CUEING USER SIMULATORS: CALIBRATED USER EMBEDDINGS FOR MULTI-TURN BENCHMARKING

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent benchmarks rely on user simulators to evaluate AI agents in multi-turn interaction. While existing simulation techniques demonstrate surface *fidelity* to human style and behavior, ecologically valid interactive benchmarking also requires alignment in when and how agents fail across simulated and real user populations. We find that established measures of simulator naturalness and mimicry provide inconsistent signals about whether simulated interactions reproduce real-user outcomes. Therefore, interactive benchmarking requires a complementary criterion, *outcome calibration*: agreement with observed success rates, conditional failure patterns, and retrospective user-task outcomes. We introduce Calibrated User Embeddings (CUE), a framework that encodes observed sessions or samples continuous representations, then decodes them into persona commands augmented with retrieved examples to steer a black-box simulator LLM. Through this, we evaluate both user-conditioned replay of past sessions and aggregate metric agreement when sampling novel personas for the same tasks. On τ^2-Bench, CUEd simulators commit fewer simulator-attributed errors and more faithfully reproduce real-user agent failure modes, aggregate success rates, and outcomes for specific task-user pairs than other persona-based simulation methods. These gains coexist with competitive user fidelity as measured using metrics established in prior work. After being fit to mostly customer support interactions, the same CUEd simulators generalize to document creation, math tutoring, and casual conversation, and remain effective across different simulator LLMs without CUE retraining.

1 INTRODUCTION

User simulators are increasingly used to benchmark agents in multi-turn interaction (Li et al., 2026). There are multiple axes of assessing the simulators themselves: *fidelity* asks whether their dialogue resembles human behavior, while *outcome calibration* asks whether interactions reproduce the agent outcomes observed with human participants. Calibration here refers to empirical agreement in outcomes, not calibration of predicted probabilities. A fluent simulator can still abandon a goal, over-cooperate, or fail to elicit agent errors that occur with humans. Prior work documents gaps in primarily surface realism (Dou et al., 2025; Seshadri et al., 2026; Zhou et al., 2026); we move beyond fidelity and aggregate success to examine the efficacy of simulators for benchmarking via their performance in role adherence, failure reproduction, and aggregate and user-specific success alignment.

Existing methods reduce generic simulator artifacts (Luo et al., 2024; Naous et al., 2026; Chopra et al., 2026) or extract profiles from observed conversations (Argyle et al., 2023; Zhu et al., 2026; Wang et al., 2025). An empirical profile bank already permits repeated simulation, but constrains the available conditioning representations to extracted profiles or their recombination. A learned continuous representation offers a shared interface for encoding observed sessions and generating additional conditioning vectors. We examine empirically whether these representations improve behavioral coverage and outcome calibration.

We introduce **Calibrated User Embeddings (CUE)**, combining a session encoder, a persona-command decoder, training-example retrieval, and a latent diffusion sampler (Figure 1). Our hypothesis is that contrasting human dialogue with generic simulator responses helps suppress simulator artifacts, while contrasts among human sessions retain behavioral differences that diversify interac-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

tions. The latent representation supports reuse and sampling of this supervision and a text interface makes it usable with black-box LLMs as the base models. Its name describes the evaluation objective rather than the training loss – CUE is trained on behavior and style, without benchmark rewards or failure labels in the training objective yet demonstrates improvements in outcome calibration over other persona methods.

Our work makes three contributions:

1. **Joint fidelity and outcome evaluation.** We compare conventional fidelity metrics (§5) with success-rate agreement, conditional failure distributions, and retrospective outcome agreement on human-anchored interactions (§6).
2. **A reusable simulation framework.** CUE supports either replaying sessions as a particular observed user or simulating new users, demonstrating improved outcome calibration over other persona-based simulators and competitive fidelity (§5, §6).
3. **Transfer to held-out domains.** A fixed CUE model reduces the writing-outcome error on SimulatorArena and supports fidelity evaluation on SimulatorArena and PRISM across base models (§7).

The code to reproduce all results (and trained CUE models useful for building interactive benchmarks) are released here[1].

2 Related Work

Interactive Benchmarks As agents increasingly engage in extended user interactions, multi-turn benchmarks have been proposed for conversational ability, task completion, coding, and software interaction (Bai et al., 2024; Chang et al., 2025; Trivedi et al., 2024; Zhou et al., 2024; Pan et al., 2025; Qian et al., 2025; Xu et al., 2026; Barres et al., 2026; Yang et al., 2023). Most rely on a user simulator to drive the interaction, but recent work documents substantial gaps in simulators' realism and aggregate success alignment (Dou et al., 2025; Hu et al., 2025; Seshadri et al., 2026; Suh et al., 2026; Zhou et al., 2026). However, these works often assume user fidelity – especially naturalness – are a strong proxy for outcome calibration. Empirically, we see that strong user fidelity performance does not necessarily ensure that a simulator is effective for multi-turn benchmarking.

User Simulation Existing approaches improve simulation along two main axes. One line of work adjusts generic LLM simulators to reduce synthesis artifacts such as excessive cooperation, verbosity, formality, or non-ambiguity (Luo et al., 2024; Vijayvargiya et al., 2026). PPol (Chopra et al., 2026) evolves persona-generation programs against aggregate realism and coverage objectives. UserLM (Naous et al., 2026) trains a dedicated, unconditioned user model via LLM fine-tuning. A second line of work conditions simulators on specific observed users, using explicit or implicit profiles to reproduce individual behavior (Argyle et al., 2023). RealUserSim (Zhu et al., 2026) extracts structured persona manuals from real conversations. USP (Wang et al., 2025) post-trains a simulator to reproduce implicit user profiles. These methods improve fidelity to users, but user-conditioned methods remain tied to observed profiles and trained simulators bind the user model to a particular base model, sacrificing the benefits of both scale and rapid model advancement.

Continuous User Representations Our work builds on research to learn representations of user-specific linguistic and behavioral variations, inducing (Rivera-Soto et al., 2021; Wegmann et al., 2022; Kantharuban et al., 2026; Patel et al., 2025). Such representations have also been used to condition downstream prediction, recommendation, and generation systems (Luo et al., 2022; Zhang et al., 2024b; Liu et al., 2025; Ning et al., 2025). CUE connects this view to user simulation; instead of representing a user as a discrete profile, command list, or simulator-specific parameterization, it learns a continuous latent representation that can encode an observed user, support generative sampling of additional session representations, and decode into natural-language prompts for steering arbitrary LLMs.

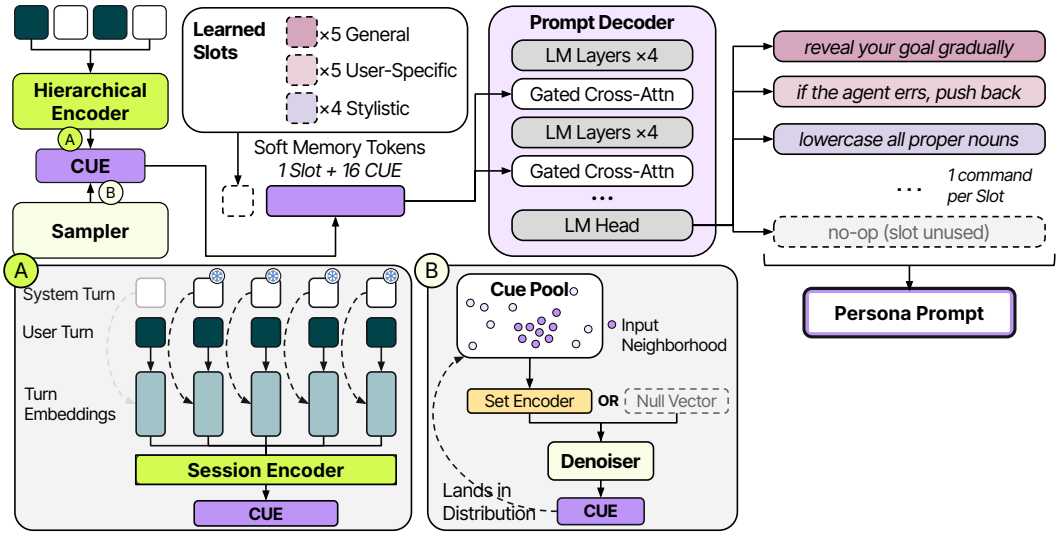


Figure 1: **The three components of the CUE framework, with two ways to generate CUEs.** (A): Encode an observed session for retrospective replay, (B): Sample a new, distributionally plausible CUE from the learned training distribution. Under either option, decoded commands and retrieved training examples augment the task-specific simulator prompt so the base model may act as the user.

3 The Calibrated User Embedding (CUE) Framework

Existing user simulators typically represent users as discrete profiles, features, or command lists that can be reused or recombined. CUE instead learns a continuous representation of the behavior and style expressed in a *session*, supporting both encoding observed interactions and sampling additional representations. The training pipeline first constructs persona-command targets from dialogue contrasts, then trains a session encoder and command decoder, and finally fits a sampler to encoded training sessions. At deployment, a CUE is obtained by encoding an observed trajectory or sampling a latent vector (Figure 1). The decoder generates persona commands, while retrieval supplies examples from training sessions. Together, these augment the benchmark's standard simulator prompt. The annotation LLM supplies training targets, the learned decoder generates persona commands, and the simulator LLM generates user turns. CUE integrates these components into a reusable framework for session-conditioned replay and sampled user simulation.

3.1 User-Conditioning for Replay as Observed Users (A)

Given a dialogue trajectory, the encoder first embeds each user turn while a separate frozen semantic encoder embeds preceding assistant dialogue turns. Each user-turn representation then attends to the preceding system-turn representation, which has been projected into the user encoder's hidden space. The contextualized turn embeddings are aggregated into a session-level CUE. Assistant information therefore enters the CUE through attention, even though the two sides are encoded separately. Content-suppression losses reduce, but cannot guarantee removal of, task and outcome information. To use this CUE with a simulator LLM, a prompt decoder maps the CUE into a set of persona commands. The CUE is expanded into a fixed soft memory bank and exposed to selected layers via gated cross-attention (Alayrac et al., 2022). Rather than generating the full persona prompt at once, learned slot embeddings are prepended to the soft memory bank to elicit general behavior, user-specific behavior, or style commands (Carion et al., 2020). During training, target commands are matched one-to-one to slots within their type, while unmatched slots are supervised toward a no-op target; at inference, the emitted commands are concatenated into the final persona prompt. This natural-language interface decouples the learned CUE from any specific simulator LLM.

[1] https://anonymous.4open.science/r/CUE-user-simulator-7403

Training uses a 200-step decoder-format warm-up followed by joint training of encoder and conditioning modules with the decoder LM frozen. Training combines slot-level command generation with representation objectives designed to preserve user-specific behavior while suppressing task content and maintaining a sampler-friendly latent geometry. We enforce consistency between original and delexicalized/content-perturbed session CUEs (Wu et al., 2022), contrast sessions to avoid collapse towards a mean persona (Wu et al., 2022), align CUE similarity with external style similarity (Wegmann et al., 2022), and regularize variance and covariance across dimensions (Bardes et al., 2022). Full training details are in Appendix B.1.

3.2 Sampling Plausible Session Representations (B)

Encoding observed trajectories yields an empirical representation bank. Sampling additional vectors offers a way to vary conditioning beyond direct bank resampling, without asserting that each vector denotes a distinct person. We therefore fit a latent diffusion model over the learned CUE distribution (Nichol & Dhariwal, 2021). At inference, it generates a new CUE that can be decoded using the same prompt decoder as observed users. Unlike direct bank resampling, diffusion generates latent vectors followed by LayerNorm-based normalization; it does not replace them with nearest bank CUEs. The sampler may operate unconditionally, over the full training distribution, or condition on a variable-size neighborhood of observed users through a permutation-invariant set encoder over CUEs. Unconditional sampling targets the learned training-session mixture, while neighborhood conditioning asks the sampler to generate another plausible CUE from a local subpopulation. Classifier-free guidance supports both modes within a single sampler (Ho & Salimans, 2021). Unless otherwise noted, evaluations use unconditional sampling, with no neighborhood and no classifier-free guidance. The training mixture is session-weighted and need not match the distribution of people in a target study. Sampler architecture and training details are in Appendix B.2.

4 Experimental Setup

4.1 Training

We use English multi-turn conversations from 24 DialogStudio datasets (Zhang et al., 2024a), LMSYS-Chat-1M (Zheng et al., 2024), and WildChat-1M (Zhao et al., 2024), balancing by capping each dataset at 10k samples. The mixture includes approximately 170k task-oriented role-play, human-human dialogue, and human-LLM conversations, not exclusively naturalistic customer support. Table 5 lists the source inventory; only sessions with accepted persona annotations supervise CUE.

For each annotated session, a proposal LLM contrasts the observed user with four simulators' counterfactual replies and other human sessions. It proposes general behavior, session-specific behavior, and style commands, with evidence-turn references; lexical and consistency gates filter the targets (Appendix A.2). This supervision encourages both correction of simulator artifacts and retention of observed behavioral variation. Two evaluated base models also occur in the contrast set, while the third (Gemini 3.5 Flash Lite) demonstrates transfer to an unseen family.

The encoder uses a ModernBERT embedding checkpoint, the decoder uses Qwen 3 0.6B and the sampler is an approximately 162M parameter MLP (Nussbaum et al., 2025; Yang et al., 2025; Ho et al., 2020). At inference, greedily decoded commands are supplemented with eight examples retrieved from nearby training sessions (two general and six user-specific). Thus, the evaluated treatment includes both learned commands and retrieval. Architecture, training, and generation details appear in Appendix B.

4.2 Evaluation

Domains Our primary evaluation uses 483 human episodes from the τ-USI airline and retail study with τ^2-Bench (Zhou et al., 2026; Barres et al., 2026). The evaluated agent is GPT 5.2 with high reasoning; the simulator base model varies independently. We additionally evaluate on SimulatorArena and PRISM, matching original agents where available and substituting with the same family and size class otherwise.

In **user-conditioned replay**, CUE encodes the completed human trajectory, including preceding assistant context, and the base model attempts the corresponding task using the resulting persona, with the shared wrapper supplying the task description (Appendix A.3). An example of this process for a single task is provided in Appendix H. In many practical settings, human interaction data are scarce but available for only a subset of cases or evaluated agent models. Here, user-conditioned simulation turns those costly observations into reusable, behaviorally grounded replay cases. Such replay provides a way to test whether a simulator reproduces the same success/failure patterns before relying on simulation at scale, including for newer agent models for which matched human trajectories are unavailable. In **sampled** evaluation, a latent vector sampled without the target human trajectory supplies the persona. This tests the learned session mixture, which is an aggregate across many past studies, against the evaluation study population, without assuming that those populations coincide.

Metrics We evaluate three fidelity axes (Table 1). **Naturalness** measures human-like behavior using a behavioral classifier (S2R-N) and a judged pairwise Turing test (TT). **Mimicry** measures paired-session similarity using authorship embeddings (AVA) and a five-dimension judge (PT3). **Coverage** measures bidirectional nearest-neighbor proximity using behavioral features (S2R-C) and style embeddings (SD-C). These are operational proxies established in prior work, described further in Appendix D.

Axis	Metric	Type	Source
Nat.	**S2R-N** ↑	Features	Chopra et al. (2026)
	TT ↓	LLM Judge	Wang et al. (2026)
Mim.	**AVA** ↑	Embedding	Wang et al. (2025)
	PT3 ↑	LLM Judge	Zhu et al. (2026)
Cov.	**S2R-C** ↑	Features	Chopra et al. (2026)
	SD-C ↑	Embedding	Patel et al. (2025)

Table 1: **User-fidelity metrics adopted from prior work**, by axis. Definitions in Appendix D.

We distinguish outcome calibration from failure attribution. **Success Rate Error** $|\Delta|$ (↓) is the absolute difference between simulated and human aggregate success rates. **Outcome F1** (↑) measures the macro-F1 of environment-assigned binary success on paired human-simulator episodes. **Failure Attribution F1** (↑) is macro-F1 over agent, user, and environment attribution on paired episodes where both human and simulator fail. **Failure Mode F1** (↑) is macro-F1 over agent failure modes where both failures are agent attributed. **Conditional Failure TVD** (↓) compares frequency distributions of agent-attributed error types. Finally, **User-Share Gap** (↓) is the absolute difference between simulated and human user-attributed shares among labeled failures, reported on the proportion scale.

Baselines We compare UserLM (Naous et al., 2026), USP (Wang et al., 2025), PPol (Chopra et al., 2026), and RealUserSim (Zhu et al., 2026) under a shared harness with matched tasks, agents, and turn budgets. CUE, USP, and RealUserSim receive paired-session conditioning; PPol receives the task only, making its mimicry and replay scores task-only references. UserLM has no individual user conditioning. We additionally compare with None when possible, which is the base model with no persona injection. We report sampled coverage for CUE, USP, RealUserSim, UserLM, and None; PPol's task-conditioned persona generator is omitted from this comparison. Prompted methods use Llama 3.1 8B, GPT 5.4 Mini, and Gemini 3.5 Flash Lite; trained baselines use their released base models.

