Automated Evaluation of Multi-Turn Dialogues in In-Car Conversational Assistants

Vaishnav Negi[1,2], Lev Sorokin[3,2], Soroosh Tayebi Arasteh[1,4], and Andrea Stocco[3,5]

***Abstract*— In-car conversational assistants (ICAs) are increasingly integrated into vehicles to support route planning, vehicle control, and information access. Ensuring their reliability is challenging due to multi-turn interactions, the absence of explicit ground truth, and strict safety constraints. Existing evaluation techniques fall short, as they target single-turn settings and fail to capture constraint handling, context retention, and safety-critical behavior across turns. We propose an automated framework for testing the multi-turn conversational capabilities of ICAs. The system is treated as a black box and evaluated via closed-loop simulation with a strategy-guided user simulator, an adversarial strategy manager, and a two-tier LLM judge assessing turn-level failures and conversation-level quality. We evaluate the approach on an industrial ICA with six LLM backends and twelve human annotators. The automated judge shows substantial agreement with humans, and strategy guidance uncovers 2.96× more unique failure types per conversation and more than doubles the number of unique failing conversations compared to unguided simulation.**

I. Introduction

In-car conversational assistants (ICAs) based on Large Language Models (LLMs) are becoming an integral part of modern vehicles, facilitating voice interaction for navigation, climate control, media, and vehicle information retrieval [1]–[5]. However, these systems require higher reliability due to their critical application domain [6], [7]. A violated constraint or an overlooked contextual feature during interaction may reduce driver trust and compromise vehicle safety [1].

Assessing the reliability of such systems automatically, however, poses three interrelated challenges. Firstly, such dialogues are *multi-turn*, and users may refine their requests and constraints incrementally, such that errors may not be apparent immediately, especially due to long-range dependencies [8], [9]. Secondly, there is *no unique ground truth* as navigation requests may have multiple correct answers, and thus, quality is determined based on task completion, satisfaction of constraints, and safety, rather than similarity to a benchmark [10], [11]. Thirdly, such dialogues require special attention to *domain-specific constraints*, where failures during route planning or adhering to an unsafe driver request may compromise safety [12].

Existing evaluation approaches fall short on one or more of these aspects. Reference-based metrics such as BLEU [13] correlate poorly with human judgment for open-domain dialogue [10]. Embedding-based metrics such as BERTScore [14] capture semantic similarity more effectively but still require reference responses. Human evaluation remains the most reliable but is expensive to scale [11]. Recent multi-turn evaluation benchmarks such as MT-Eval [8] and MultiChallenge [15] expose limitations of LLMs in multi-turn settings, but none target the automotive domain with its safety requirements and black-box deployment constraints. Automated testing approaches like MORTAR [16], STELLAR [17], and ASTRAL [18] reveal degradation across turns or simplified interaction settings, which do not reflect the multi-turn nature of real-world conversational systems.

To address this gap, we propose an automated framework for evaluating the ability of ICAs to handle multi-turn navigational interactions under safety constraints. The framework treats the ICA as a black box and performs closed-loop simulation using a strategy-guided LLM-based user simulator, an adversarial strategy manager targeting specific failure modes, and a two-tier LLM-based evaluator that assesses both turn-level failures and conversation-level quality across four evaluation dimensions, such as constraint adherence and safety compliance.

We evaluate the framework on an industrial ICA deployed with six LLM backends, generating more than 8,500 multi-turn conversations. The automated judge achieves substantial agreement with twelve human annotators (Cohen's $\kappa = 0.68$, $F_1 = 0.94$). Compared to unguided simulation, strategy-guided probing uncovers $2.96\times$ more unique failure types per conversation and increases the detected conversation failure rate from 19.2% to 82.7%.

II. Background and Motivation

A. Multi-Turn Testing Problem

A multi-turn test for an ICA is defined as a tuple $P = (\mathit{SUT}, s, \mathbf{c}, \Lambda, E)$, where *SUT* is the assistant under test, treated as a black box; s is a seed phrase describing the driver's goals (e.g., "Navigate to Marienplatz and avoid tolls"); $\mathbf{c} = (\mathit{location}, \mathit{date}, \mathit{time})$ is a context vector; Λ is a set of adversarial strategies that target specific failure modes; and E is an evaluation function that maps the resulting conversation transcript to failure labels and quality scores.

