination	C:7 M:16	C:5 M:11	C:5 M:19	C:5 M:9	C:3 M:11
Inconsistent Behavior	C:10 M:37	C:9 M:47	C:8 M:44	C:8 M:31	C:12 M:38
Incorrect Interpretation	C:50 M:69	C:54 M:95	C:53 M:77	C:29 M:71	C:54 M:85
Interruption Error	C:0 M:0	C:0 M:0	C:0 M:0	C:0 M:0	C:0 M:0
Irrelevant Tool Call	C:3 M:19	C:2 M:11	C:1 M:8	C:0 M:4	C:3 M:15
Missed Required Action	C:151 M:96	C:166 M:136	C:178 M:108	C:111 M:119	C:163 M:111
Other	C:0 M:15	C:1 M:9	C:1 M:7	C:0 M:7	C:1 M:2
Premature Termination	C:30 M:3	C:39 M:2	C:33 M:3	C:27 M:3	C:33 M:5
Revealed Info Early	C:3 M:4	C:1 M:4	C:0 M:5	C:0 M:0	C:2 M:2
Tool Call Argument Error	C:2 M:7	C:2 M:7	C:2 M:3	C:2 M:5	C:3 M:11
Tool Call Schema Error	C:2 M:1	C:0 M:0	C:1 M:3	C:0 M:0	C:5 M:4
Wrong Sequence	C:17 M:13	C:16 M:12	C:23 M:21	C:12 M:19	C:13 M:27

Table 46: Agent error tag counts — **Telecom** domain, L2 Domain settings.

Error Tag	Eng. Baseline	L2 Interaction				
	EN	VI	TH	ID	ZH	FIL
Guideline Violation	C:25 M:42	C:10 M:93	C:30 M:77	C:30 M:53	C:30 M:82	C:54 M:105
Hallucination	C:52 M:85	C:34 M:96	C:47 M:67	C:68 M:74	C:62 M:85	C:29 M:68
Inconsistent Behavior	C:24 M:194	C:19 M:207	C:26 M:186	C:33 M:139	C:40 M:197	C:52 M:194
Incorrect Interpretation	C:1 M:50	C:6 M:30	C:2 M:36	C:8 M:49	C:7 M:38	C:3 M:31
Interruption Error	C:0 M:0	C:0 M:0	C:0 M:0	C:0 M:0	C:0 M:0	C:0 M:0
Irrelevant Tool Call	C:0 M:0	C:0 M:1	C:0 M:1	C:0 M:4	C:0 M:0	C:0 M:0
Missed Required Action	C:3 M:24	C:3 M:9	C:6 M:16	C:1 M:22	C:3 M:12	C:2 M:29
Other	C:1 M:10	C:1 M:9	C:0 M:13	C:1 M:9	C:0 M:28	C:0 M:15
Premature Termination	C:13 M:16	C:9 M:24	C:12 M:20	C:10 M:18	C:14 M:21	C:17 M:16
Revealed Info Early	C:0 M:0	C:0 M:0	C:0 M:2	C:0 M:1	C:1 M:3	C:0 M:3
Tool Call Argument Error	C:0 M:3	C:2 M:7	C:0 M:2	C:2 M:1	C:1 M:0	C:1 M:2
Tool Call Schema Error	C:0 M:1	C:5 M:6	C:0 M:0	C:0 M:2	C:0 M:8	C:0 M:8
Wrong Sequence	C:0 M:0	C:0 M:0	C:0 M:11	C:4 M:6	C:2 M:5	C:0 M:4

Table 47: User error tag counts — **Telecom** domain, English baseline and L2 Interaction settings (gpt-5-mini agent, qwen3-235b user simulator, DeepSeek-V4-Flash judge, $n=342$ per condition). **C:** = critical-severity count; **M:** = minor-severity count. For the simulated user, **C:** pools critical_helped and critical_hindered. Counts are judge-annotated error instances pooled over all simulations; a simulation may contribute several instances and an instance may carry several tags, so column sums can exceed n .