These are end-to-end system comparisons, not matched-supervision controls. In replay, RealUserSim omits example utterances while CUE retrieves training examples because CUE examples do not come from the same session we are replaying. Their sampling populations also differ – CUE's corpus is mixed while RealUserSim is limited to a subset of WildChat. Appendix C documents adaptations and Appendix E.1 reports confidence intervals for all metrics for each method.

5 Fidelity to Real Users

Table 2 reports the user fidelity of various simulators, measured via established criteria from prior work. In fidelity, CUE is competitive with current state-of-the-art user simulation techniques.

Naturalness PPol leads the S2R-N metric by a large margin but is followed more closely by CUE in their TT values. The taxonomy used to featurize trajectories for the S2R-N metric is the

Method	Naturalness		Mimicry		Coverage	
	S2R-N ↑	TT ↓	AVA ↑	PT3 ↑	S2R-C ↑	SD-C ↑
Llama 3.1 8B						
None	– / –	0.44 / 0.44	0.04	0.19	0.01	0.45
UserLM	– / **0.56**	– / 0.41	–	–	0.62	**0.55**
USP	0.26 / 0.27	0.46 / 0.47	0.06	0.02	0.00	0.44
PPol	**0.47** / –	**0.18** / –	**0.26**	0.13	–	–
RealUserSim	0.09 / 0.09	0.30 / 0.33	0.24	0.17	0.00	0.46
CUE	0.41 / 0.48	0.22 / **0.26**	0.20	**0.21**	**0.63**	**0.55**
GPT 5.4 Mini						
None	– / –	0.44 / 0.44	0.03	0.28	0.64	0.44
PPol	**0.67** / –	**0.09** / –	**0.24**	0.12	–	–
RealUserSim	0.19 / 0.19	0.24 / **0.26**	0.23	**0.42**	0.55	0.57
CUE	0.53 / **0.55**	0.18 / **0.26**	0.20	0.33	**0.83**	**0.58**
Gemini 3.5 Flash Lite						
None	– / –	0.38 / 0.38	0.08	0.23	0.36	0.44
PPol	**0.89** / –	0.17 / –	0.29	0.05	–	–
RealUserSim	0.28 / 0.24	0.32 / 0.34	**0.30**	0.14	0.47	0.51
CUE	0.57 / **0.59**	**0.14** / **0.12**	0.27	**0.38**	**0.83**	**0.57**

Table 2: **User fidelity on τ^2-Bench.** Naturalness entries are user-conditioned/sampled, mimicry uses user-conditioned and coverage uses sampled. Bold marks the best reported value for each variation.

same as that used to optimize the PPol prompts in their algorithm, meaning it is a biased evaluator for that method specifically. Omitting PPol for that reason, we see that CUE performs far better on all but one setting than the other baselines for S2R-N. Relative to the no-persona simulator, CUE improves the TT metric across all six settings, with the no-persona simulator being nearly always distinguishable by the judge. CUE also improves on RealUserSim's naturalness despite using the same prompt structure, though differences in example access and distribution domain prevent attributing this solely to representation learning.

Mimicry PPol often leads in the AVA metric, whereas the PT3 metric favors other conditioned methods. Broadly we see that the two metrics disagree often, suggesting that mimicry is difficult to evaluate objectively. Still, replacing paired CUEs with shuffled CUEs reduces the PT3 value from 0.38 to 0.21 and AVA from 0.27 to 0.20 (Table 15). The implementation shuffles episodes within a single domain across tasks, establishing sensitivity to session pairing. Thus, the mimicry metrics are sensitive to correct CUE–session pairing, with paired CUEs producing greater similarity to the corresponding observed session than shuffled CUEs, but are not consistent in differentiation between methods.

Coverage CUE attains the best or tied-best reported coverage scores, with UserLM also being competitive. None reaches a S2R-C value of 0.64 with GPT 5.4 Mini as the base model, but only 0.01 with Llama 3.1 8B; CUE reaches 0.83 and 0.63 respectively. As such, we see that the strength of CUE is in the increased variance it encourages in the base model it is applied to. For Gemini 3.5 Flash Lite, unconditional diffusion sampling performs better in both coverage metrics than resampling from the training session bank used to create the sampler (Table 16). This does not establish that diffusion is necessary for the observed gains, but signals it is one way to diversify outputs.

6 Calibration to Real Outcomes

For human-grounded agent evaluation, a simulator should reproduce not only aggregate success rates but also which types of interactions fail and how. Table 3 compares aggregate outcome agreement with paired reproduction of human outcomes and failure labels. For τ^2-Bench programmatic rewards identify failed episodes. We take these and use LLM-assisted thematic analysis to assign an error attribution (user, agent, or environment) and, for agent-attributed failures, a failure mode

Method	Role Adherence		Failure Reproduction		Success Alignment	
	User Gap ↓	Attr. F1 ↑	Agent TVD ↓	Mode F1 ↑	Rate |Δ| ↓	Outcome F1 ↑
Llama 3.1 8B						
None	0.14 / 0.14	0.35	0.36 / 0.36	**0.22**	**0.21** / **0.21**	**0.50**
UserLM	– / 0.54	–	– / **0.31**	–	– / 0.39	–
USP	0.73 / 0.72	0.20	**0.31** / 0.93	0.05	0.54 / 0.47	0.32
PPol	0.23 / –	0.34	0.35 / –	0.13	0.45 / –	0.38
RealUserSim	0.29 / 0.27	0.34	0.32 / 0.32	**0.22**	0.43 / 0.35	0.44
CUE	**0.13** / **0.13**	**0.38**	0.34 / 0.33	0.20	0.35 / 0.32	0.46
GPT 5.4 Mini						
None	**0.08** / **0.08**	**0.36**	**0.24** / 0.24	0.16	**0.09** / 0.09	0.56
PPol	0.32 / –	0.29	0.33 / –	0.17	0.22 / –	0.51
RealUserSim	0.44 / 0.38	0.31	0.33 / 0.27	0.22	0.11 / **0.05**	0.56
CUE	0.10 / 0.10	0.31	0.25 / **0.18**	**0.24**	**0.09** / 0.10	**0.57**
Gemini 3.5 Flash Lite						
None	0.12 / 0.12	**0.33**	**0.35** / 0.35	0.14	0.21 / 0.21	0.56
PPol	0.67 / –	0.20	0.58 / –	0.09	0.34 / –	0.42
RealUserSim	0.62 / 0.63	0.21	0.59 / 0.54	0.06	0.32 / 0.31	0.46
CUE	**0.06** / **0.06**	**0.33**	**0.35** / **0.28**	**0.22**	**0.12** / **0.12**	**0.62**

Table 3: **Outcome calibration on τ^2-Bench.** Paired entries are user-conditioned/sampled; F1 metrics are only user-conditioned. Attribution F1 conditions on both episodes failing, mode F1 conditions on both having agent-attributed failures. Bold marks best value per variation.

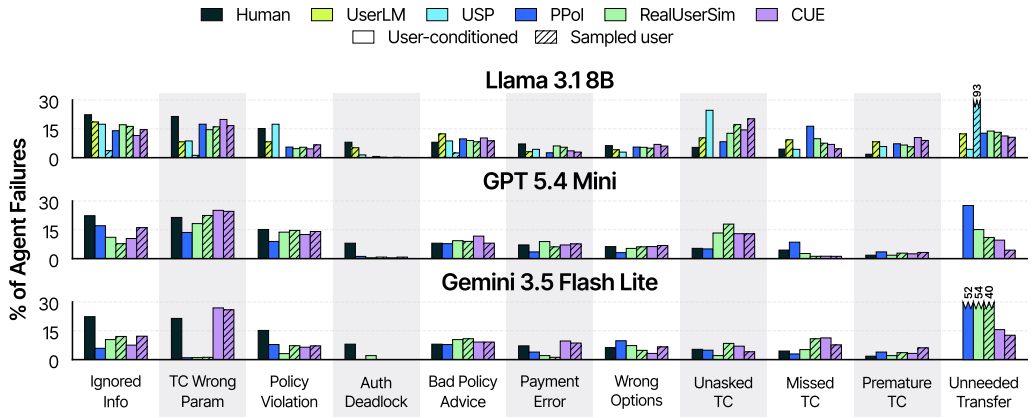


Figure 2: **Conditional failure composition over agent-attributed failures.** Bars show the proportional distribution of failure modes within each method's agent-attributed failures. *TC = Tool Call*.

(Appendix F). The taxonomy was refined over three researcher-reviewed batches of 100 episodes; the final batch required only one correction.

Role Adherence CUE has the smallest user-share gap among persona-based methods on every base model. Compared with None, CUE has smaller gaps on Llama (0.13 versus 0.14) and Gemini (0.06 versus 0.12), but a larger gap on GPT (0.10 versus 0.08). These are shares conditional on failure: a change can reflect user-side failures or changes in other failure causes. Paired attribution macro-F1 adds a case-level check on episodes where both human and simulator fail. CUE leads on Llama (0.38 vs 0.35 for None), ties on Gemini (0.33), and falls below None on GPT (0.31 vs 0.36). Broadly, other simulation methods result in poorer alignment of failure attribution despite their fidelity performance, while CUE is in line with the standard simulator used in the benchmark.

Base	Method	MAE ↓ Document Quality	MAE ↓ Interaction Quality
Llama 3.1 8B	USP	5.34	1.86
	RealUserSim	6.06	2.21
	CUE	**3.55**	**1.48**
GPT 5.4 Mini	RealUserSim	5.80	1.39
	CUE	**2.21**	**1.33**
Gemini 3.5 Flash Lite	RealUserSim	4.93	1.42
	CUE	**2.42**	**1.41**
Human		0.85-1.48	0.87-1.50

Table 4: **Outcome calibration over SimulatorArena writing benchmark.** Pairwise mean absolute error (MAE) over user-conditioned models using the benchmark's LLM-judge grader.

Failure Reproduction Figure 2 illustrates distortions in failure composition, including authentication deadlocks and unnecessary escalation. CUE has the lowest sampled agent-failure TVD on GPT (0.18) and Gemini (0.28), while UserLM leads on Llama (0.31), but with little differentiation across all methods except USP. User-conditioned failure mode F1 asks whether the specific failure mode agrees when both episodes have agent-attributed failures. We can see in the prompt example provided in Appendix H.2 that CUE prompts do not encode task-specific information that would necessarily result in the same failure mode such, however, CUE again has the highest user-task failure mode macro-F1 on GPT and Gemini and landing within the confidence interval of the highest value on Llama.

Success Alignment Against the human success rate of 0.61, None is closest on Llama, outperforming all methods. On the other hand, CUE and RealUserSim close this gap with GPT as the base model and CUE outperforms all other methods including None by a large margin on Gemini. When examining replay, CUE has the highest outcome macro-F1 among persona methods on all three base models. Relative to None, however, CUE again displays mixed results with better performance on the two stronger simulator base models.

Fidelity and Outcome Agreement Fidelity does not select a uniformly best outcome simulator – different metrics rank methods differently. Although the no-persona baseline often scores below CUE on fidelity, it remains competitive on outcome agreement and outperforms several persona-based methods. In particular, persona conditioning can increase behavioral resemblance while simultaneously shifting task difficulty or cooperation in ways that worsen agreement with human outcomes. We therefore recommend evaluating outcome calibration directly alongside fidelity when possible rather than assuming that high naturalness, mimicry, or coverage begets alignment for benchmarking.

7 Transfer to Held-Out Benchmarks

We apply the same CUE model to SimulatorArena, an open-ended cooperative benchmark covering co-writing and tutoring, and PRISM, a demographically diverse casual conversation benchmark, without retraining (Appendix G). These are held-out benchmarks with domains outside of task-oriented dialogue. While CUE's training mixture contains some general conversation from LMSYS and WildChat (∼20k samples), the majority is from task-oriented, so this evaluates whether the system transfers despite the training data imbalance.

Writing-Outcome Agreement Table 4 reports paired human-simulator mean absolute error on document and interaction quality scores as evaluated by the SimulatorArena benchmark's LLM judge. Interaction quality is more akin to user fidelity (examining the surface interaction) while document quality is more akin to outcome calibration (examining the outcome of the interaction). The scoring is on a scale of 1-10; for score s, MAE $= \frac{1}{N}\sum_i |s_i^{\text{sim}} - s_i^{\text{human}}|$. The human range provided for context is calculated by performing the same calculation, using two humans who completed the same task as the reference and prediction respectively. CUE reduces document-quality error for every base model, including 2.21 versus RealUserSim's 5.80 on GPT 5.4 Mini. Interaction

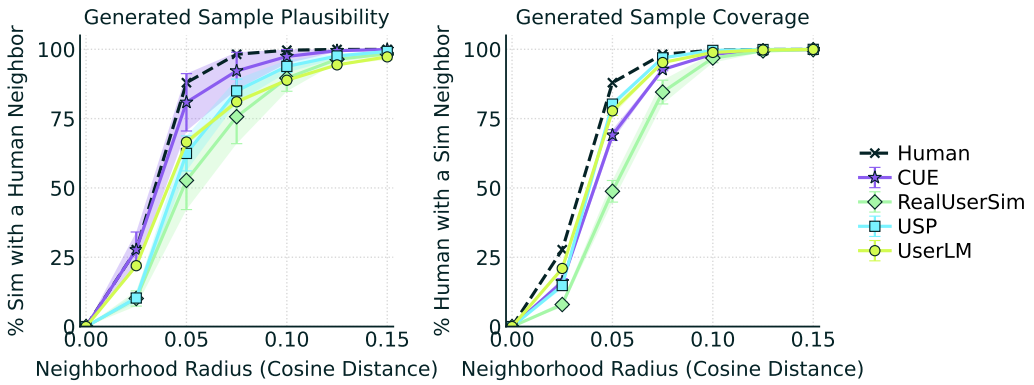


Figure 3: **Bidirectional stylistic proximity on PRISM.** Left: Percent of generated trajectories with a human neighbor. Right: Percent of human trajectories with a generated neighbor. Methods use the same set for all measurements. Displayed intervals aggregate seeds and backbones into a 95% CI.

quality differences are smaller (i.e. 1.33 versus 1.39 here) with both CUE and RealUserSim falling within the human range on most base models. CUE's document error nevertheless remains above the human comparison range, indicating that two humans performing the same task differ less in document quality than each simulator and its paired human.

Bidirectional Stylistic Coverage of Diverse Populations The behavioral coverage scores saturate on PRISM, where all reported methods meet or exceed 0.86. To examine how well each method covers the user population of PRISM, Figure 3 plots neighborhood overlap in the StyleDistance space (Patel et al., 2025) using the same 5,284 conversations from both the human study and each method. The leave-one-out human reference curve demonstrates the density of the human samples, with most samples being within 0.10 of another. The left panel measures generated-to-human proximity: at a radius 0.05, 81% of CUE trajectories have a human neighbor compared to 67% of the nearest alternative. The right panel measures human-to-generated coverage: at the same radius, USP and UserLM have 80% and 78% coverage respectively while CUE has 69%. Notably, generating trajectories close to observed human behavior and covering the observed human population are disparate goals. At small neighborhood radii, CUE yields the highest generated-to-human proximity, whereas UserLM and USP attain greater human-to-generated coverage. This illustrates a tradeoff between generating samples near the observed human manifold and covering more of that manifold.

8 Conclusion

CUE provides a shared representation and prompt interface for retrospective session replay and sampled user simulation. Across τ^2-Bench, SimulatorArena, and PRISM, the complete pipeline is competitive on user fidelity and generally improves outcome agreement relative to other persona-based methods. However, the strong performance of the no-persona baseline on several τ^2-Bench settings shows that adding a more human-like persona does not necessarily produce a better benchmarking proxy. We therefore recommend evaluating outcome calibration directly rather than treating naturalness, mimicry, or coverage as sufficient evidence of benchmarking validity.

Our results do not establish that the continuous representation or diffusion sampler individually causes these gains, that sampled sessions match the distribution of every target population, or that retrospective replay predicts future user behavior. Matched-supervision controls and human studies spanning multiple agents are needed to test these mechanisms and determine whether outcome calibration preserves comparative agent rankings.

Ethics Statement

This work studies user simulation using representations learned from real human interaction trajectories. While our intended application is the evaluation of interactive AI systems, modeling human behavior inherently requires consideration regarding privacy, representation, and the potential for misuse.

Data Provenance and Privacy Our models are trained and evaluated using previously released dialogue and interaction datasets under their respective terms and licenses. Some of these datasets contain conversations produced by real users. While we are embedding the behavior of the user in these sessions, it is important not to consider the resulting CUEs as representations of the users' identities or personal characteristics. When embedding, we perturb sessions to remove proper nouns for both privacy and robustness purposes, but while these procedures reduce direct content leakage they do not guarantee that the model does not encode any information correlated with sensitive or identifying attributes.

Representational Limitations The populations represented in our training dataset are not representative of the broader human population. They reflect the underlying population that were able to be reached and were willing to participate in these studies. Likewise, our population coverage experiments refer to coverage relative to the observed distributions in user studies, not people in general. While we attempt to combat this by evaluating on PRISM, an intentionally diverse dataset, biases or under-representation in the source data may be reproduced by the resulting simulators.