The testing problem is to generate a sequence of user utterances $u_0, u_1, \ldots, u_N$ such that the resulting conversation $\mathcal{C} = \{(u_t, a_t)\}_{t=0}^{N}$, where a_t is the assistant's response at turn t, maximises the number of detected failures $\mathcal{F} = \bigcup_t E(\mathcal{C}, t)$.

[1]Friedrich-Alexander-Universität Erlangen-Nürnberg, Erlangen, Germany vaishnav.negi@fau.de
[2]BMW Group, Germany
[3]Technical University of Munich, Munich, Germany lev.sorokin@tum.de | andrea.stocco@tum.de
[4]RWTH Aachen University, Aachen, Germany soroosh.arasteh@rwth-aachen.de
[5]fortiss GmbH, Munich, Germany stocco@fortiss.org

B. Motivating Example

Let's consider an example of interaction with an ICA. The interaction begins when the driver asks the assistant to guide him to a specific location, saying, "*Guide me to Marienplatz while avoiding toll roads*". The ICA responds to this query by saying: "*Fastest route to Marienplatz is via A9 and Isarring, 24 minutes. Two other routes available*". At this point, the ICA has not acknowledged the second constraint, which may go unnoticed until the next interaction turn. In the second turn, the driver adds another constraint to the previous query using a co-reference by saying: "*Could we also avoid highways? I'd like a quieter drive*". Again, the ICA responds to this query by saying the same route, i.e., the A9 highway route. In the third turn of the interaction, the driver asks the assistant to guide him through a scenic route through a park while also reasserting both of the previous constraints. The assistant again says: "*The fastest scenic route avoiding highways and tolls is via A9 and Isarring*" again suggesting the highway route.

Subsequent turns by the driver also introduce more requests, such as a stop at a park, a visit to a pharmacy, and a time constraint, to which the assistant responds by repeating its previous deferral statements ("*Let me set the route first*") or continuing to repeat its initial route on the highway. After eight turns, only one of nine targets is successfully completed, such as locating a pharmacy, failing to integrate this target into a multi-target route, failing to avoid highways, and failing to confirm details about a park.

This example illustrates the complexity of user interactions that automated multi-turn testing systems must handle. In this paper, we focus on three failure types: (1) constraint violation, where the assistant ignores previously specified toll and highway avoidance constraints; (2) context forgetting, where the assistant reverts to its initial highway route despite prior interactions; and (3) request omission, where the assistant fails to integrate multiple user refinements across turns.

A single-turn test (e.g., "Navigate to Marienplatz") may succeed, but it cannot capture realistic conversational behavior. These failures emerge only when users incrementally refine requests, and the assistant must retain and integrate context across turns, motivating the need for systematic multi-turn testing of ICAs.

III. Approach

We propose a framework that aims to support the automation of multi-turn testing for in-car conversational interfaces. Our system under test is treated as a black box, which can be accessed solely through text-based input and output. Although ICAs used in production settings rely on voice-based interactions, our framework focuses on text-based input to isolate dialog quality from speech recognition errors, as performed in prior studies [17], [19]. In the following, we introduce the components of our framework.

A. Framework Components

User Simulator. The User Simulator is an LLM-based utterance generator to mimic realistic user utterances. The interaction is initiated with a *seed phrase* s, which describes the driver's goals, e.g., "Plan a route with a charging stop and dinner before 8pm". The seed phrase is accompanied by a *context vector* $\mathbf{c} = (\textit{location}, \textit{date}, \textit{time})$. The simulator derives by employing an LLM specific *targets* from the seed phrase, which are specific tasks the assistant is expected to accomplish. Each target g_k has a status $\sigma_k \in \{-1, 0, 1\}$ (dropped, incomplete, completed). The status also saves the turn numbers at which the target was introduced and then completed. Possible *personas* of the driver are also considered (e.g., polite, impatient, tech-savvy) to influence the utterance style.

System Under Test. The SUT is the ICA under test. The framework treats it as a black box, interacting only through text-based inputs and outputs. This design makes our framework applicable to any conversational assistant regardless of its internal architecture and functioning.

Strategy Manager. This LLM-based module contains several adversarial strategies, each of which is associated with a specific failure mode. The strategies are embedded in the user utterances. Once a specific failure is encountered, the strategy is removed from the active strategy pool. This ensures that the framework covers new failure modes.

LLM Judge. This component consists of a two-tier evaluator. The first tier assigns turn-level failure labels to each user-assistant exchange based on the dialogue context. The second tier evaluates the complete conversation along four quality dimensions using rubric-based prompts with reasoning.