Error Tag	Tool Adaptation — Single-language				
	VI	TH	ID	ZH	FIL
Guideline Violation	C:33 M:52	C:36 M:61	C:16 M:74	C:19 M:64	C:35 M:58
Hallucination	C:45 M:91	C:48 M:88	C:51 M:78	C:49 M:55	C:32 M:60
Inconsistent Behavior	C:17 M:186	C:47 M:175	C:15 M:180	C:22 M:160	C:23 M:135
Incorrect Interpretation	C:4 M:30	C:6 M:36	C:5 M:49	C:6 M:52	C:4 M:46
Interruption Error	C:0 M:0	C:0 M:0	C:0 M:0	C:0 M:0	C:0 M:0
Irrelevant Tool Call	C:0 M:1	C:0 M:1	C:0 M:8	C:0 M:3	C:0 M:10
Missed Required Action	C:2 M:11	C:2 M:30	C:2 M:21	C:5 M:16	C:9 M:11
Other	C:0 M:15	C:1 M:3	C:0 M:12	C:0 M:32	C:0 M:11
Premature Termination	C:5 M:23	C:3 M:14	C:12 M:20	C:8 M:15	C:11 M:14
Revealed Info Early	C:0 M:1	C:0 M:1	C:0 M:0	C:0 M:4	C:0 M:15
Tool Call Argument Error	C:0 M:10	C:0 M:5	C:1 M:9	C:0 M:1	C:0 M:2
Tool Call Schema Error	C:0 M:3	C:0 M:6	C:0 M:5	C:0 M:2	C:0 M:4
Wrong Sequence	C:0 M:7	C:1 M:2	C:0 M:6	C:2 M:3	C:0 M:3

Table 48: User error tag counts — **Telecom** domain, Tool Adaptation (single-language) settings.

Error Tag	Tool Adaptation — Mixed-language			
	Mix-2	Mix-3	Mix-4	Mix-5
Guideline Violation	C:15 M:49	C:13 M:52	C:15 M:77	C:12 M:56
Hallucination	C:48 M:93	C:40 M:95	C:38 M:69	C:74 M:83
Inconsistent Behavior	C:28 M:158	C:32 M:209	C:23 M:169	C:14 M:184
Incorrect Interpretation	C:4 M:38	C:7 M:35	C:4 M:39	C:4 M:29
Interruption Error	C:0 M:0	C:0 M:0	C:0 M:0	C:0 M:0
Irrelevant Tool Call	C:0 M:0	C:0 M:1	C:0 M:3	C:0 M:0
Missed Required Action	C:3 M:22	C:2 M:8	C:4 M:17	C:1 M:21
Other	C:1 M:9	C:0 M:16	C:1 M:10	C:1 M:10
Premature Termination	C:11 M:15	C:12 M:17	C:15 M:14	C:5 M:16
Revealed Info Early	C:1 M:1	C:1 M:1	C:0 M:0	C:0 M:0
Tool Call Argument Error	C:1 M:4	C:4 M:8	C:0 M:1	C:0 M:1
Tool Call Schema Error	C:0 M:3	C:0 M:1	C:0 M:5	C:0 M:1
Wrong Sequence	C:0 M:10	C:0 M:1	C:0 M:5	C:0 M:2

Table 49: User error tag counts — **Telecom** domain, Tool Adaptation (mixed-language) settings.

Error Tag	L2 Domain				
	VI	TH	ID	ZH	FIL
Guideline Violation	C:13 M:42	C:17 M:46	C:11 M:59	C:23 M:85	C:22 M:85
Hallucination	C:39 M:48	C:49 M:56	C:57 M:54	C:37 M:61	C:52 M:47
Inconsistent Behavior	C:18 M:146	C:16 M:151	C:20 M:140	C:10 M:115	C:20 M:142
Incorrect Interpretation	C:3 M:30	C:7 M:34	C:8 M:25	C:6 M:28	C:7 M:42
Interruption Error	C:0 M:0	C:0 M:0	C:0 M:0	C:0 M:0	C:0 M:0
Irrelevant Tool Call	C:0 M:1	C:0 M:0	C:0 M:1	C:0 M:3	C:0 M:3
Missed Required Action	C:6 M:8	C:2 M:12	C:1 M:9	C:3 M:14	C:2 M:9
Other	C:1 M:11	C:0 M:33	C:0 M:20	C:1 M:8	C:0 M:27
Premature Termination	C:11 M:23	C:17 M:12	C:14 M:29	C:9 M:18	C:11 M:31
Revealed Info Early	C:1 M:0	C:1 M:1	C:1 M:2	C:0 M:2	C:0 M:1
Tool Call Argument Error	C:0 M:5	C:0 M:2	C:0 M:1	C:0 M:3	C:0 M:2
Tool Call Schema Error	C:0 M:0	C:0 M:1	C:0 M:22	C:1 M:1	C:0 M:2
Wrong Sequence	C:0 M:2	C:1 M:1	C:0 M:1	C:0 M:1	C:0 M:1

Table 50: User error tag counts — **Telecom** domain, L2 Domain settings.