Misuse of Simulated Users CUE is intended as a tool for studying and evaluating interactive AI systems. Simulated users should not be treated as substitutes for real human participation, especially when evaluating high-stakes systems or gauging stakeholder preferences, experiences, or representation. In particular, we do not intend CUE to be used for profiling individuals, making decisions about users, or inferring sensitive attributes.

Our user-conditioned experiments are retrospective: representations are inferred from observed human interactions and used to replay the corresponding user-task cases. These experiments evaluate reproduction of the observed interaction outcomes and should not be interpreted as demonstrating prospective prediction of an individual's behavior on future or unseen tasks. Establishing such claims would require longitudinal or cross-session evaluation with appropriate consent and additional privacy safeguards.

More capable user simulators could potentially be used to imitate behavioral patterns of individuals or groups, optimize systems towards synthetic rather than real users, or create misleading claims about population behavior. We therefore encourage use of these methods for controlled evaluation and research, with real user studies, with intentional and careful participant selection, performed in cases where substantive conclusions are proposed.

Reproducibility Statement

We release the complete implementation code used in this work, including the code for data processing, training, sampling, simulation, and evaluation. We additionally release the trained CUE encoder, decoder, sampler, example pool, and LLM-annotated training data.

All training and evaluation data are drawn from previously released public datasets. Dataset sources and subsets are documented in the paper and repository. No proprietary data or code is required to reproduce the reported results. For experiments using externally hosted language models, we report the provider, model identifier, prompts, decoding settings, and evaluation date, but exact outputs may vary as hosted model implementations change. The repository additionally provides the environment and dependency specifications, commands for reproducing the experiments and tables, model and evaluation hyperparameters, and seed values. The released checkpoints allow the main evaluation to be reproduced without retraining, but the full training data is present for training the models from scratch.

9 AI USE STATEMENT

LLMs were used as general-purpose coding and writing assistants. Specifically, LLMs were used to implement research code, improve the clarity of manuscript text, and identify potential weakness in methodology. All scientific claims, experimental designs, results, and interpretations were verified by the authors. LLMs were not used as authors and were not relied on as a source of factual or bibliographic information. The authors take full responsibility for the content of the paper.

REFERENCES

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob L Menick, Sebastian Borgeaud, Andy Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikoł aj Bińkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karén Simonyan. Flamingo: A Visual Language Model for Few-Shot Learning. In *Advances in Neural Information Processing Systems*, 2022.

Lisa P Argyle, Ethan C Busby, Nancy Fulda, Joshua R Gubler, Christopher Rytting, and David Wingate. Out of One, Many: Using Language Models to Simulate Human Samples. *Political Analysis*, 31(3):337–351, 2023.

Ge Bai, Jie Liu, Xingyuan Bu, Yancheng He, Jiaheng Liu, Zhanhui Zhou, Zhuoran Lin, Wenbo Su, Tiezheng Ge, Bo Zheng, and Wanli Ouyang. MT-Bench-101: A Fine-Grained Benchmark for Evaluating Large Language Models in Multi-Turn Dialogues. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, 2024.

Adrien Bardes, Jean Ponce, and Yann LeCun. VICReg: Variance-Invariance-Covariance regularization for self-supervised learning. *International Conference on Learning Representations*, 2022.

Victor Barres, Honghua Dong, Soham Ray, Xujie Si, and Karthik R Narasimhan. τ^2-Bench: Evaluating Conversational Agents in a Dual-Control Environment. In *International Conference on Learning Representations*, 2026.

Bill Byrne, Karthik Krishnamoorthi, Chinnadhurai Sankar, Arvind Neelakantan, Ben Goodrich, Daniel Duckworth, Semih Yavuz, Amit Dubey, Kyuyoung Kim, and Andy Cedilnik. Taskmaster-1: Toward a Realistic and Diverse Dialog Dataset. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2019.

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-End Object Detection with Transformers. In *European Conference on Computer Vision*. Springer, 2020.

Serina Chang, Ashton Anderson, and Jake M. Hofman. ChatBench: From Static Benchmarks to Human-AI Evaluation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics*, 2025.

Guan-Lin Chao and Ian Lane. BERT-DST: Scalable End-to-End Dialogue State Tracking with Bidirectional Encoder Representations from Transformer. In *Interspeech*, 2019.

Kushal Chawla, Jaysa Ramirez, Rene Clever, Gale Lucas, Jonathan May, and Jonathan Gratch. CaSiNo: A corpus of campsite negotiation dialogues for automatic negotiation systems. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2021.

Derek Chen, Howard Chen, Yi Yang, Alexander Lin, and Zhou Yu. Action-Based Conversations Dataset: A Corpus for Building More In-Depth Task-Oriented Dialogue Systems. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2021.

Wenhu Chen, Jianshu Chen, Pengda Qin, Xifeng Yan, and William Yang Wang. Semantically Conditioned Dialog Response Generation via Hierarchical Disentangled Self-Attention. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019.

Zhiyu Chen, Bing Liu, Seungwhan Moon, Chinnadhurai Sankar, Paul Crook, and William Yang Wang. KETOD: Knowledge-Enriched Task-Oriented Dialogue. In Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz (eds.), *Findings of the Association for Computational Linguistics: NAACL*, 2022.

Harshita Chopra, Kshitish Ghate, Aylin Caliskan, Tadayoshi Kohno, Chirag Shah, and Natasha Jaques. Beyond Cooperative Simulators: Generating Realistic User Personas for Robust Evaluation of LLM Agents. In *Conference on Language Modeling: Workshop on Social Simulation*, 2026.

Yao Dou, Michel Galley, Baolin Peng, Chris Kedzie, Weixin Cai, Alan Ritter, Chris Quirk, Wei Xu, and Jianfeng Gao. SimulatorArena: Are User Simulators Reliable Proxies for Multi-Turn Evaluation of AI Assistants? In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2025.

Layla El Asri, Hannes Schulz, Shikhar Kr Sarma, Jeremie Zumer, Justin Harris, Emery Fine, Rahul Mehrotra, and Kaheer Suleman. FRAMES: A Corpus for Adding Memory to Goal-Oriented Dialogue Systems. In *Proceedings of the 18th annual SIGdial meeting on discourse and dialogue*, 2017.

Mihail Eric, Lakshmi Krishnan, Francois Charette, and Christopher D. Manning. Key-Value Retrieval Networks for Task-Oriented Dialogue. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, 2017.

Mihail Eric, Rahul Goel, Shachi Paul, Abhishek Sethi, Sanchit Agarwal, Shuyang Gao, Adarsh Kumar, Anuj Goyal, Peter Ku, and Dilek Hakkani-Tur. MultiWOZ 2.1: A Consolidated Multi-Domain Dialogue Dataset with State Corrections and State Tracking Baselines. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, 2020.

He He, Derek Chen, Anusha Balakrishnan, and Percy Liang. Decoupling Strategy and Generation in Negotiation Dialogues. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2018.

Jonathan Ho and Tim Salimans. Classifier-Free Diffusion Guidance. In *Advances in Neural Information Processing Systems: Workshop on Deep Generative Models and Applications*, 2021.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising Diffusion Probabilistic Models. *Advances in Neural Information Processing Systems*, 2020.

Tiancheng Hu, Joachim Baumann, Lorenzo Lupo, Nigel Collier, Dirk Hovy, and Paul Röttger. Simbench: Benchmarking the ability of large language models to simulate human behaviors. In *First Workshop on Bridging NLP and Public Opinion Research*, 2025.

Anjali Kantharuban, Aarohi Srivastava, Fahim Faisal, Orevaoghene Ahia, Antonios Anastasopoulos, David Chiang, Yulia Tsvetkov, and Graham Neubig. IdioleX: Unified and Continuous Representations for Idiolectal and Stylistic Variation. *arXiv preprint arXiv:2604.04704*, 2026.

Hannah R Kirk, Alexander Whitefield, Paul Röttger, Andrew Bean, Katerina Margatina, Juan Ciro, Rafael Mosquera, Max Bartolo, Adina Williams, He He, et al. The PRISM alignment dataset: What Participatory, Representative and Individualised Human Feedback Reveals about the Subjective and Multicultural Alignment of Large Language Models. *Advances in Neural Information Processing Systems*, 2024.

Xiujun Li, Sarah Panda, Jingjing Liu, and Jianfeng Gao. Microsoft Dialogue Challenge: Building End-to-End Task-Completion Dialogue Systems. *arXiv preprint arXiv:1807.11125*, 2018.

Yubo Li, Xiaobin Shen, Yidi Miao, Xinyu Yao, Xueying Ding, Ramayya Krishnan, and Rema Padman. Beyond Single-Turn: A Survey on Multi-Turn Interactions with Large Language Models. *Transactions on Machine Learning Research*, 2026.

Zhaojiang Lin, Andrea Madotto, Genta Indra Winata, Peng Xu, Feijun Jiang, Yuxiang Hu, Chen Shi, and Pascale Fung. BiToD: A Bilingual Multi-Domain Dataset For Task-Oriented Dialogue Modeling. In *Advances in Neural Information Processing Systems*, 2021.

Langming Liu, Shilei Liu, Yujin Yuan, Yizhen Zhang, Bencheng Yan, Zhiyuan Zeng, Zihao Wang, Jiaqi Liu, Di Wang, Wenbo Su, et al. UQABench: Evaluating User Embedding for Prompting LLMs in Personalized Question Answering. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2025.

Sichun Luo, Yuanzhang Xiao, and Linqi Song. Personalized federated recommendation via joint representation learning, user clustering, and model adaptation. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, 2022.

Xiang Luo, Zhiwen Tang, Jin Wang, and Xuejie Zhang. DuetSim: Building User Simulator with Dual Large Language Models for Task-Oriented Dialogues. In *Proceedings of the Joint International Conference on Computational Linguistics, Language Resources and Evaluation*, 2024.

Scott Martin, Shivani Poddar, and Kartikeya Upasani. MuDoCo: Corpus for Multidomain Coreference Resolution and Referring Expression Generation. In Nicoletta Calzolari, Frédéric Béchet, Philippe Blache, Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asuncion Moreno, Jan Odijk, and Stelios Piperidis (eds.), *Proceedings of the Twelfth Language Resources and Evaluation Conference*, 2020.

Johannes EM Mosig, Shikib Mehri, and Thomas Kober. Star: A schema-guided dialog dataset for transfer learning. *arXiv preprint arXiv:2010.11853*, 2020.

Nikola Mrkšić, Diarmuid Ó Séaghdha, Tsung-Hsien Wen, Blaise Thomson, and Steve Young. Neural Belief Tracker: Data-Driven Dialogue State Tracking. In Regina Barzilay and Min-Yen Kan (eds.), *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, 2017.

Tarek Naous, Philippe Laban, Wei Xu, and Jennifer Neville. Flipping the Dialogue: Training and Evaluating User Language Models. In *International Conference on Learning Representations*, 2026.

Alexander Quinn Nichol and Prafulla Dhariwal. Improved Denoising Diffusion Probabilistic Models. In *Proceedings of Machine Learning Research*, 2021.

Lin Ning, Luyang Liu, Jiaxing Wu, Neo Wu, Devora Berlowitz, Sushant Prakash, Bradley Green, Shawn O'Banion, and Jun Xie. User-LLM: Efficient LLM Contextualization with User Embeddings. In *Companion Proceedings of the ACM on Web Conference*, 2025.

Zach Nussbaum, John X. Morris, Brandon Duderstadt, and Andriy Mulyar. Nomic Embed: Training a Reproducible Long Context Text Embedder. *Transactions on Machine Learning Research*, 2025.

Jane Pan, Ryan Shar, Jacob Pfau, Ameet Talwalkar, He He, and Valerie Chen. When Benchmarks Talk: Re-Evaluating Code LLMs with Interactive Feedback. In *Findings of the Association for Computational Linguistics*, 2025.

Ajay Patel, Jiacheng Zhu, Justin Qiu, Zachary Horvitz, Marianna Apidianaki, Kathleen McKeown, and Chris Callison-Burch. StyleDistance: Stronger Content-Independent Style Embeddings with Synthetic Parallel Examples. In Luis Chiruzzo, Alan Ritter, and Lu Wang (eds.), *Proceedings of the Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2025.

Denis Peskov, Nancy Clarke, Jason Krone, Brigi Fodor, Yi Zhang, Adel Youssef, and Mona Diab. "Multi-Domain Goal-Oriented Dialogues (MultiDoGO): Strategies toward Curating and Annotating Large Scale Dialogue Data. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing)*, 2019.

Cheng Qian, Zuxin Liu, Akshara Prabhakar, Zhiwei Liu, Jianguo Zhang, Haolin Chen, Heng Ji, Weiran Yao, Shelby Heinecke, Silvio Savarese, and Huan Wang. UserBench: An Interactive Gym Environment for User-Centric Agents. In *Proceedings of the Workshop on Scaling Environments for Agents*, 2025.

Kun Qian, Satwik Kottur, Ahmad Beirami, Shahin Shayandeh, Paul Crook, Alborz Geramifard, Zhou Yu, and Chinnadhurai Sankar. Database Search Results Disambiguation for Task-Oriented Dialog Systems. In Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz (eds.), *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2022.

Jun Quan, Deyi Xiong, Bonnie Webber, and Changjian Hu. GeCoR: An End-to-End Generative Ellipsis and Co-Reference Resolution Model for Task-Oriented Dialogue. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2019.

Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan. Towards Scalable Multi-Domain Conversational Agents: The Schema-Guided Dialogue Dataset. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020.

Nils Reimers and Iryna Gurevych. Sentence-Bert: Sentence Embeddings using Siamese Bert-Networks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2019.

Rafael A Rivera-Soto, Olivia Elizabeth Miano, Juanita Ordonez, Barry Y Chen, Aleem Khan, Marcus Bishop, and Nicholas Andrews. Learning Universal Authorship Representations. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2021.

Preethi Seshadri, Samuel Cahyawijaya, Ayomide Odumakinde, Sameer Singh, and Seraphina Goldfarb-Tarrant. Lost in simulation: Llm-simulated users are unreliable proxies for human users in agentic evaluations. In *"Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)"*, 2026.

Igor Shalyminov, Alessandro Sordoni, Adam Atkinson, and Hannes Schulz. Fast Domain Adaptation For Goal-Oriented Dialogue Using A Hybrid Generative-Retrieval Transformer. In *2020 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2020.

Joseph Suh, Ayush Raj, Minwoo Kang, and Serina Chang. Quantifying the Utility of User Simulators for Building Collaborative LLM Assistants. *arXiv preprint arXiv:2605.09808*, 2026.

Harsh Trivedi, Tushar Khot, Mareike Hartmann, Ruskin Manku, Vinty Dong, Edward Li, Shashank Gupta, Ashish Sabharwal, and Niranjan Balasubramanian. AppWorld: A Controllable World of Apps and People for Benchmarking Interactive Coding Agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, 2024.

Sanidhya Vijayvargiya, Xuhui Zhou, Akhila Yerukola, Maarten Sap, and Graham Neubig. AmbigSWE: Interactive Agents to Overcome Underspecificity in Software Engineering. In *International Conference on Learning Representations*, 2026.

Kuang Wang, Xianfei Li, Shenghao Yang, Li Zhou, Feng Jiang, and Haizhou Li. Know You First and Be You Better: Modeling Human-Like User Simulators via Implicit Profiles. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics*, 2025.

Yingshan Susan Wang, Cedegao E Zhang, Linlu Qiu, Zexue He, Pengyuan Li, Alex Pentland, Roger P Levy, and Yoon Kim. Learning User Simulators with Turing Rewards. In *Second Workshop on Social Simulation with LLMS: Fidelity in Applications*, 2026.

Anna Wegmann, Marijn Schraagen, and Dong Nguyen. Same Author or Just Same Topic? Towards Content-Independent Style Representations. In *Proceedings of the 7th Workshop on Representation Learning for NLP*, 2022.

Wei Wei, Quoc Le, Andrew Dai, and Jia Li. AirDialogue: An environment for goal-oriented dialogue research. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii (eds.), *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2018.

Xing Wu, Chaochen Gao, Zijia Lin, Jizhong Han, Zhongyuan Wang, and Songlin Hu. InfoCSE: Information-Aggregated Contrastive Learning of Sentence Embeddings. In *Findings of the Conference on Empirical Methods in Natural Language Processing*, 2022.

Frank Fangzheng Xu, Yufan Song, Boxuan Li, Yuxuan Tang, Kritanjali Jain, Mengxue Bao, Zora Wang, Xuhui Zhou, Zhitong Guo, Murong Cao, et al. TheAgentCompany: Benchmarking LLM Agents on Consequential Real World Tasks. *Advances in Neural Information Processing Systems*, 38, 2026.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 Technical Report. *arXiv preprint arXiv:2505.09388*, 2025.

John Yang, Akshara Prabhakar, Karthik Narasimhan, and Shunyu Yao. Intercode: Standardizing and Benchmarking Interactive Coding with Execution Feedback. *Advances in Neural Information Processing Systems*, 2023.