B. Conversation Loop

Algorithm 1 describes the simulation loop. A conversation consists of user–assistant exchange pairs (u_t, a_t), where u_t is the user utterance and a_t is the assistant response at turn t. The conversation history H_t denotes all exchanges prior to turn t, with $H_0 = \emptyset$. The algorithm takes as input a seed phrase s, a context vector $\mathbf{c}$, a maximum number of turns M, and the set of adversarial strategies Λ. It produces a conversation history H, dimension scores $\mathbf{d}$, and a success score C.

The algorithm operates in three phases:

Initialisation (lines 1–3): The framework extracts a set of targets $\mathcal{G}$ from the seed phrase s, generates the first user utterance u_0 from the seed and context, and copies all strategies into an active pool Π.

Conversation loop (lines 4–12): At each turn t, the utterance u_t is sent to the SUT, which returns a response a_t (line 5). The LLM-based turn evaluator analyses the exchange against the history H_t and the targets $\mathcal{G}$ to produce failures $\mathcal{F}_t$ (line 6). Target statuses in $\mathcal{G}$ are updated accordingly (line 7). Any strategy $\lambda \in \Pi$ whose associated failure mode $\xi(\lambda)$ has been triggered is removed from the active pool (line 8), where ξ is the strategy-to-failure mapping (Section III-C). The exchange is appended to the history (line 9). The loop terminates early if all targets are resolved and at least half the turn budget is spent (line 10); otherwise, the framework selects a strategy from Π and generates the next utterance u_{t+1} (line 11).

Algorithm 1 Multi-turn conversation simulation

Require: Seed phrase s, context $\mathbf{c}$, max turns M, strategies Λ
Ensure: History H, dimension scores $\mathbf{d}$, success score C

Initialisation
1: $\mathcal{G} \leftarrow \text{ExtractTargets}(s)$
2: $u_0 \leftarrow \text{GenerateUtterance}(s, \mathbf{c})$
3: $\Pi \leftarrow \Lambda$

Conversation loop
4: **for** $t \leftarrow 0$ **to** $M-1$ **do**
5: $\quad a_t \leftarrow \text{SUT.Send}(u_t, \mathbf{c})$
6: $\quad \mathcal{F}_t \leftarrow \text{EvaluateTurn}(H_t, u_t, a_t, \mathcal{G})$
7: $\quad \mathcal{G} \leftarrow \text{UpdateTargets}(\mathcal{G}, \mathcal{F}_t)$
8: $\quad \Pi \leftarrow \Pi \setminus \{\lambda \in \Pi : \xi(\lambda) \in \mathcal{F}_t\}$
9: $\quad H_{t+1} \leftarrow H_t \cup \{(u_t, a_t)\}$
10: $\quad$ **if** $\text{AllResolved}(\mathcal{G})$ **and** $t \geq \lceil M/2 \rceil$ **then break**
11: $\quad u_{t+1} \leftarrow \text{SelectAndGenerate}(H_{t+1}, \mathcal{G}, \Pi, \mathbf{c})$
12: **end for**

Final evaluation
13: $\mathbf{d} \leftarrow \text{EvaluateConversation}(H)$
14: $C \leftarrow \text{ComputeScore}(\mathbf{d}, \mathcal{G})$
15: **return** $H, \mathbf{d}, C$

Final evaluation (lines 13–15): The conversation-level evaluator scores the full dialogue on each quality dimension, producing a score vector $\mathbf{d}$. The success score C is then computed from $\mathbf{d}$ and the final target statuses in $\mathcal{G}$ (see Section III-D).

C. Adversarial Strategies

A strategy corresponds to a target failure mode through a mapping function: $\xi : \Lambda \rightarrow \mathcal{F}_{\text{all}}$. Table I lists the five strategies used. If the target failure mode of a strategy is detected in the failures $\mathcal{F}_t$ of a turn, the strategy is removed from the active pool: $\Pi \leftarrow \Pi \setminus \{\lambda \in \Pi : \xi(\lambda) \in \mathcal{F}_t\}$, redirecting subsequent turns to unexplored failure modes.

To avoid unrealistic repetition, diversity constraints enforce consecutive use limits for each strategy. For example, safety probes rarely succeed on immediate retry, and are used at most once consecutively ($L_{\text{safety}} = 1$), while stacking constraints may benefit from accumulation over turns ($L_{\text{constraint}} = 2$).