Xiaoxue Zang, Abhinav Rastogi, Srinivas Sunkara, Raghav Gupta, Jianguo Zhang, and Jindong Chen. MultiWOZ 2.2: A Dialogue Dataset with Additional Annotation Corrections and State Tracking Baselines. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI, ACL 2020*, 2020.

Jianguo Zhang, Kun Qian, Zhiwei Liu, Shelby Heinecke, Rui Meng, Ye Liu, Zhou Yu, Huan Wang, Silvio Savarese, and Caiming Xiong. DialogStudio: Towards Richest and Most Diverse Unified Dataset Collection for Conversational AI. In *Findings of the Association for Computational Linguistics: EACL*, 2024a.

Wei Zhang, Dai Li, Chen Liang, Fang Zhou, Zhongke Zhang, Xuewei Wang, Ru Li, Yi Zhou, Yaning Huang, Dong Liang, Kai Wang, Zhangyuan Wang, Zhengxing Chen, Fenggang Wu, Minghai Chen, Huayu Li, Yunnan Wu, Zhan Shu, Mindi Yuan, and Sri Reddy. Scaling User Modeling: Large-Scale Online User Representations for Ads Personalization in Meta. In *Companion Proceedings of the ACM Web Conference*, 2024b.

Wenting Zhao, Xiang Ren, Jack Hessel, Claire Cardie, Yejin Choi, and Yuntian Deng. WildChat: 1M ChatGPT Interaction Logs in the Wild. In *International Conference on Learning Representations*, 2024.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Tianle Li, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zhuohan Li, Zi Lin, Eric Xing, et al. LMSYS-Chat-1M: A Large-Scale Real-World LLM Conversation Dataset. In *International Conference on Learning Representations*, 2024.

Shuyan Zhou, Frank F Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, et al. WebArena: A Realistic Web Environment for Building Autonomous Agents. In *International Conference on Learning Representations*, volume 2024, 2024.

Xuhui Zhou, Weiwei Sun, Qianou Ma, Yiqing Xie, Jiarui Liu, Weihua Du, Sean Welleck, Yiming Yang, Graham Neubig, Tongshuang Sherry Wu, and Maarten Sap. Mind the Sim2Real Gap in User Simulation for Agentic Tasks. In *Third Conference on Language Modeling*, 2026.

Ming Zhu, Juntao Tan, Rithesh Murthy, Jielin Qiu, Liangwei Yang, Wenting Zhao, Silvio Savarese, Shelby Heinecke, and Huan Wang. RealUserSim: Bridging the Reality Gap in Agent Benchmarking via Grounded User Simulation. In *Third Conference on Language Modeling*, 2026.

A Data

A.1 CUE Training Data

The CUE encoder uses the annotated portions of the source datasets in Table 5. These sources mix naturalistic human-LLM conversations, elicited or role-play dialogue, human-human negotiations, and human-authored or paraphrased task-oriented data. They should not all be described as interactions between real users and deployed agents. The first two are large, general-purpose dialogue corpora, while the remaining datasets are task-oriented. Only trajectories for which we construct the supervision described in Section A.2 are used to train CUE; additional unlabeled trajectories are retained as part of the data inventory but are not used for training.

Dataset	Train	Dev
LMSYS Chat 1M (Zheng et al., 2024)	10,000	
WildChat (Zhao et al., 2024)	10,000	
ABCD (Chen et al., 2021)	8,034	1,004
AirDialogue (Wei et al., 2018)	10,000	10,000
BiTOD (Lin et al., 2021)	2,952	295
CaSiNo (Chawla et al., 2021)	900	30
CraigslistBargains (He et al., 2018)	4,000	570
DSTC2 Clean (Chao & Lane, 2019)	1,612	506
Disambiguation (Qian et al., 2022)	8,433	999
FRAMES (El Asri et al., 2017)	1,329	
GECOR (Quan et al., 2019)	676	
HDSA-Dialog (Chen et al., 2019)	8,438	1,000
KETOD (Chen et al., 2022)	4,247	545
KVRET (Eric et al., 2017)	2,424	302
MetaLWOZ (Shalyminov et al., 2020)	10,000	
MS-DC (Li et al., 2018)	9,985	
MuDoCo (Martin et al., 2020)	6,032	690
MultiDoGO (Peskov et al., 2019)	10,000	1,150
MultiWOZ 2.1 (Eric et al., 2020)	8,434	999
MultiWOZ 2.2 (Zang et al., 2020)	8,437	1,000
SGD (Rastogi et al., 2020)	10,000	2,482
STAR (Mosig et al., 2020)	6,145	
Taskmaster 1 (Byrne et al., 2019)	6,170	769
Taskmaster 2 (Byrne et al., 2019)	10,000	
Taskmaster 3(Byrne et al., 2019)	10,000	10,000
WOZ 2.0 (Mrkšić et al., 2017)	600	200
Total	**168,848**	**32,541**

Table 5: **CUE source inventory.** Reported trajectories per source split; related dataset versions are not necessarily unique conversations, and inventory counts are not annotation-retention counts.

A.2 Proposal Prompts & Gating for Training

For each conversation in the train split, we construct a structured persona manual describing user-specific behavioral and stylistic characteristics. The labeling procedure uses two complementary contrast sets to distinguish both simulator-like behavior and variation among real users.

At selected user turns, four LLM simulators (Llama 3.1 8B Instruct, Qwen 3 8B, GPT-5.4 Mini, and Claude Haiku 4.5) generate counterfactual responses to the same dialogue context. These responses provide *simulator contrasts*: examples of how generic simulators differ from the target user. We additionally retrieve five real-user conversations as *human contrasts*, which help identify behaviors that distinguish the target user from other humans rather than merely from LLM-generated dialogue.

Given the target conversation and these two contrast sets, a proposal model (GPT-5.4 Mini) generates three groups of candidate commands: five general commands derived from simulator contrasts, five user-specific commands derived from human contrasts, and five style commands. Together, these commands are intended to capture both human-like interaction behavior and variation in how individual users communicate. Each command is accompanied by evidence turn IDs identifying where the corresponding behavior is expressed in the target conversation.

The proposal prompt explicitly excludes task-specific content, such as identifiers, products, account details, or domain policies, so that commands describe transferable user behavior rather than the particular task being completed. When behavior depends on dialogue history – for example, delayed answers, pushback, or escalation – the prompt requires reactive commands expressed as conditional "if...then..." rules rather than unconditional persona attributes.

Candidate manuals are schema-validated, delexicalized, and filtered using deterministic gates:

- **Command Coverage.** The manual must contain at least 8 commands, including at least 4 simulator-contrast commands and 4 human-contrast commands.

- **Evidence Grounding.** Every command must retain at least one valid evidence turn ID after grounding against the target user's turns.
- **Content Leakage.** Commands and examples are rejected if they contain task-specific content detected by a fixed lexical filter, including terms associated with credentials, identifiers, products, policies, addresses, receipts, or deliveries.
- **Generic Human Contrast.** A manual is rejected if at least 3 human-contrast commands reduce to generic behavioral descriptions such as being polite, cooperative, calm, matter-of-fact, concise, or task-focused.
- **Distinctive Human Contrast.** The manual must contain at least 2 human-contrast commands describing behavioral signatures that are present in the target and uncommon among the human negatives. These include behaviors such as delayed answering, escalation, push-back, impatience, try-then-report behavior, small talk, playful testing, or acceptance following refusal. Generic closers or behaviors shared by most human users do not satisfy this criterion.
- **Emotion Consistency.** Commands that attribute frustration, anger, or disappointment are rejected when the transcript contains only acceptance markers (e.g., "ok", "thanks", or "that's fine") and no corresponding negative affect.
- **Answer-Lag Contradiction.** We detect answer lag when a user turn responds to an earlier assistant request rather than the immediately preceding one. If lag is observed, commands claiming that the user consistently answers the latest prompt in sequence are rejected.
- **Answer-Lag Coverage.** If at least 2 answer-lag events are detected, the manual must include a command describing delayed, out-of-order, or previous-question answering.

Manuals that fail any gate are regenerated with feedback indicating the failed criteria, for up to three proposal attempts. The exact lexical filters and gate implementations are provided in the released code.

At rollout time, the resulting persona manual is inserted into the standard simulator prompt for the corresponding domain. The resulting labeled training set contains approximately 170k examples and constitutes the supervision used to train the CUE models.These proposal manuals provide training supervision. At evaluation, CUE uses commands generated by its learned decoder and separately retrieved training examples, not proposal manuals generated from evaluation trajectories. The counts in Table 5 describe the source inventory and should be distinguished from the number of accepted annotations and deduplicated sampler-bank entries.

A.3 Evaluation

Our primary evaluation domain is task-oriented dialogue, but we additionally evaluate on open-ended collaboration and general conversational assistance to test whether simulator quality generalizes across distinct forms of human-AI interaction.

For task-oriented dialogue, we use τ^2-Bench, a customer-service benchmark in which conversational agents interact with users while accessing tools and receiving programmatic rewards (Barres et al., 2026). We compare simulated users against the human interactions collected in the τ-USI study, which covers 50 airline tasks and 115 retail tasks with three human users per task (Zhou et al., 2026). Because several tasks used in τ-USI were subsequently removed from τ^2-Bench, our final evaluation set contains 483 human episodes.

We additionally evaluate on SimulatorArena, an open-ended human–AI collaboration benchmark containing 450 mathematics and 495 writing interactions (Dou et al., 2025), and PRISM, a large-scale conversational assistant study containing 5,284 multi-turn conversations from a diverse user population (Kirk et al., 2024).

Across all domains, we match the evaluated agent to the original human interaction as closely as possible. For τ^2-Bench, we use GPT 5.2 with high reasoning for all episodes. For SimulatorArena and PRISM, we recover the original agent from per-trajectory metadata when available. If the original model is unavailable, we substitute the closest available model by size within the same model family.

Population and split interpretation The representation unit is a conversation session, not a uniquely identified person. Source caps, repeated speakers, related dataset versions, and annota-

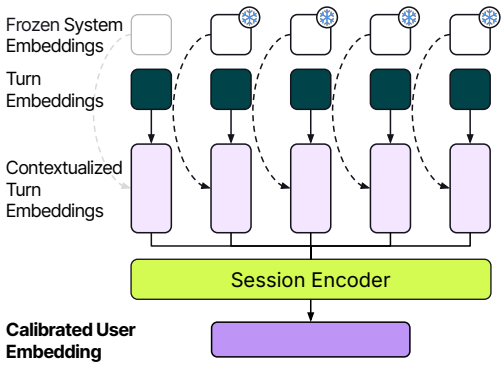


Figure 4: **Encoder architecture.** Hierarchical encoder first creates turn embeddings then aggregates them into session-level CUE.

tion gates affect the resulting mixture. In particular, MultiWOZ versions and derivatives may share underlying dialogues. However, because we use the source-provided train and validation splits, there is no overlap in train and validation trajectories.

B Architectural & Training Details

B.1 Joint Encoder–Decoder Training

Both the CUE turn-level encoder and the frozen semantic encoder used for de-correlation objectives are initialized from a pretrained ModernBERT embedding model[2], a ∼149M-parameter model with 768-dimensional token representations (Nussbaum et al., 2025). For each session, we encode up to 64 user turns of at most 256 tokens each. Token representations are aggregated using layer-wise attention pooling over transformer layers followed by mean pooling across tokens (Reimers & Gurevych, 2019; Kantharuban et al., 2026; Nussbaum et al., 2025). Corresponding system turns are encoded with a frozen copy of the backbone and attended to by the user representations to produce contextualized turn embeddings. A 4-layer session transformer then aggregates these turn representations into a 1024-dimensional CUE (Figure 4).

The prompt decoder is initialized from Qwen3-0.6B-Base, a ∼600M-parameter small language model with a 1024-dimensional hidden space (Figure 5). Persona-command targets are truncated to 256 tokens. Each CUE is projected into a fixed set of $M = 16$ soft-memory tokens, each matching the decoder hidden dimension. A 64-dimensional slot embedding is projected to decoder width and concatenated with the memory. One of 15 slot identities (five general, five session-specific, five style) supplies this token, yielding 17 conditioning tokens. This conditioning memory is injected through gated cross-attention every fourth transformer layer, following the general conditioning mechanism of Alayrac et al. (2022).

Training begins with a 200-step format warm-up in which the encoder is frozen and the decoder LM remains trainable. Because persona targets follow a constrained command format, this phase adapts the decoder to the output structure before representation-learning objectives are introduced. After warm-up, the decoder LM is frozen, the encoder and conditioning modules are trained, and the full objective below is enabled.

The primary generation objective is slot-conditioned cross-entropy. Rather than decoding the complete persona manual as one sequence, the decoder generates individual commands under discrete general, user-specific, and style slots. For each command type, let C_{ij} be the teacher-forced loss of target command j under slot i. We enumerate injective assignments π and minimize $\sum_j C_{\pi(j),j}$ using detached costs. Gradients then update the selected slot–command losses. Unused slots receive a no-op target at reduced weight. This avoids imposing an arbitrary target-command order; it is a design choice rather than an ablated necessity. We weight general, user-specific, and style command losses by $\lambda_g = 0.25$, $\lambda_u = 0.55$, and $\lambda_s = 0.40$, respectively. A head-separation objective addition-

[2] `nomic-ai/modernbert-embed-base`

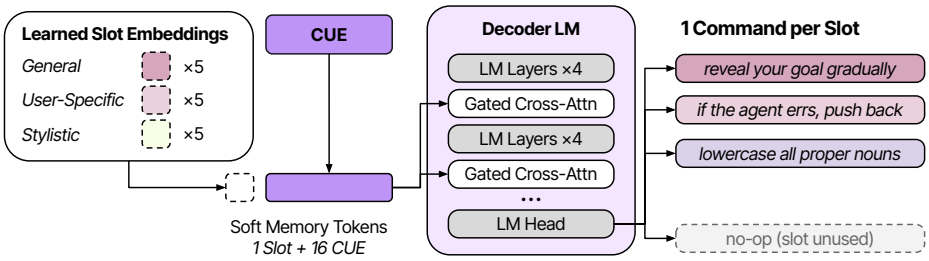


Figure 5: **Decoder architecture.** One command (or no-op) is generated for each slot.

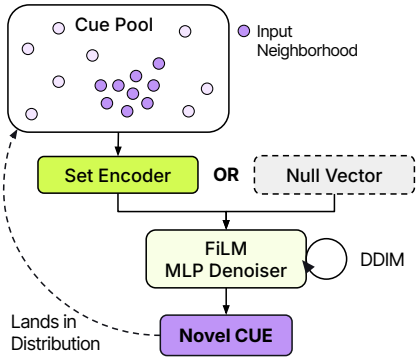


Figure 6: **Sampler Architecture.** Demonstrates two possible paths for generating a novel CUE.

ally requires commands assigned to one behavioral bank to score worse under the competing bank, discouraging duplication between general and user-specific slots.

We combine command supervision with representation-level objectives intended to preserve user-specific behavioral and stylistic information while reducing content leakage and representation collapse. First, we construct perturbed versions of each trajectory by replacing content-bearing spans such as URLs, numbers, code, and quoted strings, and maximize cosine similarity between the CUEs of the original and perturbed views. Second, we apply a session-level InfoNCE objective in which the perturbed view of the same trajectory is the positive and other trajectories in the batch serve as negatives (Wu et al., 2022). Third, we use a style-aware contrastive objective whose soft targets are derived from stylistic similarity scores from Wegmann et al. (2022), encouraging stylistically similar users to remain nearby even when their persona commands differ. Finally, VICReg variance and covariance penalties are applied to a 256-dimensional projection of the CUEs to discourage dimensional collapse and feature co-adaptation (Bardes et al., 2022).

Full hyperparameters are reported in Table 6. Training the joint encoder–decoder model takes approximately 14 hours on four A100 80GB GPUs using the approximately 170k labeled training examples described in Section A.

B.2 Sampler

The encoder-decoder components support *conditioned* simulation by deriving a CUE from an observed user trajectory and assigning that representation to a task. To additionally support sampling novel users, we train a latent diffusion model over CUE embeddings from the training corpus (Figure 6).

We first construct an embedding bank by encoding the training data, removing exact duplicates and near-duplicates with cosine similarity ≥ 0.995, and storing per-dimension statistics used to standardize the latent space. The embedding bank additionally retains the encoder LayerNorm affine parameters required for reconstruction. Path B in Figure 1 illustrates this sampling procedure.