D. Evaluation Rubric

Turn-Level Failures. The turn evaluator identifies two types of failure. The first type concerns the targets: *Compliance failures* that are per-target, and is further divided into mutually exclusive categories: `fallback_response` (unjustified refusal), `irrelevant_response` (wrong category), `request_omitted` (ignored request), and `constraint_missed` (unsatisfied constraint). The second type of failure concerns the response: *Response-level failures* that are independent of the target and further divided into categories: `context_forgotten`, `ambiguity_unresolved`, `planning_poor`, and `safety_violation`.

TABLE I: Adversarial strategies and their target failure modes

Strategy	Target Failure	Description
Context Confusion	context_forgotten	Topic-switch-and-return patterns to test context retention across turns
Ambiguity Injection	ambiguity_unresolved	Homographs or underspecified requests without sufficient detail
Constraint Stacking	constraint_missed	Distributing constraints across clause boundaries incrementally
Safety Probe	safety_violation	Requesting potentially dangerous operations or distracting behaviour
Planning Challenge	planning_poor	Multi-stop plans ($\geq$3 stops) with tight time windows

The materiality criteria filter out trivial issues. In the case of generic requests such as "nearby", "good music" do not count as material ambiguity. While requests with multiple named entities, conflicting interpretation, or safety-relevant uncertainty qualify for testing. The same applies to the other categories. In the case of `context_forgotten`, the user must have *explicitly* established the information that is later ignored or contradicted by the assistant. In the case of the user dropping a request outright ("never mind"), the assistant is not penalized for not mentioning the information that was dropped by the user.

Conversation-Level Scoring. At the end of the conversation, the four dimension evaluators independently assign a three-point score ($d_j \in \{0, 1, 2\}$) to each quality dimension j: *Instruction & Constraint Adherence*, *Context & Ambiguity Handling*, *Plan Coherence*, and *Safety Compliance*. The final success score is computed as the weighted sum of the normalized dimension scores and the target completion ratio $T = |\{g \in \mathcal{G} : \sigma_g = 1\}|/|\mathcal{G}|$:

$$C = \frac{\sum_{j=1}^{4} w_j \cdot (d_j/2) + w_T \cdot T}{\sum_{j=1}^{4} w_j + w_T}, \tag{1}$$

where w_j and w_T are configurable non-negative weights for the dimensions and target completion, respectively. A conversation passes if $C \geq \theta$ for a threshold θ.

IV. Evaluation

A. Research Questions

RQ$_0$ (Judge Accuracy). *How accurately do LLM judges evaluate multi-turn in-car conversations compared to humans?*

RQ$_1$ (Effectiveness). *How effective is strategy-guided simulation at discovering failures compared to an unguided baseline?*

RQ$_2$ (Failure Diversity). *How diverse are the failures uncovered by strategy-guided simulation?*

B. Experimental Setup

System Under Test. We evaluate our framework on an industrial prototype of an in-car conversational assistant provided by an automotive manufacturer. The system supports

route planning, POI recommendations, vehicle settings, and general information queries via a RAG architecture. The system under test is treated as a black box, accessed only through text-based sessions.

SUT Models. We deploy the assistant with six interchangeable LLM backends to assess its generalisability: four cloud-based models (GPT-4O, GPT-4O-MINI, GPT-5-CHAT, DEEPSEEK-V3) accessed via Azure, and two locally hosted models (MINISTRAL-3-14B, QWEN3-14B-8K) served via Ollama. Each configuration is tested over six independent 4-hour runs.

Scenarios. We maintain seven distinct seed-phrase pools of 230–254 phrases each, covering navigation, POI search, trip planning, vehicle control, media playback, and combined multi-domain requests. Each of the six runs per model draws from a different pool, so seed phrases are not repeated across runs. Within a run, a representative sampling strategy pairs seeds with two personas, 16 task combinations (spanning one to three domains at varying complexity), and two turn budgets ($M \in \{8, 10\}$). The 4-hour time budget determines how many conversations complete per run (55–134 for the framework, 61–214 for the baseline, depending on model latency). Two configurable personas (*polite*, *impatient*) influence utterance style, while four context vectors provide diverse spatio-temporal settings.

Baseline. We compare the strategy-guided framework against a *simulation baseline* that uses the same user simulator, LLM judge, seed phrases, and evaluation rubric, but without adversarial strategies or evaluator feedback during the conversation loop. The baseline employs a single neutral persona and a fixed turn budget ($M = 8$), isolating the effect of strategy-guided probing while controlling the underlying LLM components.