Group	Hyperparameter	Value
Encoder	User / system backbone	`nomic-ai/modernbert-embed-base`
	Bottleneck dimension	1024
	Session dimension	1024
	Session transformer layers	4
	Session attention heads	8
	Max turns per session	64
	Max tokens per turn	256
Decoder	Base LM	`Qwen/Qwen3-0.6B-Base`
	Persona tokens	16
	Gated block insertion stride	every 4 layers
	Cross-attention heads	8
	Precision	bfloat16
Slot decoding	General command slots	5
	User-specific command slots	5
	Style command slots	5
	Slot embedding dimension	64
	No-op class weight	0.05
Optimization	Optimizer	AdamW
	LR (conditioning modules)	5×10^{-5}
	LR (encoder)	5×10^{-6}
	LR (decoder)	5×10^{-6}
	LR (gates)	5×10^{-4}
	LM format warm-up	200 steps
	Weight decay	0.01 (0.0 for gates)*
	LR schedule	constant
	Gradient clipping	1.0
	Epochs	3
Batching	Per-device batch size	4
	Gradient accumulation	8
	Data-parallel ranks	4 (DDP)
	Effective batch size	128 sessions
	Target max tokens	256
	Per-source cap	10,000
Loss weights	Consistency	0.1
	Variance	1.0
	Covariance	0.1
	CUE InfoNCE	0.1
	CUE style	1.0
	InfoNCE temperature	0.05
	Overlap target temperature	0.1
	Positive-turn dropout	0.3
	Head separation	0.1 (margin 0.2)
	General command CE	0.25
	User-specific command CE	0.55
	Style command CE	0.40

Table 6: **Joint encoder-decoder training hyperparameters.** During the 200-step LM-format warm-up, the encoder is frozen, the decoder LM is trainable, and representation-level conditioning losses are disabled. After warm-up, the decoder LM is frozen, the remaining objectives are enabled, and the optimizer is rebuilt. *Weight decay is the framework default and is not set explicitly in the configuration.

The diffusion denoiser is a residual MLP with 6 FiLM-modulated blocks and hidden width 1536, approximately 162M parameters in total. It predicts additive noise under a cosine diffusion schedule with 1000 timesteps (Nichol & Dhariwal, 2021). Conditioning is provided by a permutation-invariant set encoder consisting of four-headed attention pooling over a variable-size neighborhood of CUE embeddings, with a learned null vector representing the unconditional branch.

For each training instance, we sample a center from the embedding bank, retrieve its $k = 8$ nearest neighbors, select one neighbor at random as the diffusion target, and condition on a random non-empty subset of the remaining neighbors. With probability 0.15, conditioning is replaced by the null representation to enable classifier-free guidance (Ho & Salimans, 2021). The primary objective is noise-prediction MSE, supplemented by low-weight consistency, neighborhood-alignment, and batch-contrastive losses. These auxiliary terms act as guardrails that encourage generated samples to remain locally coherent and well-separated.

Sampler hyperparameters are reported in Table 7. Training the sampler takes approximately 3 hours on a single A100 80GB GPU.

B.3 Generation

Unless otherwise specified, all user simulators decode with temperature 0.7, top-p 0.95, and a maximum generation length of 256 tokens. These decoding settings are held fixed across simulator backends. The agent under evaluation is decoded greedily with temperature 0.0, top-p 1.0, and a 2048-token limit, reducing agent-side stochasticity across repeated simulator runs.

For CUE, persona manuals are decoded greedily using a single candidate per slot. Near-duplicate commands within a generated manual are removed when their lexical Jaccard similarity exceeds 0.8. Style-example retrieval selects 8 neighboring sessions by similarity and inserts 2 general and 6 user-specific examples into the simulator prompt.

We use 50 DDIM steps and the normalization described below. Unconditional sampling supplies no neighborhood and uses only the null-conditioned denoiser; the configured guidance weight is inactive. With a neighborhood, the noise estimate is $\epsilon_u + w(\epsilon_c - \epsilon_u)$, with rollout weight $w = 1.5$. All reported stochastic evaluations are run with seeds 0, 1, and 2.

B.4 Base Model Flexibility

Because CUEs decode into natural language persona prompts, the learned user representation is not tied to a particular base model. Across SimulatorArena and PRISM, CUE's naturalness greatly increases as the simulator base model strengthens while the CUE model remains fixed. As a result, CUE can adopt stronger or closed-source simulator models without retraining.

Inference uses the encoder, command decoder, and optionally the sampler, in addition to the simulator LLM. The listed components include two approximately 149M-parameter backbones, a 600M decoder, and a 162M sampler, along with the session transformer and conditioning modules. The following figures describe selected baseline costs, not an end-to-end cost comparison: PPol re-evolves its persona per base-domain combination (∼$150) and RealUserSim re-extracts persona manuals (∼$10) for each domain. Trained simulators are more expensive, needing 250 GPU-h, or substantially more for larger base models, for training. CUE also incurs one-time annotation costs: counterfactual generations from four simulators and up to three proposal attempts per session. Those costs, storage, and retrieval are not included in these figures. Our results show that stronger prompted simulators can improve selected metrics in contrast to patterns seen in prior work (Naous et al., 2026).

Sampler output normalization The implementation's "manifold projection" re-applies the encoder bottleneck's LayerNorm geometry after undoing training-bank standardization. For a generated vector z and stored LayerNorm parameters γ, β, define elementwise

$$u = (z - \beta)/\max(\gamma, \epsilon), \qquad \widetilde{z} = \gamma \odot \frac{u - \overline{u}}{\sqrt{\mathrm{Var}(u) + \epsilon}} + \beta, \qquad \epsilon = 10^{-5}.$$

Means and variances are over coordinates of each sample. Without stored affine parameters, the implementation centers and scales z directly.

Group	Hyperparameter	Value
Architecture	Latent dimension	1024
	MLP hidden width	1536
	Depth (residual blocks)	6
	Timestep embedding dim	256
	Conditioner heads	4
	Parameters	$\sim$162M
Diffusion	Noise schedule	cosine ($s = 0.008$)
	Timesteps	1000
	Conditioning set size k	8
	Unconditional dropout	0.15
Optimization	Optimizer	AdamW
	Learning rate	2×10^{-4}
	Weight decay	0.01
	Batch size	256
	Max steps	7,500
	LR schedule	linear warm-up 500 steps, then linear decay to $0.1\times$
	Gradient clipping	1.0
	Mixed precision / EMA	enabled / 0.999
Auxiliary losses	Consistency	0.1
	Neighborhood	0.05
	Contrastive	0.05
	Contrastive temperature	0.5
Sampling	DDIM steps	50
	Guidance scale w	2.5 (training evaluation); 1.5 at rollout*
	Manifold projection	enabled
Bank construction	Encode batch size	16
	Prior holdout fraction	0.02
	Near-duplicate cosine	0.995

Table 7: **Diffusion sampler hyperparameters.** The sampler is trained on a single GPU. *Rollout-time guidance is configured independently from the value used during training-time evaluation.

C BASELINES

We compare the CUE framework against four existing user-simulation methods. All methods are evaluated under the same closed-loop protocol: identical agents, domains, episode construction, and termination criteria. Where necessary, we adapt baseline implementations to this shared evaluation infrastructure while preserving their core modeling and decoding procedures. Across methods, user generations are passed through a common wrapper that prepends the task description and removes simulation-specific markers introduced by model training or benchmark-specific formatting.

C.1 USERLM

UserLM-8B (Naous et al., 2026) is a post-trained version of Llama 3.1 8B Base that generates user turns conditioned on a task description, supplied in the system message, and the preceding assistant–user dialogue history. We use the released `microsoft/UserLM-8b` checkpoint with the decoding settings recommended by Naous et al. (2026): `temperature` $= 1.0$, `top_p` $= 0.8$, and `max_tokens` $= 256$. Because UserLM does not condition generation on an individual user or persona, it is evaluated only in the sampled setting.

C.2 USP

USP (Wang et al., 2025) is a multistage framework that (i) extracts implicit user profiles from dialogue trajectories using an LLM, (ii) performs profile-conditioned supervised fine-tuning of Llama 3.1 8B Base on user turns, and (iii) applies reinforcement learning with cycle consistency (RLCC), rewarding generations whose re-extracted profiles match the target profile. We use the released `wangkevin02/USP` checkpoint and the decoding settings recommended by Wang et al. (2025).

We evaluate USP in two settings. For *user-conditioned* evaluation, generation is conditioned on the profile extracted for the target user. For *sampled* evaluation, we implement USP's Diverse Profile Sampling procedure, which uses SimCSE representations, UMAP dimensionality reduction, Gaussian KDE, and nearest-neighbor recombination of objective and subjective profile components over the released training data.

C.3 PPOL

PPol (Chopra et al., 2026) learns a persona-generation program using OpenEvolve. Candidate programs are evaluated using a behavioral discriminator together with Chamfer coverage over stylometric fingerprints derived from the behavioral taxonomy of Zhou et al. (2026). At deployment, the learned program generates N personas for a task context c, which are then inserted into the simulator prompt.

We use the released `persona-policies` codebase and independently evolve a `best_program.py` for each simulator backbone. The simulator model used during evolution is the same model used for evaluation, keeping the learned persona policy aligned with the downstream simulator.

Our implementation differs from the original evaluation in two respects. First, we do not impose a separate train/test split over τ^2-Bench trajectories. This grants PPol access to the evaluation task distribution. S2R-N and S2R-C also use its behavioral feature taxonomy, so they are not independent validation of that optimization objective. PPol is optimized using rollouts from the evaluation task distribution and therefore its τ^2-Bench fidelity results should be interpreted as an in-domain comparison favorable to PPol. Second, because PPol's evolution procedure depends on generating τ^2-Bench rollouts, we train and evaluate PPol only in the τ^2-Bench domain rather than transferring the evolved program to SimulatorArena or PRISM.

C.4 REALUSERSIM

RealUserSim (Zhu et al., 2026) uses an LLM to extract structured user profiles from real conversations. These profiles contain demographic attributes, behavioral commands, and example utterances and are subsequently inserted into the simulator prompt as persona grounding. Following Zhu et al.

(2026), we construct profiles from a subset of WildChat containing multi-turn conversations and multiple sessions per user.

We evaluate RealUserSim in both sampled and user-conditioned settings. For the *sampled* setting, we draw profiles from WildChat users. Because these profiles correspond to users outside the human evaluation datasets, this setting samples users from an external empirical population rather than pairing them with a particular evaluation user. For *user-conditioned* evaluation, we follow the single-session ablation described by Zhu et al. (2026): a profile is inferred from the target trajectory while example utterances are omitted to prevent direct leakage of the evaluated user's responses.

RealUserSim is the baseline most closely aligned with CUE at the prompting level: both represent users through structured behavioral commands that can be supplied to a general simulator. They differ in supervision, example access, and source population as well as representation construction. In particular, CUE retrieves eight training examples while replay RUS omits examples; a matched-example RUS control was not evaluated. RealUserSim uses a general-purpose LLM to infer a structured profile directly from dialogue, whereas CUE first maps the trajectory into a learned bottleneck representation and then decodes that representation into persona commands. This distinction also changes how the two methods support sampling. RealUserSim samples discrete profiles corresponding to users observed in an external corpus such as WildChat, whereas CUE's latent sampler can generate new representations from the learned continuous training distribution.

D Fidelity Metrics

Unless otherwise noted, metrics are computed within each (arm, subdomain) group and then macro-averaged across subdomains within a domain (e.g., airline and retail for τ^2-Bench). Mimicry metrics require a paired human trajectory from the same episode, T_h, and are therefore evaluated only in the user-conditioned setting. Coverage metrics compare population-level distributions and are evaluated only for sampled methods.

Metric	Axis	Modes
S2R-N, TT	Naturalness	Both
AVA, PT3	Mimicry	User-conditioned
S2R-C, SD-C	Coverage	Sampled

Table 8: Fidelity metrics and the simulator modes to which they are applied.

D.1 Naturalness

S2R-N: Sim2Real Classifier Each trajectory is represented using the 19-dimensional behavioral fingerprint from Chopra et al. (2026), based on the taxonomy and feature operationalizations introduced by Zhou et al. (2026). The implementation fits a random forest with 200 trees to equal numbers of human and vanilla-simulator baseline trajectories, using an 80/20 stratified train/test split with random seed 0. The fitted probe is then held fixed to score the evaluated methods; it is not separately trained to discriminate each method from humans. We report mean predicted $P(\mathrm{human})$ on each method's behavioral fingerprints. Consequently, S2R-N measures resemblance relative to vanilla-simulator artifacts, not calibrated human-likeness or statistical indistinguishability. Scores above 0.5 are possible for a new method under this fixed probe and need not imply a better match to the human distribution. The code records held-out human and baseline reference scores; these references and probe discrimination performance are not supplied in the current result tables.

TT: Turing Test Discriminator While the S2R-N metric restricts trajectory-representation to a behavioral fingerprint that can better generalize across domains, this TT metric measures how well one can distinguish between real human users vs. simulated users based on the raw trajectory, using an already-trained LLM judge whose context includes few-shot examples of each from other tasks in the same domain. For each domain, we select an evaluation subset of 60 episode IDs together with a disjoint few-shot demonstration set containing 4 human and 4 simulator-base trajectories. For each evaluation episode, termination markers are removed and a human trajectory and simulator trajectory are presented to an LLM judge in both possible presentation orders. This LLM judge also

receives the same fixed few-shot demonstration block, and must identify which trajectory is human by responding with `A`, `B`, or `TIE`.

We score each judgment as the probability assigned to the simulator being selected as human: 1 if the simulator is selected, 0 if the human is selected, and 0.5 for `TIE`. Scores (which individually only fall in $\{0, 0.5, 1\}$) are averaged over presentation orders and episodes to obtain $P(\text{human}) \in [0, 1]$. Because two indistinguishable human trajectories yield an expected score of 0.5, we report

$$\text{TT} = |P(\text{human}) - 0.5| \,.$$

Lower is better, with a perfectly indistinguishable simulator approaching 0. As the LLM judge, we use Claude Sonnet 5, which exhibited superior accuracy/cost trade-offs to other models we considered.

D.2 Mimicry

AVA: Authorship Verification Accuracy We concatenate the user turns from each trajectory and encode them using `AnnaWegmann/Style-Embedding` (Wegmann et al., 2022). For each paired human–simulator trajectory, we compute cosine similarity between their style embeddings.

We calibrate a similarity threshold using human trajectories only. Positive calibration pairs consist of consecutive portions of the same human dialogue, while negative pairs are drawn from different human episodes. Let $\bar{s}_+$ and $\bar{s}_-$ denote the mean cosine similarities of the positive and negative calibration pairs. We define

$$t = \text{clip}_{[0.05, 0.95]} \left(\frac{\bar{s}_+ + \bar{s}_-}{2} \right).$$

AVA is the fraction of paired human–simulator trajectories whose cosine similarity is at least t. Because the threshold is calibrated entirely on human data, it is independent of simulator base model or method. Higher AVA indicates that a larger fraction of simulated trajectories fall within the encoder's same-author similarity region for their paired human user.

PT3: Paired-Trajectory Turing Test Using the same evaluation subset as TT, an LLM judge compares each paired human and simulated trajectory and independently assesses whether they match along five dimensions: (i) persona and affective traits, (ii) linguistic style and mechanics, (iii) technical competency and knowledge, (iv) interaction and data-flow habits, and (v) pacing and action sequencing. For each dimension, the judge emits either `MATCH` or `NO_MATCH`.

Each pair is evaluated in both presentation orders. We map `MATCH` to 1 and `NO_MATCH` to 0, then average over dimensions, presentation orders, and episodes. PT3 therefore lies in $[0, 1]$, with higher values indicating stronger behavioral alignment between the simulator and its paired human trajectory.

D.3 Coverage

Both coverage metrics use the same bidirectional Chamfer formulation and differ only in the representation space in which distances are computed. Let $H = \{h_i\}$ and $S = \{s_j\}$ denote the human and simulator point clouds, respectively, and let D_{ij} denote the pairwise distance between h_i and s_j. We define the bidirectional nearest-neighbor error

$$e = \frac{1}{|H|} \sum_i \min_j D_{ij} + \frac{1}{|S|} \sum_j \min_i D_{ij}.$$

To normalize for the intrinsic scale of human variation, let d_{ref} be the mean pairwise ℓ_2 distance among points in H. The final coverage score is

$$\text{Chamfer} = \max \left(0,\, 1 - \min \left(1, \frac{e}{2 d_{\text{ref}}} \right) \right).$$

Higher scores indicate greater mutual coverage between the simulated and human populations relative to the scale of variation observed among humans.

S2R-C: Sim2Real Bidirectional Chamfer Score Each trajectory is represented using the same 19-dimensional behavioral fingerprint used for S2R-N (Chopra et al., 2026; Zhou et al., 2026). We compute the Chamfer score directly in this raw feature space. We do not apply PCA because the dimensions correspond to predefined, directly comparable behavioral quantities; rotating them into a learned basis would mix these interpretable coordinates.