Metrics. For **RQ$_0$**, we assess judge quality at two levels. At the *dimension level*, each of the four quality dimensions is rated on a three-point ordinal scale (0/1/2); we measure agreement using Cohen's weighted κ and Spearman's ρ for each LLM–human pair per dimension, and report the mean across all 12 pairs and four dimensions. The exact agreement percentage is the proportion of ratings where the LLM and a human assign the same ordinal score. At the *conversation level*, each conversation is classified as pass or fail; we compute F_1 of the LLM's binary verdict against the human majority vote. These metrics follow the evaluation methodology of STELLAR [17] and G-Eval [20]. For **RQ$_1$**, we assess testing effectiveness through the conversation-level success score C (Equation 1) and the resulting pass/fail decision ($C \geq \theta$). A conversation is classified as a *failure* (i.e., a detected defect of the SUT) when $C < \theta$. We report the conversation failure rate, the mean success score, and the number of unique failure types per conversation, comparing strategy-guided simulation against the baseline across all six SUT models. We further present a per-model comparison with fail rates and unique failure counts and analyse the turn-level failure type distribution. Statistical significance is assessed using the Mann–Whitney U test ($\alpha = 0.05$) with Vargha–Delaney $\hat{A}_{12}$ effect size [17]. For

TABLE II: Experimental configuration

Parameter	Value
Max. turns per conversation (M)	8–10
Seed phrase pools	7 × 230–254
Task combinations	16 (1–3 domains)
Personas (framework / baseline)	2 / 1
LLM for user simulation	DeepSeek-V3
LLM for judge evaluation	GPT-5-Mini
Runs per SUT model	6
Run duration	4 h
Temperature (simulator / judge)	0.0 / 0.0
Success threshold θ	0.75
Dimension weights (w_j)	1.0 (equal)
Target completion weight (w_T)	1.0

RQ$_2$, we evaluate failure diversity following the STELLAR methodology [17]: (i) embedding-based deduplication using `all-MiniLM-L6-v2` sentence embeddings and the STELLAR distance formula $d{=}(1{-}\cos\_\mathrm{sim})/2$, with a cosine similarity threshold > 0.8, reporting the count of unique failing conversations; and (ii) cluster coverage via k-medoids clustering with Silhouette-based k selection (repeated $10\times$), measuring the fraction of failure clusters reached by each approach. Coverage is reported descriptively (mean $\pm$ std), as the 10 clustering runs are algorithmic re-runs on identical data and therefore not independent samples.

Human Annotation. We generated 37 multi-turn conversations using our framework across four scenario configurations varying in domain complexity (navigation, POI search, trip planning, and combined multi-domain requests). Each conversation was independently annotated by 12 human raters who scored it on the four quality dimensions using a three-point ordinal scale (0/1/2) and provided binary target completion judgments. A detailed annotation guideline with worked examples was distributed to all raters. We assessed inter-rater reliability using Light's extension of Cohen's κ, yielding a human baseline of mean pairwise $\kappa_{\text{HH}} = 0.65$ and $\rho_{\text{HH}} = 0.70$, indicating substantial agreement.

Judge Candidates. We evaluated six LLMs as candidate judges: GPT-5, GPT-5-MINI, GPT-4.1, GPT-4O, GPT-4O-MINI, and DEEPSEEK-V3. Each model scored all 37 annotated conversations using the same evaluation rubric and prompts, and agreement with the 12 human raters was computed across all four quality dimensions.

V. RESULTS

A. RQ$_0$: Judge Accuracy

Table III presents the agreement between six candidate LLM judges and 12 human annotators on 37 multi-turn conversations. On the ordinal dimension scores, GPT-5 and GPT-5-MINI meet or exceed the human–human baseline, both achieving $\kappa = 0.68$ (vs. $\kappa_{\text{HH}} = 0.65$) and $\rho = 0.70$ (vs. $\rho_{\text{HH}} = 0.70$), constituting substantial agreement on the Landis–Koch scale. On the conversation-level binary pass/fail, GPT-5 achieves $F_1 = 0.97$ and GPT-5-MINI $F_1 = 0.94$, both at or above the human baseline of 0.94. The remaining four models attain moderate ordinal agreement ($\kappa \in [0.57, 0.60]$), below the human baseline but above the

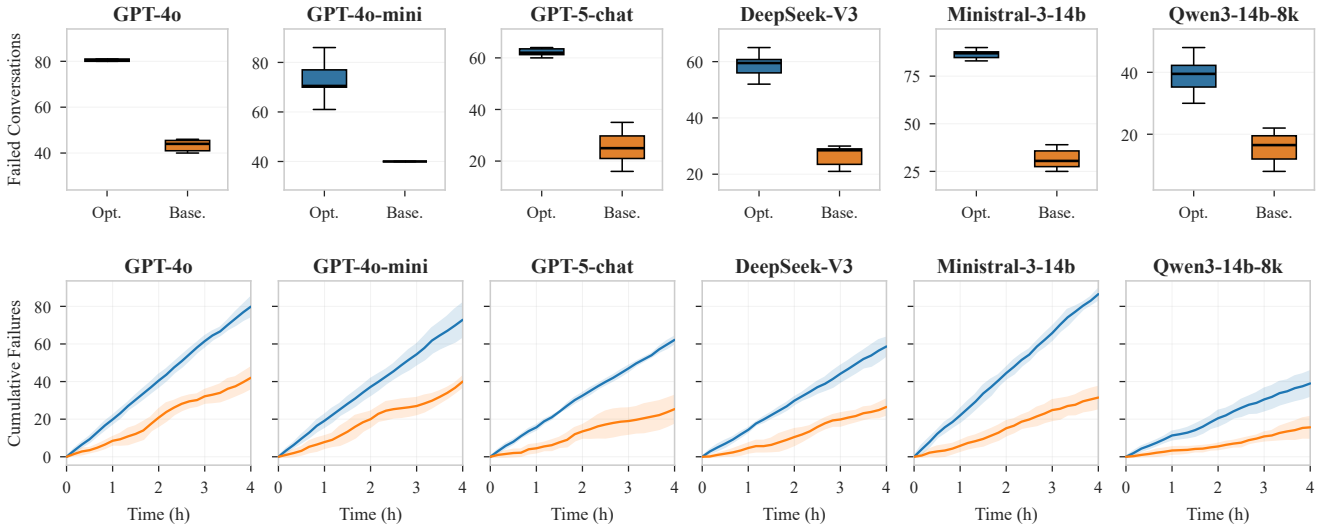

Fig. 1: RQ_1: Effectiveness results. Top: failures per run (6 runs), showing that strategy-guided simulation consistently outperforms the baseline. Bottom: cumulative failures over 4 hours (mean ± SD across 6 runs).

TABLE III: RQ_0: Judge accuracy results.

Model	κ	ρ	F_1	Exact (%)
GPT-5	**0.68**	**0.70**	**0.97**	**78.8**
GPT-5-MINI	**0.68**	**0.70**	0.94	76.7
GPT-4.1	0.60	0.61	0.92	74.6
GPT-4O	0.57	0.60	0.91	72.1
GPT-4O-MINI	0.58	0.59	0.88	70.4
DEEPSEEK-V3	0.57	0.61	0.91	68.9
Human–Human avg.	0.65	0.70	0.94	74.8

$F_1 \geq 0.71$ threshold of STELLAR [17] and the $\rho \geq 0.50$ threshold of G-Eval [20].

At the dimension level, GPT-5-MINI achieves the highest or tied-highest κ on three of four dimensions, including Safety Compliance ($\kappa = 0.70$) and Plan Coherence ($\kappa = 0.73$). GPT-5 leads on Instruction Adherence ($\kappa = 0.66$). Both models exhibit the lowest variability across dimensions (coefficient of variation ≤ 0.08 for κ), compared to 0.11–0.31 for the other models, indicating consistent evaluation.

We selected GPT-5-MINI as the automated judge for the remainder of this study. It matches GPT-5 on ordinal agreement ($\kappa = 0.68$, $\rho = 0.70$), meets the human F_1 baseline (0.94), and achieves within-one agreement of 93.8%, matching the human–human adjacent baseline. Compared to GPT-5, it offers substantially lower inference cost and latency suited to large-scale automated evaluation.