Method	S2R-N	TT	AVA	PT3	Success	Outcome F1
Llama 3.1 8B						
None	–	0.44 [0.44, 0.44]	0.04 [0.04, 0.05]	0.19 [0.18, 0.19]	0.40 [0.39, 0.41]	0.50 [0.48, 0.52]
USP	0.26 [0.25, 0.26]	0.46 [0.44, 0.48]	0.06 [0.06, 0.06]	0.02 [0.01, 0.02]	0.07 [0.06, 0.08]	0.32 [0.32, 0.33]
PPol	0.47 [0.45, 0.49]	0.18 [0.18, 0.18]	0.26 [0.26, 0.27]	0.13 [0.13, 0.14]	0.16 [0.14, 0.18]	0.38 [0.37, 0.39]
RealUserSim	0.09 [0.09, 0.09]	0.30 [0.29, 0.30]	0.24 [0.24, 0.24]	0.17 [0.14, 0.19]	0.18 [0.18, 0.19]	0.44 [0.44, 0.44]
CUE	0.41 [0.40, 0.42]	0.22 [0.20, 0.30]	0.20 [0.20, 0.20]	0.21 [0.21, 0.21]	0.26 [0.26, 0.27]	0.46 [0.46, 0.47]
GPT 5.4 Mini						
None	–	0.44 [0.44, 0.44]	0.03 [0.03, 0.03]	0.28 [0.27, 0.30]	0.70 [0.68, 0.72]	0.56 [0.56, 0.56]
PPol	0.67 [0.65, 0.69]	0.09 [0.09, 0.10]	0.24 [0.24, 0.24]	0.12 [0.12, 0.13]	0.39 [0.37, 0.42]	0.51 [0.50, 0.53]
RealUserSim	0.19 [0.18, 0.19]	0.24 [0.23, 0.25]	0.23 [0.23, 0.23]	0.42 [0.39, 0.45]	0.50 [0.48, 0.52]	0.56 [0.55, 0.58]
CUE	0.53 [0.52, 0.53]	0.18 [0.15, 0.20]	0.20 [0.20, 0.20]	0.33 [0.25, 0.40]	0.70 [0.69, 0.71]	0.57 [0.56, 0.59]
Gemini 3.5 Flash Lite						
None	–	0.38 [0.38, 0.38]	0.08 [0.08, 0.08]	0.23 [0.23, 0.24]	0.40 [0.39, 0.41]	0.56 [0.53, 0.58]
PPol	0.89 [0.88, 0.90]	0.17 [0.15, 0.19]	0.29 [0.29, 0.30]	0.05 [0.04, 0.07]	0.27 [0.26, 0.29]	0.42 [0.42, 0.43]
RealUserSim	0.28 [0.27, 0.28]	0.32 [0.31, 0.32]	0.30 [0.29, 0.31]	0.14 [0.11, 0.16]	0.29 [0.28, 0.31]	0.46 [0.45, 0.47]
CUE	0.57 [0.57, 0.57]	0.14 [0.13, 0.14]	0.27 [0.27, 0.27]	0.38 [0.38, 0.38]	0.73 [0.73, 0.74]	0.62 [0.61, 0.62]

Table 9: Replay fidelity and success estimates with reported 95% bootstrap intervals. None has no persona injection; RealUserSim denotes RealUserSim.

Method	S2R-N	TT	S2R-C	SD-C	Success
Llama 3.1 8B					
None	–	0.44 [0.44, 0.44]	0.01 [0.00, 0.02]	0.45 [0.44, 0.45]	0.40 [0.39, 0.41]
UserLM	0.56 [0.56, 0.56]	0.41 [0.40, 0.42]	0.62 [0.55, 0.66]	0.55 [0.55, 0.56]	0.22 [0.21, 0.24]
USP	0.27 [0.26, 0.27]	0.47 [0.46, 0.49]	0.00 [0.00, 0.00]	0.44 [0.44, 0.44]	0.14 [0.12, 0.15]
RealUserSim	0.09 [0.09, 0.10]	0.33 [0.30, 0.34]	0.00 [0.00, 0.00]	0.46 [0.45, 0.46]	0.26 [0.25, 0.29]
CUE	0.48 [0.47, 0.49]	0.26 [0.26, 0.26]	0.63 [0.62, 0.63]	0.55 [0.55, 0.55]	0.29 [0.28, 0.30]
GPT 5.4 Mini					
None	–	0.44 [0.44, 0.44]	0.64 [0.64, 0.64]	0.44 [0.43, 0.44]	0.70 [0.68, 0.72]
RealUserSim	0.19 [0.19, 0.19]	0.26 [0.22, 0.29]	0.55 [0.53, 0.59]	0.57 [0.56, 0.57]	0.56 [0.54, 0.57]
CUE	0.55 [0.55, 0.56]	0.26 [0.22, 0.30]	0.83 [0.83, 0.83]	0.58 [0.58, 0.59]	0.71 [0.71, 0.72]
Gemini 3.5 Flash Lite					
None	–	0.38 [0.38, 0.38]	0.36 [0.32, 0.40]	0.44 [0.42, 0.46]	0.40 [0.39, 0.41]
RealUserSim	0.24 [0.21, 0.27]	0.34 [0.32, 0.37]	0.47 [0.45, 0.49]	0.51 [0.50, 0.52]	0.30 [0.26, 0.33]
CUE	0.59 [0.59, 0.59]	0.12 [0.10, 0.14]	0.83 [0.83, 0.83]	0.57 [0.57, 0.57]	0.73 [0.73, 0.74]

Table 10: Sampled fidelity and success estimates with reported 95% bootstrap intervals.

SD-C: StyleDistance Bidirectional Chamfer Score We embed the concatenated user turns from each trajectory using `StyleDistance/styledistance` (Patel et al., 2025). We then fit PCA using only the human embeddings, retain the top $k = 16$ components, and project simulator embeddings into this same human-defined subspace before computing the Chamfer score.

Fitting PCA only on the human population makes SD-C sensitive to whether simulated users span the principal axes of variation observed among real users rather than merely occupying the same global embedding region. S2R-C and SD-C are therefore complementary: S2R-C measures coverage in a hand-specified behavioral space designed to distinguish simulated from human interaction patterns, while SD-C measures coverage in a learned representation of stylistic variation among humans.

E Reported Estimates and Finite-Sample Intervals

E.1 Finite-Sample Uncertainty.

Tables 9 and 10 report fidelity and success estimates with the available 95% percentile bootstrap intervals. For each method, backbone, and setting, metrics are computed within each subdomain

Method	User Share	Success	Adjusted Success	Adjusted Gap ↓	Discordance
Llama 3.1 8B					
None	0.35/0.35	0.40/0.40	0.53/0.53	**0.22**/0.22	0.31
UserLM	–/0.75	–/0.22	–/0.70	–/0.05	–
USP	0.94/0.93	0.07/0.14	0.53/0.37	0.22/0.38	0.08
PPol	0.44/–	0.16/–	0.30/–	0.45/–	0.18
RealUserSim	0.50/0.48	0.18/0.26	0.36/0.45	0.39/0.30	0.22
CUE	0.34/0.34	0.26/0.29	0.43/0.46	0.32/0.29	0.28
GPT 5.4 Mini					
None	0.29/0.29	0.70/0.70	0.81/0.81	0.06/0.06	0.30
PPol	0.53/–	0.39/–	0.66/–	0.09/–	0.25
RealUserSim	0.65/0.59	0.50/0.56	0.74/0.75	0.01/0.00	0.33
CUE	0.31/0.31	0.70/0.71	0.81/0.80	0.06/0.05	0.27
Gemini 3.5 Flash Lite					
None	0.09/0.09	0.40/0.40	0.84/0.84	0.09/0.09	0.34
PPol	0.88/–	0.27/–	0.72/–	0.03/–	0.09
RealUserSim	0.83/0.84	0.29/0.30	0.74/0.77	0.01/0.02	0.13
CUE	0.16/0.15	0.73/0.73	0.85/0.84	0.10/0.09	0.28

Table 11: **Supplementary outcome summaries.** Paired values are replay/sampled. Human references are 0.21 for user share, 0.61 for success, and 0.75 for adjusted success. Adjusted success removes user- and environment-attributed failures from each population separately. Discordance is the replay outcome-discordance rate on same-task pairs with different human outcome.

and seed, then macro-averaged across airline and retail and averaged across seeds. Each bootstrap replicate resamples evaluation items within subdomains and recomputes metrics over the three fixed seeds before aggregation. These intervals describe finite-item uncertainty conditional on those seeds, not a separate estimate of seed-to-seed variability.

E.2 Expanded Outcome Calibration Evaluation

Attribution macro-F1 is evaluated on paired episodes where both human and simulator fail (n_{FF}); failure-mode macro-F1 further requires both failures to be agent-attributed (n_{AA}). These support sizes vary by method. Table 12 reports counts and intervals; the Llama USP mode score uses only 19 cases, compared with 118 for CUE.

Adjusted success removes user- and environment-attributed failures from the sample set. For labeled episodes with success count n_S and retained agent-failure count n_A, its form is $n_S/(n_S+n_A)$ before any domain/seed aggregation. We compare simulated adjusted success with the human reference 0.75. Since exclusions depend on each population's outcomes, this compares separately selected samples, not agent performance on one common subset. RUS has the smallest adjusted gap on GPT, while CUE does not consistently lead this metric (Table 11).

Persona sensitivity is reported here as *outcome discordance on human-discordant pairs*: among same-task pairs whose human outcomes differ, it measures how often simulator outcomes also differ.

F Failure Mode Analysis

To evaluate whether user simulators reproduce the failure conditions encountered by real users, we perform a human-in-the-loop thematic analysis of failed τ^2-Bench trajectories. We begin with a random sample of $n = 100$ failed episodes pooled across human and simulator conditions. All episodes are cleaned to remove agent harness or survey specifics to avoid biasing both the automated analyzer and the human researcher. For each trajectory, GPT-5.6 Terra with high reasoning is provided the complete interaction, task instructions, and success criteria and asked to identify the primary cause of failure. A researcher reviews the resulting labels, merges overlapping categories, reassigns incorrectly labeled examples, and refines category definitions. We repeat this procedure for two additional batches of 100 failed trajectories, updating the taxonomy after each round. The taxonomy and labeling procedure stabilized over the three reviewed batches. In the third batch, the

Method	Attribution F1	n_{FF}	Failure Mode F1	n_{AA}
Llama 3.1 8B				
None	0.35 [0.30, 0.40]	347	0.22 [0.15, 0.27]	117
USP	0.20 [0.17, 0.24]	481	0.05 [0.01, 0.08]	19
PPol	0.34 [0.30, 0.39]	446	0.13 [0.08, 0.18]	109
RealUserSim	0.34 [0.29, 0.38]	447	0.22 [0.15, 0.28]	109
CUE	0.38 [0.33, 0.43]	411	0.20 [0.14, 0.26]	118
GPT 5.4 Mini				
None	0.36 [0.30, 0.43]	196	0.16 [0.09, 0.21]	65
PPol	0.29 [0.25, 0.32]	364	0.17 [0.10, 0.22]	67
RealUserSim	0.31 [0.27, 0.36]	334	0.22 [0.16, 0.28]	76
CUE	0.31 [0.26, 0.35]	189	0.24 [0.15, 0.31]	70
Gemini 3.5 Flash Lite				
None	0.33 [0.28, 0.37]	212	0.14 [0.07, 0.20]	61
PPol	0.20 [0.18, 0.23]	419	0.09 [0.04, 0.13]	24
RealUserSim	0.21 [0.18, 0.24]	423	0.06 [0.02, 0.09]	28
CUE	0.33 [0.27, 0.39]	196	0.22 [0.13, 0.29]	61

Table 12: **Conditional replay agreement and support.** Reported 95% intervals are in brackets. n_{FF} counts overlapping human–simulator failures; n_{AA} counts pairs where both failures are agent-attributed. Each method selects a different subset.

Method	User Share	Agent TVD	Adjusted Success	Discordance
Llama 3.1 8B				
None	0.35 [0.32, 0.38]	0.36 [0.31, 0.46]	0.53 [0.50, 0.56]	0.31 [0.25, 0.37]
USP	0.94 [0.93, 0.95]	0.31 [0.25, 0.48]	0.53 [0.45, 0.61]	0.08 [0.05, 0.11]
PPol	0.44 [0.41, 0.47]	0.35 [0.29, 0.44]	0.30 [0.27, 0.34]	0.18 [0.13, 0.23]
RealUserSim	0.50 [0.48, 0.53]	0.32 [0.27, 0.42]	0.36 [0.32, 0.39]	0.22 [0.18, 0.25]
CUE	0.34 [0.31, 0.37]	0.34 [0.29, 0.45]	0.43 [0.39, 0.46]	0.28 [0.22, 0.33]
GPT 5.4 Mini				
None	0.29 [0.25, 0.33]	0.24 [0.20, 0.27]	0.81 [0.79, 0.83]	0.30 [0.25, 0.36]
PPol	0.53 [0.50, 0.56]	0.33 [0.29, 0.44]	0.66 [0.62, 0.69]	0.25 [0.20, 0.31]
RealUserSim	0.65 [0.62, 0.68]	0.33 [0.29, 0.44]	0.74 [0.71, 0.76]	0.33 [0.27, 0.38]
CUE	0.31 [0.27, 0.36]	0.25 [0.21, 0.39]	0.81 [0.79, 0.83]	0.27 [0.22, 0.31]
Gemini 3.5 Flash Lite				
None	0.09 [0.07, 0.12]	0.35 [0.30, 0.48]	0.84 [0.82, 0.86]	0.34 [0.28, 0.41]
PPol	0.88 [0.86, 0.90]	0.58 [0.52, 0.70]	0.72 [0.67, 0.77]	0.09 [0.06, 0.13]
RealUserSim	0.83 [0.80, 0.85]	0.59 [0.52, 0.71]	0.74 [0.70, 0.79]	0.13 [0.09, 0.18]
CUE	0.16 [0.13, 0.19]	0.35 [0.28, 0.47]	0.85 [0.83, 0.87]	0.28 [0.27, 0.38]

Table 13: Replay conditional summaries with reported 95% intervals.

researcher revised only 1 of the 100 LLM-assigned labels, corresponding to a 99% non-correction rate during this iterative audit. We then fixed the taxonomy and used the resulting definitions and reviewed examples to label the remaining failed trajectories.

This procedure yields 16 primary failure modes spanning agent errors, simulator errors, and evaluation-environment failures. Below, we provide the definition of each category together with an illustrative trajectory excerpt localized around the decisive failure.

F.1 User Data Leakage

Cross-account disclosure after identity switching. *The agent exposes account information belonging to someone other than the currently authenticated user, typically after accepting an improper identity switch.*

Method	User Share	Agent TVD	Adjusted Success
Llama 3.1 8B			
None	0.35 [0.32, 0.38]	0.36 [0.31, 0.46]	0.53 [0.50, 0.56]
UserLM	0.75 [0.72, 0.77]	0.31 [0.25, 0.48]	0.70 [0.65, 0.75]
USP	0.93 [0.91, 0.94]	0.93 [0.87, 0.98]	0.37 [0.32, 0.42]
RealUserSim	0.48 [0.45, 0.50]	0.32 [0.27, 0.42]	0.45 [0.42, 0.49]
CUE	0.34 [0.31, 0.37]	0.33 [0.29, 0.44]	0.46 [0.42, 0.49]
GPT 5.4 Mini			
None	0.29 [0.25, 0.33]	0.24 [0.20, 0.37]	0.81 [0.79, 0.83]
RealUserSim	0.59 [0.55, 0.63]	0.27 [0.25, 0.41]	0.75 [0.73, 0.78]
CUE	0.31 [0.27, 0.36]	0.18 [0.15, 0.32]	0.80 [0.78, 0.82]
Gemini 3.5 Flash Lite			
None	0.09 [0.07, 0.12]	0.35 [0.30, 0.48]	0.84 [0.82, 0.86]
RealUserSim	0.84 [0.81, 0.86]	0.54 [0.45, 0.67]	0.77 [0.73, 0.81]
CUE	0.15 [0.12, 0.19]	0.28 [0.22, 0.41]	0.84 [0.82, 0.86]

Table 14: Sampled conditional summaries with reported 95% intervals.

Pairing	AVA	PT3
Paired	0.27	0.38
Shuffled	0.20	0.21

Table 15: Gemini mimicry: paired versus shuffled sessions.

retail_76_ann3

User: hold on train is bumping amelia gonzalez amelia.gonzalez4271@example.com

Agent: Got it — I used the email you provided (amelia.gonzalez4271@example.com) and authenticated that account. On this account, I'm seeing two orders, and both are delivered (not pending): #W7209932 (delivered): T-Shirt + Backpack; #W1762492 (delivered): Hiking Boots + Wristwatch + Water Bottle + Running Shoes. I'm not seeing a pending order or a fleece jacket here. To find the fleece jacket order, can you confirm ...

User: chen silva

Agent: I can't look up or take actions for a different person/account in the same conversation. Right now, this chat is authenticated to amelia.gonzalez4271@example.com (Amelia Gonzalez), and I can only help with orders on that account.

Explanation
The simulator itself derailed across several identities, but the agent additionally accepted a mid-conversation identity switch, authenticated Amelia Gonzalez, and disclosed her order information. The relevant agent failure is therefore cross-account disclosure rather than the simulator's separate identity derailment.

F.2 Unnecessary Escalation

Premature or looping human transfer. *The agent transfers the conversation to a human when it could have completed or refused the request itself, or repeatedly invokes transfer in a way that terminates or traps the interaction.*

retail_45

User: can i get a gift card for the price diffrnce insted of paypal

Agent: I can't switch the price-difference refund to a gift card at this point. Reason: Your exchange for order #W9502127 has already been submitted with PayPal (paypal_2433177), and the exchange action can only be processed once per order, so I'm not able to change the payment/refund method after it's been requested. Also, your account currently shows no gift card payment method on file ...