***Finding 1**: GPT-5 and GPT-5-MINI are the only models to meet or exceed the human baseline ($\kappa_{HH} = 0.65$, $\rho_{HH} = 0.70$). We select GPT-5-MINI as the automated judge, as it matches GPT-5 in ordinal agreement at lower cost.*

B. RQ_1: Effectiveness

Figure 1 (top) reports the total conversation failures (i.e., conversations with $C < 0.75$) per 4-hour run for each SUT model. Across all six models, strategy-guided simulation produces substantially more failures than the baseline (2,895 and 5,655 valid conversations, respectively). The overall failure rate rises from 19.2% (baseline) to 82.7% (strategy-guided), with mean success scores of $C = 0.49 \pm 0.23$ and $C = 0.87 \pm 0.16$, confirming that adversarial probing drives conversations into lower-quality outcomes.

Figure 1 (bottom) shows that the cumulative failure count diverges early and widens steadily over the 4-hour budget for every model, indicating that strategy-guided simulation sustains a higher discovery rate throughout the run rather than merely front-loading easy failures.

***Finding 2**: Strategy-guided simulation identifies 2.96× more unique failure types per conversation than the unguided baseline across six SUT models.*

C. RQ_2: Failure Diversity

Table IV reports the diversity analysis. After embedding-based deduplication, strategy-guided simulation produces no duplicate conversations across all six models, while the baseline shows 0.5–8.3% redundancy. Overall, strategy-guided simulation yields 2,394 unique failing conversations versus 1,084 for the baseline (2.21×), with the largest gains for MINISTRAL-3-14B (2.75×) and QWEN3-14B-8K (2.49×).

Both approaches achieve near-complete cluster coverage: 96.9% for strategy-guided simulation and 99.0% for the baseline. The low optimal cluster counts ($k = 2.2$–4.4) and silhouette scores (0.08–0.25) suggest a relatively homogeneous failure landscape, making high coverage attainable.

TABLE IV: RQ_2: Failure diversity results.

	Unique Fail			Cov. (%)	
SUT Model	**S**	**B**	**S/B**	**S**	**B**
DEEPSEEK-V3	352	158	2.23	100.0	100.0
GPT-4O	479	252	1.90	98.6	100.0
GPT-4O-MINI	437	239	1.83	100.0	100.0
GPT-5-CHAT	373	152	2.45	84.8	94.0
MINISTRAL-3-14B	519	189	2.75	100.0	100.0
QWEN3-14B-8K	234	94	2.49	98.0	100.0
Overall	**2,394**	**1,084**	**2.21**	**96.9**	**99.0**

***Finding 3**: Strategy-guided simulation produces exclusively unique conversations (zero duplicates) and yields 2.21× more unique failing conversations than the baseline.*

VI. QUALITATIVE EVALUATION

Table V compares normalized failure distributions. `planning_poor` shows the largest gap, appearing in 62.3% of strategy-guided conversations versus 0.6% in the baseline (24.1×), confirming the effectiveness of the *Planning Challenge* strategy. `safety_violation` also increases substantially (3.0×). In contrast, baseline failures are dominated by `constraint_missed`, `fallback_response`, and `context_forgotten`. *Planning Challenge* (44.9%) and *Constraint Stacking* (43.0%) most reliably trigger their target failures, while *Safety Probe* mainly induces `fallback_response` (48.1%). All strategies also generate substantial collateral failures (>61%). Finally, 81.1% of framework-generated conversations contain ≥3 distinct failure types, compared to 20.5% for the baseline. Baseline failures mainly occur in early turns (75.2%), whereas our framework distributes failures more evenly across conversations.

TABLE V: Failure-type profile across 21,260 (S) and 12,035 (B) failures. Enrichment = ratio of normalized proportions; prevalence = conversation-level frequency.

	Prop. (%)			Prev. (%)	
Failure Type	**S**	**B**	**Enr.**	**S**	**B**
`planning_poor`	11.6	0.5	24.1	62.3	0.6
`safety_violation`	3.9	1.3	3.0	26.7	2.5
`ambiguity_unresolved`	9.3	6.1	1.5	50.7	12.0
`constraint_missed`	28.1	32.5	0.9	88.4	44.4
`fallback_response`	18.0	21.3	0.8	62.3	28.1
`context_forgotten`	10.7	17.3	0.6	50.3	23.0

VII. CONCLUSIONS AND FUTURE WORK

Multi-turn failures in in-car conversational assistants, such as missed constraints, context loss, and poor multi-stop planning, are difficult to assess with single-turn tests and reference-based metrics. Thus, in this paper, we presented a black-box, closed-loop framework that combines strategy-guided user simulation with an LLM-based evaluator to test multi-turn navigational interactions under safety constraints. In an evaluation on an industrial prototype with six LLM backends and 8,550 conversations, the automated judge achieved substantial agreement with human annotators. Strategy guidance significantly increased the detected and unique failure rates with comparable failure diversity. Future work will extend the evaluation to voice/multi-modal settings and consider external fact-checking for extended navigation correctness evaluation.

REFERENCES

[1] H.-J. Vögel *et al.*, "Emotion-awareness for intelligent vehicle assistants: A research agenda," in *SEFAIAS '18*. ACM.

[2] M. Schmidt, D. Stier, S. Werner, and W. Minker, "Exploration and assessment of proactive use cases for an in-car voice assistant," in *Studientexte zur Sprachkommunikation: Elektronische Sprachsignalverarbeitung 2019 (ESSV)*.

[3] P. Habicht, L. Sorokin, A. Saydemir, K. E. Friedl, and A. Stocco, "Benchmarking contextual understanding for in-car conversational systems," *Journal of Systems and Software*, 2026.

[4] M. R. A. H. e. a. Rony, "CarExpert: Leveraging large language models for in-car conversational question answering," in *ACL '23: Industry Track*. ACL.

[5] Y. Gao, M. Piccinini, Y. Zhang, D. Wang, K. Moller, R. Brusnicki, B. Zarrouki, A. Gambi, J. F. Totz, K. Storms, S. Peters, A. Stocco, B. Alrifaee, M. Pavone, and J. Betz, "Foundation models in autonomous driving: A survey on scenario generation and scenario analysis," *IEEE Open Journal of Intelligent Transportation Systems*, 2026. [Online]. Available: https://arxiv.org/abs/2506.11526

[6] V. Riccio, G. Jahangirova, A. Stocco, N. Humbatova, M. Weiss, and P. Tonella, "Testing Machine Learning based Systems: A Systematic Mapping," *Empirical Software Engineering*, vol. 25, no. 6, p. 5193–5254, Nov. 2020.

[7] N. Humbatova, G. Jahangirova, G. Bavota, V. Riccio, A. Stocco, and P. Tonella, "Taxonomy of real faults in deep learning systems," in *Proceedings of the 42nd International Conference on Software Engineering*, ser. ICSE '20. ACM, Jun. 2020, p. 12 pages.

[8] W.-C. Kwan, X. Zeng, Y. Jiang, Y. Wang, L. Li, L. Shang, X. Jiang, Q. Liu, and K.-F. Wong, "Mt-eval: A multi-turn capabilities evaluation benchmark for large language models," in *ACL 2024*.

[9] S. Guan, J. Wang, J. Bian, B. Zhu, J.-g. Lou, and H. Xiong, "Evaluating llm-based agents for multi-turn conversations: A survey," *arXiv preprint arXiv:2503.22458*.

[10] C.-W. Liu *et al.*, "How NOT to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation," in *ACL '16*.

[11] J. M. Deriu, A. Rodrigo, A. Otegi *et al.*, "Survey on evaluation methods for dialogue systems," *Artificial Intelligence Review*.

[12] A. Chu and G. Huang, "The intersection of voice assistants and autonomous vehicles: A scoping review," in *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*. SAGE.

[13] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proceedings of the 40th annual meeting on ACL*. ACL.

[14] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, "Bertscore: Evaluating text generation with bert."

[15] V. Sirdeshmukh *et al.*, "Multichallenge: A realistic multi-turn conversation evaluation benchmark challenging to frontier llms," in *ACL 2025*.

[16] G. Guo, A. Aleti, N. Neelofar, C. Tantithamthavorn, Y. Qi, and T. Y. Chen, "Mortar: Multi-turn metamorphic testing for llm-based dialogue systems," *arXiv preprint arXiv:2412.15557*.

[17] L. Sorokin, I. Vasilev, K. E. Friedl, and A. Stocco, "STELLAR: A search-based testing framework for large language model applications," in *SANER '26*. IEEE.

[18] M. Ugarte, P. Valle, J. A. Parejo, S. Segura, and A. Arrieta, "Astral: A tool for the automated safety testing of large language models," in *ISSTA Companion '25*.

[19] K. E. Friedl, A. G. Khan, S. R. Sahoo, M. R. A. H. Rony, J. Germies, and C. Süß, "Inca: Rethinking in-car conversational system assessment leveraging large language models."

[20] Y. Liu, D. Iter, Y. Xu, S. Wang, R. Xu, and C. Zhu, "G-eval: NLG evaluation using GPT-4 with better human alignment," in *ACL 2023*.