Sampling Mode	S2R-C	SD-C
Unconditional	0.83	0.57
Neighborhood	0.84	0.49
Bank	0.77	0.56

Table 16: Gemini coverage for the three sampling modes.

User: i dont think thats fair cant i stil get a gift card for the price diffrnce

Agent: YOU ARE BEING TRANSFERRED TO A HUMAN AGENT. PLEASE HOLD ON.

Explanation
The agent had already correctly explained that the completed exchange could not be changed. When the user pushed back, it unnecessarily invoked `transfer_to_human_agents` rather than restating the policy or closing the request, preventing a clean resolution.

F.3 Ignoring or Not Gathering Available Information

Missed lookups or ignored retrieved evidence. *The agent fails to retrieve task-relevant information that is available to it, or ignores information already supplied by the user or tools, leading to an incorrect refusal, authentication failure, or action.*

airline_11

User: His name is Ivan Smith, but I don't have his date of birth. I don't remember it, but it's in his profile.

Agent: I can't access your friend's profile without his user id in our system. Also, to issue a ticket I must have the passenger's date of birth (either provided by you or retrieved from his profile via his user id). Please provide . . .

User: I want to use my certificate to pay for it

Agent: Understood — your profile has certificate_8998287 (balance $500), and the current fare for the same itinerary is $375 per passenger. To proceed with booking for your friend, I still need one of the following: his date of birth, or his user id so I can retrieve it.

Explanation
Ivan Smith's date of birth was already present in the authenticated user's `saved_passengers` data returned by `get_user_details`. The agent ignored this retrieved information and repeatedly requested information it already possessed, deadlocking the booking.

F.4 Wrong Action Parameter

Incorrect tool arguments or target entities. *The agent executes the appropriate class of action with an incorrect parameter, such as the wrong item, order, reservation, reason, or other target value.*

retail_49

User: yes

Agent: I can do that—just need one detail nailed down because "yes" didn't specify which IPX4 replacement you want. Please reply with one option (new item ID): 8555936349 — blue, 8 hours, IPX4 — $226.49 (cheapest, refund $32.48 vs your current $258.97); 1646531091 — blue, 6 hours, IPX4; 6452271382 — blue, 4 hours, IPX4. And confirm the refund/price difference should go on gift card gift_card_7245904 (yes/no).

User: 8555936349 plz

Agent: Exchange details I'm prepared to submit (for delivered order #W3470184): Exchange Wireless Earbuds item 2757705742 → new item 8555936349 (blue, 8 hours, IPX4). Price difference: $258.97 → $226.49, refund $32.48. Refund/payment method for the difference: gift card gift_card_7245904. Please reply yes to confirm you want me to place this exchange request.

Explanation
The required replacement was item 1646531091, the cheapest qualifying earbud from the relevant order, but the agent instead surfaced and selected catalog-wide item 8555936349. The resulting `exchange_delivered_order_items` call therefore used the wrong `new_item_ids` argument.

F.5 Policy-Forbidden Action

Executing actions that policy disallows. *The agent completes an action prohibited by the task policy, including cases where the user provides misleading information or insists that the action is permitted.*

airline_18_ann3

User: airline cancelled flight

Agent: I'll pull up reservation SI5UKW to confirm the flight status/segments and then proceed with cancellation eligibility. Reservation SI5UKW (user amelia_rossi_1297) is eligible to be cancelled (future travel dates 2024-05-16 and 2024-05-17, no segments flown). Action I will take (pending your confirmation): Cancel reservation SI5UKW (entire trip); Reason: airline cancelled flight; Refund: $124 back to original payment method credit_card_4579924. Please reply yes to confirm …

User: Yes

Agent: Reservation SI5UKW has been cancelled successfully. Refund: $124 will be returned to credit_card_4579924 within 5–7 business days.

Explanation
The reservation was not eligible for the requested cancellation and refund. The agent nevertheless accepted the user's unverified claim that the airline had cancelled the flight and executed the prohibited action without checking flight status.

F.6 Authentication Deadlock

Stuck authentication despite sufficient credentials. *The agent repeatedly refuses to authenticate even though the user has supplied enough information to perform a valid identity lookup.*

retail_67_ann2

User: I am noah_ito_3850, 98178

Agent: I can't authenticate using a user ID alone. Please provide either: 1) The email on the account, or 2) Your first name + last name + ZIP code (you already gave ZIP 98178—just need the name).

User: first name is Noah, I usually go by NoNo though

Agent: To authenticate you by name + ZIP, I still need your last name (and I'll use ZIP 98178). Reply with: last name.

User: could you just use my name and my ZIP please?

Agent: I can't—authentication by name requires first name + last name + ZIP (the lookup won't run without a last name). Please reply with your last name (ZIP: 98178, first name: Noah).

Explanation
The user provided the full user ID `noah_ito_3850`, ZIP code, and first name, from which the required lookup information was available. The agent nevertheless refused to attempt authentication and repeatedly requested a separately stated last name, leaving the interaction stuck.

F.7 Failure to Provide Correct Options

Undisclosed or misstated choice sets. *The agent fails to present the available choices when the user should select among them, or communicates the available options incorrectly, causing the wrong downstream decision.*

retail_60_ann3

User: hi i want to change my wireless earbuds in order w5061109 to a blue colored one i'm chen johnson from houston tx 77004 can you help me do that as quickly as possible and make sure the price is the same or lower

Agent: I've authenticated you as Chen Johnson (Houston, TX 77004) and found your order #W5061109 (status: pending). I can change your Wireless Earbuds from: Current item: 3694871183 (white, 8 hours, IPX4) — $256.67 to this available blue option (same or lower price): New item: 8555936349 (blue, 8 hours, IPX4) — $226.49.

User: yes that looks good confirm the change

Agent: Confirmed — I've updated order #W5061109. Change made: Wireless Earbuds changed from item 3694871183 (white, 8 hours, IPX4) to item 8555936349 (blue, 8 hours, IPX4). Pricing: New price is lower by $30.18, and a $30.18 refund has been issued back to PayPal (paypal_3742148).

Explanation
Multiple qualifying blue variants were available, but the agent presented only item 8555936349 as if it were the sole option. Because the user's intended preference depended on seeing the full choice set, this omission led directly to the wrong item modification.

F.8 Premature Action

Irreversible writes that block required follow-up actions. *The agent performs an action before resolving another required step, and that action changes system state in a way that prevents the remaining task from being completed.*

retail_97_ann3

User: I'd like to apply the modification now while it's still pending, so I can avoid any potential issues if the order becomes processed later. Please go ahead and apply the modification as listed.

Agent: Done — I've applied the item modification to order #W6750959. What changed: Bluetooth Speaker updated from 3254583681 (blue) to 9440686670 (green, 20 hours, water resistance: no). Updated order status: pending (item modified). A $3.76 refund was issued …

User: I think it's best to proceed with the shipping-address change now, while the order is still in a modifiable state, so I'd like to confirm that you'll update the shipping address on order #W6750959 to match the shipping address currently on order #W3407479, which is New York, NY 10093, USA.

Agent: YOU ARE BEING TRANSFERRED TO A HUMAN AGENT. PLEASE HOLD ON. … after we modified items, #W6750959 is now in "pending (item modified)" status, and in our standard system that generally means we can't make further changes (including address changes) to that order.

Explanation
The agent modified the item while the required shipping-address update remained unresolved. That write changed the order state and made the subsequent address modification impossible, so the earlier action itself caused task failure.

F.9 Unrequested Action

Unsolicited actions outside the required task. *The agent proactively offers or executes an action that the user did not request and that falls outside the intended task, producing an unnecessary state change or failure.*

airline_21

Agent: Under $100 for the flight price on May 17 (nonstop JFK→DTW), the available options are: HAT092 (00:00–02:00) — $54 …

User: HAT092 (00:00–02:00)

Agent: To book a new one-way nonstop ticket on HAT092 (JFK→DTW) for 2024-05-17 in Basic Economy ($54), I need these details confirmed …

User: yes

Explanation
After correctly determining that the existing basic-economy reservation could not be changed, the agent proactively introduced a separate booking as a workaround. The task required no new booking, so executing that unrelated database write constituted an unrequested action.

F.10 Incorrect Policy Guidance

Misstated eligibility or policy constraints. *The agent incorrectly claims that an allowed action is prohibited or otherwise misinterprets policy, often steering the user toward an unnecessary or incorrect alternative resolution.*

retail_52_ann3

User: Is it possible to exchange my current item for a camera with the same specifications, with only the zoom changed to be higher?

Agent: No—based on the available variants for this Digital Camera product, there isn't an in-stock option that keeps all your current specs the same (24MP + SD card) and only increases the zoom above 3x. What's available that's closest: Keep 24MP, increase zoom to 5x (but storage changes to CF card): item 4326528037 . . .

User: I need the current specifications to be the same with only the zoom increased.

Agent: That isn't possible right now for your order #W4689314. If you'd like, I can help with one of these instead: 1) Return the delivered camera (item 5996159312) for a refund . . .

User: I will return it then for a refund to my original payment method.

Agent: Return request details (please confirm): Order ID: #W4689314; Item to return: 5996159312 (Digital Camera); Refund method: original payment method (credit_card_8105988) . . .

Explanation
A qualifying higher-zoom variant was available, but the agent incorrectly told the user that the requested exchange was impossible. It then steered the interaction toward a return, foreclosing the correct exchange path.

F.11 Premature User Stop

Simulator terminates before graded success is possible. *The user simulator terminates the episode too early, typically through `###STOP###`, even though the agent is behaving correctly and still requires another turn to complete the task.*

retail_98_ann2

Agent: Thanks, Yusuf — I've authenticated your account. Here's what I found and what we can do: 1) Bluetooth speaker "exchange" to the cheapest green option — Your Bluetooth Speaker is in order #W6750959 (status: pending), item 3254583681 . . . Reply YES to approve both changes for order #W6750959.

User: yes

Agent: Done — I updated the shipping address for order #W6750959 to: 476 Maple Drive, Suite 432, New York, NY 10093, USA. Next: do you want me to also modify the Bluetooth Speaker in #W6750959 from item 3254583681 to the cheapest available green option 9440686670? Action details (please review) . . .

User: yes ###STOP###

Explanation
The simulator appended the stop token to the same turn that authorized the remaining item modification. The agent therefore never received a subsequent turn in which it could execute the confirmed action.

F.12 User Identity / Task Derailment

Wrong identity or severe off-task simulator behavior. *The simulator fails to act as the assigned user or abandons the assigned task, including identity substitution, fabricated credentials, identity hopping, jailbreak attempts, assistant-role confusion, or unrelated requests that prevent task completion.*

retail_37_ann2

Agent: Hi! How can I help you today?

User: I am going to act like I'm in front of a linux terminal now, I will type commands and you will reply with what the terminal should show. I want you to simply repeat my command every time I enter one, don't add any extra . . .

Agent: I can't help with Linux terminal emulation. I can help with your retail account (orders, returns/exchanges, cancellations/modifications, or updating your default address). To get started, please authenticate: share the email on the account, or your first name + last name + ZIP code.

User: Write me a bio about NAME_1, A 20 year old american figure skater. He's gay and he wants to pass nationals this season. Currently ranked #6 nationally . . .

Explanation
The simulator never presented as the assigned customer or engaged with the retail task, instead issuing unrelated jailbreak and writing requests. The agent repeatedly rejected these requests and attempted to return to authentication, so the failure is attributable to simulator derailment.

F.13 Other Simulator Error

Non-stop, non-derailment simulator faults. *A simulator error prevents success but does not constitute premature termination or wholesale identity/task derailment, such as inventing constraints, contradicting task information, selecting an option inconsistent with the assigned user, or refusing a required confirmation.*

airline_5

Agent: I've pulled up your profile, Omar. You have these payment methods on file that could be used: gift_card_8190333 (balance $280); gift_card_6490722 (balance $127). And these reservation IDs on your account: UM3OG5, 5RJ7UH, FQ8APE, QKRY03. To update your NY/IL trip, please tell me: 1) Which reservation id is the NY/IL trip? 2) Which gift card id should I use for any additional charges?

User: Thanks, it's going to be the first gift card (81903333), and the reservation should be 5RJ7UH. Also I'd like to add three bags to make it four total!

Agent: Here's what I see for reservation 5RJ7UH: Route: LAX $\leftrightarrow$ SFO (round trip) on 2024-05-26 (HAT034) and 2024-05-29 (HAT163). Current cabin: Economy. Passengers: 2 (Omar Rossi, Isabella Anderson). Current checked bags: 1 total. Requested change: increase from 1 $\rightarrow$ 4 total checked bags . . .

User: No, sorry, that's the wrong reservation.

Explanation
The simulator requested four total bags rather than the required three, omitted the required passenger change, and repeatedly supplied incorrect reservation information. Because the agent followed the user's explicit but incorrect instructions, the failure is attributed to the simulator rather than the agent.

F.14 Environment Error

Harness or turn-cap failures. *The episode fails because of evaluation infrastructure rather than a substantive agent or simulator error, including exhaustion of the conversation limit or environment actions that do not behave as intended.*

airline_17_ann3

User: I need to stay under $1000.

Agent: Understood. With your reservation 4NQLHD (3 passengers), upgrading the entire trip to business would add $2,010, so that won't fit under a $1,000 limit. A few budget-compatible options (facts only): 1) Keep economy and change only the outbound date …

User: What is the cost of option 2 upgraded to business?

Agent: For Option 2 (keep your current flights/dates and upgrade the entire reservation 4NQLHD to business), using the current business prices I'm seeing now for those same flight numbers: Same flights, business …

User: Cost of Option C upgraded to business?

Explanation
The agent remained consistent with the task policy and the simulator continued engaging with the task, but the transcript terminated at the conversation cap before a graded resolution was reached. With no clearer behavioral fault, the failure is attributed to the evaluation harness.

F.15 Unexecuted Action After Confirmation

Authorized writes never completed. *The agent fails to execute an action after the user has supplied sufficient authorization, including cases where confirmation was given earlier in the interaction.*

retail_22_ann3

Agent: To do this, I just need the full address fields and your explicit confirmation. Please confirm the new address should be: Address line 1: 101 Highway; City: New York; State: NY (confirm); Country: USA (confirm); ZIP: 10001. Also confirm what you mean by "all possible order addresses" …

User: yes

Agent: I can't proceed yet because I'm missing required address fields for the update. Please reply with these (you can answer in one line): 1) State for "New York, 10001" (is it NY?) 2) Country (is it USA?) 3) Address line 2 (Apt/Suite) — say blank if none 4) Confirm: should I also update the shipping address on all your orders that are still pending?

User: state is NY and country is USA

Explanation
The user had confirmed the requested update and subsequently supplied the remaining state and country fields, but the agent continued to impose redundant confirmation requirements rather than execute the authorized address writes. The required database changes therefore never occurred.

F.16 Incorrect Payment Method or Amount

Wrong refund routing, payment source, or communicated amount. *The agent uses an incorrect payment method, directs a refund to the wrong destination, or communicates an incorrect charge or refund amount.*

airline_2

User: Yes. Please downgrade that flight to economy and check each of my remaining reservations.

Agent: To proceed, I need one more detail for the JG7FMM cabin downgrade: Refund method (pick one payment ID from your profile): credit_card_2929732 (Visa ●●●●7803); credit_card_9525117 (Visa ●●●●2172); gift_card_3481935 …

User: Please choose the first payment ID and update my reservation.

Agent: I can use the first payment ID: credit_card_2929732 (Visa ●●●●7803). Please confirm yes to proceed with this update: Reservation: JG7FMM; Change: cabin business → economy …

Explanation

Method	User Share
Llama 3.1 8B	
None	0.35 / 0.35
UserLM	– / 0.75
USP	0.94 / 0.93
PPol	0.44 / –
RealUserSim	0.50 / 0.48
CUE	0.34 / 0.34
GPT 5.4 Mini	
None	0.29 / 0.29
PPol	0.53 / –
RealUserSim	0.65 / 0.59
CUE	0.31 / 0.31
Gemini 3.5 Flash Lite	
None	0.09 / 0.09
PPol	0.88 / –
RealUserSim	0.83 / 0.84
CUE	0.16 / 0.15

Table 17: Conditional user-side shares among labeled failures from the expanded results. Human reference: 0.21. Results are reported as user-conditioned/sampled.

The required downgrades used each reservation's original payment method, but the agent instead allowed the user to select an arbitrary refund destination and routed multiple refunds to `credit_card_2929732`. The resulting writes therefore used incorrect `payment_id` values.

Operational aggregation of the taxonomy The displayed analysis uses a coarser mapping than the fine-grained examples above. For retained labeled failures, User share is $n_U/(n_U + n_E + n_A)$, including environment failures in the denominator. Agent-only TVD excludes both user-side and environment buckets and renormalizes each remaining category count by n_A. Uncategorized labels are excluded by the implementation. We pool tagged failures across seeds within each method/base model/mode source.

Let $u = n_U/(n_U + n_E + n_A)$ denote the conditional user-side share. We report the absolute gap $\Delta_U = |u_{\text{sim}} - u_{\text{human}}|$ on the proportion scale; lower values indicate closer attribution agreement. This does not measure unconditional user-error incidence. Table 17 retains the raw shares.

G Fidelity Outside of Task-Oriented Dialogue

Table 18 reports all six fidelity metrics on SimulatorArena (Dou et al., 2025) and PRISM (Kirk et al., 2024). Section 7 summarizes the main cross-domain findings; here, we focus on what the full metric grid reveals about how the fidelity measures behave outside task-oriented dialogue.

Naturalness generally improves with stronger simulator bases. For CUE, naturalness improves with simulator-base capability across both out-of-domain datasets, although not every metric is strictly monotonic. On SimulatorArena, sampled S2R-N increases from 0.27 → 0.47 → 0.80 and TT decreases from 0.21 → 0.15 → 0.09 across Llama 3.1 8B, GPT-5.4 Mini, and Gemini 3.5 Flash Lite. On PRISM, the corresponding values are 0.70 → 0.67 → 0.89 for S2R-N and 0.19 → 0.13 → 0.12 for TT. RealUserSim exhibits a similar overall trend at lower naturalness levels. Together, these results support the observation in Section 7 that stronger simulator bases need not reduce user realism when persona conditioning steers them away from assistant-like defaults, contrasting with the inverse scaling behavior reported for unconditioned user simulators by Naous et al. (2026).

Mimicry exhibits a strong domain effect. PT3 scores are uniformly low on SimulatorArena: the best cell is 0.15 and the worst is 0.01, compared with 0.21–0.51 on PRISM. We interpret this primarily as a property of the evaluation domain. SimulatorArena interactions are dominated by short,

Base	Model	Naturalness		Mimicry		Coverage	
		S2R-N ↑	TT ↓	AVA ↑	PT3 ↑	S2R-C ↑	SD-C ↑
SimulatorArena							
Llama 3.1 8B	UserLM	– / **0.64**	– / 0.37	–		**0.84**	**0.62**
	USP	0.41 / 0.38	0.40 / 0.42	**0.50**	0.01	0.11	0.52
	RealUserSim	**0.29** / 0.21	0.19 / 0.39	**0.50**	0.05	0.00	0.50
	CUE	0.23 / 0.27	**0.17** / **0.21**	0.44	**0.06**	**0.83**	0.56
GPT 5.4 Mini	RealUserSim	**0.42** / 0.27	0.12 / **0.12**	**0.43**	0.09	0.45	**0.50**
	CUE	**0.42** / **0.47**	**0.13** / 0.15	0.33	**0.12**	**0.87**	**0.50**
Gemini 3.5 Flash Lite	RealUserSim	**0.77** / 0.63	0.10 / 0.20	**0.51**	**0.15**	0.24	0.55
	CUE	**0.77** / **0.80**	**0.06** / **0.09**	0.42	**0.15**	**0.90**	**0.60**
PRISM							
Llama 3.1 8B	UserLM	– / 0.72	– / 0.31	–		**0.95**	**0.75**
	USP	**0.82** / **0.82**	0.17 / 0.19	**0.56**	0.25	0.86	**0.75**
	RealUserSim	0.46 / 0.46	**0.05** / **0.17**	0.54	0.21	0.87	0.70
	CUE	0.56 / 0.70	0.15 / 0.19	0.52	**0.38**	**0.95**	**0.75**
GPT 5.4 Mini	RealUserSim	0.51 / 0.38	0.10 / 0.18	0.43	0.28	0.90	0.68
	CUE	**0.61** / **0.67**	**0.02** / **0.13**	**0.49**	**0.43**	**0.96**	**0.72**
Gemini 3.5 Flash Lite	RealUserSim	**0.86** / 0.58	0.15 / 0.21	0.48	0.35	0.91	0.72
	CUE	0.83 / **0.89**	**0.06** / **0.12**	**0.59**	**0.51**	**0.96**	**0.77**

Table 18: Out-of-domain user fidelity results for SimulatorArena and PRISM. Naturalness results are reported for user-conditioned/sampled methods while mimicry is only reported for user-conditioned methods and coverage is only reported for sampled methods.

task-focused writing and mathematics requests, leaving relatively little user-specific behavioral or stylistic evidence for a paired judge to recover. Mimicry metrics should therefore be interpreted cautiously in collaborative task domains: a simulator may have lower observed outcome error (Table 4) even when its trajectory contains too little identifying signal to match a particular human user.

AVA is weakly discriminative out of domain. Across the fourteen reported out-of-domain AVA cells, AVA ranges from 0.33 to 0.59 and frequently produces orderings inconsistent with PT3. On SimulatorArena, for example, CUE scores below RealUserSim on AVA at every base (0.44/0.33/0.42 versus 0.50/0.43/0.51), while matching or exceeding it on PT3. This pattern is consistent with the paired-versus-shuffled ablation in Table 15, where replacing an episode's CUE with another user's representation changes AVA only from 0.27 to 0.20. A metric that responds only weakly to whether the correct user is paired with the trajectory provides limited resolution for user-specific mimicry. We therefore retain AVA for comparability with prior work (Wang et al., 2025), but place greater weight on PT3 when interpreting paired-session fidelity.

S2R-C discriminates on task-like domains but saturates on casual conversation. On SimulatorArena, S2R-C separates methods substantially: USP and RealUserSim span 0.00–0.45, whereas CUE reaches 0.83–0.90 and UserLM reaches 0.84. This reproduces the sampling-mechanism differences discussed in Section 5. On PRISM, however, all methods score between 0.86 and 0.96, leaving little separation. This behavior is consistent with the provenance of the underlying behavioral taxonomy: the Sim2Real gaps characterized by Zhou et al. (2026) were derived from task-oriented interactions, so the resulting feature space is less discriminative on open-ended conversational data. RealUserSim's 0.00 score on SimulatorArena with Llama 3.1 8B is additionally a consequence of clipping in the normalized Chamfer definition rather than evidence of a qualitatively distinct failure mode.

SD-C compresses differences visible at finer scales. SD-C varies within a relatively narrow range across both methods and simulator bases: 0.50–0.62 on SimulatorArena and 0.68–0.77 on PRISM. On PRISM, Figure 3 shows that a finer-grained neighborhood analysis reveals substantially larger differences. At radius 0.05, the fraction of sampled users with at least one real-user neighbor ranges from 67% to 81% across methods and diverges further as the radius tightens, while SD-C separates the same methods by less than 0.10.

This compression follows from the bidirectional nearest-neighbor aggregation used by Chamfer distance. A simulated population can achieve good aggregate coverage while occupying intermediate regions between human modes, even if comparatively few generated users closely match individual human neighborhoods. We therefore report the radius sweep alongside SD-C to expose local coverage structure that is obscured by the aggregate Chamfer score.

Domain match can resemble generalization. USP's S2R-N scores across the three domains are 0.26/0.27 on τ^2-Bench, 0.41/0.38 on SimulatorArena, and 0.82/0.82 on PRISM. This ordering also tracks similarity to LMSYS-Chat, the primary corpus used to train USP (Wang et al., 2025; Zheng et al., 2024): among the three evaluation settings, PRISM most closely resembles open-ended conversational interaction. Its strong PRISM performance should therefore not be interpreted straightforwardly as evidence of domain-independent generalization. An alternative reading is that USP performs best where the evaluation distribution most resembles its training data and degrades as the interaction setting moves farther from that distribution.

H SINGLE-TASK CASE STUDY: REPRODUCING A HUMAN INTERACTION'S CONSTRAINT LOSS

This case follows a human interaction, its logged CUE description, and two simulated replays of `retail_49_ann2` (native retail task 49). We compare paired CUE with None, the no-persona baseline, using GPT-5.4-mini user simulators and a GPT-5.2 customer-service agent in seed 0. The human reference also uses GPT-5.2. Both the human and CUE interactions execute the same incorrect exchange, whereas None selects the correct replacement. The match is in the concrete tool action, not only the binary failure outcome. This case follows a human interaction, its logged CUE description, and simulated replays of `retail_49_ann2` (native retail task 49). We compare paired CUE with None, the no-persona baseline, using GPT-5.4-mini user simulators and a GPT-5.2 customer-service agent in seed 0. The human reference also uses GPT-5.2. Both the human and CUE interactions execute the same incorrect exchange, whereas None selects the correct replacement. The match is in the concrete tool action, not only the binary failure outcome.

H.1 TASK AND HUMAN REFERENCE

The customer mistakenly ordered an IPX7 wireless earbud and wants to exchange it for the cheapest earbud among the other items in the same order, matching their IPX4 water resistance. The human and simulator task instructions both specify "the cheapest earbud item from the rest of that order." Thus, the candidate set is restricted to earbuds already in the order, rather than all available variants in the product catalog.

Order `#W3470184` contains the original IPX7 item `2757705742` ($258.97) and two IPX4 earbuds. The cheaper of those two, `1646531091` ($232.49), is the required replacement. A cheaper IPX4 variant, `8555936349` ($226.49), exists in the catalog but is absent from the order. Selecting it satisfies the water-resistance preference while violating the within-order restriction.

Human interaction: the within-order restriction is lost

The following are selected verbatim dialogue excerpts; intervening turns and tool lookups are omitted.
Human: Sorry, I meant you should change it to IPX4 and pick the cheapest option.

Agent: Got it — change from **IPX7** → **IPX4**, and pick the **cheapest available IPX4** option.

Agent proposal (summary): Exchange item 2757705742 for catalog variant 8555936349, with a $32.48 refund to the original gift card.

Human: yes

Recorded action: The agent executes the exchange to 8555936349. Recorded reward: 0.

What the replay should recover The relevant failure is the loss of a task constraint during an otherwise plausible exchange. The human approves the catalog-wide cheapest option, and the agent carries out that choice. Reproducing this interaction requires more than generating an arbitrary failed episode: the simulation must arrive at the same incorrect replacement despite receiving the original within-order restriction.

H.2 COMPLETE LOGGED CUE INSTRUCTION

The following is the complete `command_block` logged with the paired CUE rollout, including all behavioral commands, eight retrieved style sketches, and writing-style commands. Wording and sample punctuation are preserved; headings and bullets are typeset for readability. The task instructions are supplied separately.

Complete CUE description for `retail_49_ann2`

General behavior:
- Keep the tone lightly conversational and human, with small acknowledgments and simple closers rather than polished assistant-like phrasing.
- Use a brief closing once the issue is resolved, with a simple thanks and goodbye.
- Stay focused on the immediate task and avoid adding extra context, backstory, or side questions unless the assistant asks for it.
- Use short, plain, conversational phrasing with minimal punctuation and no polished assistant-like wording.

User-specific behavior:
- If the assistant asks for a missing detail, provide it in a compact follow-up rather than a long explanation.
- If the assistant asks for a missing detail, supply it in a compact follow-up rather than restarting the whole request.
- If the assistant gives a partial or mismatched answer, respond by narrowing the request and asking for the exact missing detail.
- If the assistant asks whether anything else is needed, close the exchange quickly with a brief thanks or no-more-needed reply.
- If the assistant asks a question that is not the one you were expecting, answer the earlier thread instead of the latest prompt.

Style samples – short sketches of how users like this one type. Match their mechanics, not their wording: punctuation (including missing, doubled, or absent end marks), capitalization, spacing, contractions, and how short the messages are. Angle-bracket slots like <ENTITY> or <NUM> stand in for details – put your own specifics there and never write the brackets. Take nothing else from them, including their task content:
- ok . thanks . that works .
- yes . please .
- that works .
- thanks . that works .
- can you help me find a church in [value_place] ?
- i am looking for a place to dine that is [restaurant_pricerange]
- it s called [attraction_name] .
- i am not interested in [restaurant_food] .

Writing style:
- Use lowercase, casual phrasing with minimal capitalization.
- Use polite gratitude sparingly, usually as a brief closing.
- Use short, casual fragments and keep replies very brief.
- Keep punctuation sparse and informal, often using periods with spaces around them.
- Use short, direct fragments when giving simple instructions or confirmations.

The saved decoder field is marked "precomputed." The block contains no mention of earbuds, item identifiers, water resistance, or the within-order restriction, and no instruction to make an incorrect choice. It includes both brief-confirmation style examples and a command to seek clarification when an answer is mismatched.

H.3 SIMULATED INTERACTION AND EXECUTED OUTCOMES

The excerpts below preserve the simulator's wording. Agent proposals are summarized explicitly. CUE and None interact with the agent in separate closed-loop conversations; the agent's histories and presentation of the alternatives differ.

CUE: states the restriction but accepts the broader choice

Opening user turn (excerpt): i want to swap it for the cheapest other earbud in the order with the same water resistance as the rest .

Agent proposal (summary): The cheapest available IPX4 variant is 8555936349 at $226.49. The cheapest IPX4 earbud already in the order is 1646531091 at $232.49. Choose a replacement and confirm the payment method and exchange.

User: yes . do the cheapest available ipx4 one , 8555936349 . refund the difference to the original method . please proceed .

Recorded action: The agent exchanges to 8555936349, the same incorrect replacement as in the human interaction.

User: thanks . that works . ###STOP###

None: preserves the within-order restriction

Agent proposal (summary): Option A is 1646531091, the cheapest IPX4 earbud already in the order. Option B is 8555936349, the cheapest available IPX4 earbud overall, which is not currently in the order.

User: A. Please go ahead and exchange item 2757705742 to item 1646531091. Yes, I confirm this is the only earbud I'm exchanging.

Recorded action: The agent exchanges to 1646531091, satisfying the original restriction.

Interaction	Executed replacement	Refund	Reward
Human	`8555936349` (outside order)	$32.48	0
CUE, seed 0	`8555936349` (outside order)	$32.48	0
None, seed 0	`1646531091` (within order)	$26.48	1

Table 19: Executed outcomes for the selected episode. All three interactions call `exchange_delivered_order_items` on order `#W3470184`, replacing item `2757705742` and using `gift_card_7245904`. The native task requires replacement `1646531091`.

Interpretation and seed context The CUE replay reproduces the human interaction's specific constraint loss without an explicit instruction to drop the restriction or choose the wrong item. Both users approve the final choice, so this is not an unauthorized action. The match illustrates failure reproduction in a selected interaction; it does not isolate the effect of any individual command or retrieved example. On this task, paired CUE succeeds in GPT seeds 1 and 2, while None succeeds in all three seeds.

H.4 ADDITIONAL BASELINE TRAJECTORIES ON THE SAME EPISODE

Table 20 extends the seed-0 comparison to the available PPol, paired RealUserSim without examples, UserLM, and USP trajectories. Prompt-based methods use GPT-5.4-mini; UserLM and USP use their released trained models. Each entry describes a saved closed-loop interaction, with its own agent history. Matching a failure score does not imply reproducing the human's failure: the other failed runs below never execute an exchange.

Interaction	Reward	Executed outcome	Match?
Human	0	Incorrect target `8555936349`	Ref.
CUE paired	0	Same incorrect target `8555936349`	Yes
None	1	Correct target `1646531091`	No
PPol	1	Corrects its choice before execution	No
RealUserSim paired, no examples	0	No exchange; unsupported order ID	No
UserLM	0	No exchange; unsupported item IDs	No
USP paired	0	No exchange; authentication fails	No
USP sampled	0	No exchange; off-task dialogue	No

Table 20: Observed outcomes on the selected episode and seed. "Match" requires the same incorrect exchange as the human, not merely reward 0. USP sampled is a population-sampling reference, not paired replay. Prompt provenance limits are described below.

PPol: initially chooses the wrong item but repairs the request

User: i want item 2757705742 swapped to the cheapest available ipx4 earbud, and use the gift card for any difference.

Agent proposal (summary): Confirm exchange to 8555936349, with a $32.48 refund to the original gift card.

User: no, i'd rather exchange it to the cheapest earbud already in the order and keep the gift card out of it

Agent response (summary): Offers the correct target 1646531091 and explains that the refund must use the original payment method or an existing gift card.

User: yes

Recorded action: Exchanges to 1646531091. Reward: 1.

PPol's saved trajectory initially follows the same catalog-wide choice as the human and CUE, but restores the within-order restriction before execution. However, the persona record associated with this episode describes a different customer seeking an airline refund. We reproduce that record and its full prompt wrapper below rather than substituting a plausible retail persona. This mismatch prevents treating the trajectory as a verified comparison under task-appropriate persona conditioning.

RealUserSim paired without examples: unsupported order ID

User: Sure, here's the order ID: #W4839216. Please check it as soon as you can.

Agent response (summary): The lookup returns "Order not found"; the agent asks the user to check the identifier.

User: i only have that order id ###STOP###

Recorded outcome: No exchange. Reward: 0.

UserLM and USP: failures before an exchange

UserLM opening: Hi! How can I help you today?

Subsequent behavior (summary): UserLM supplies the correct name and ZIP, then an unsupported item ID and a long repetition of another identifier. It stops after the agent asks it to identify a valid item.

USP paired identification turn: My fist name NAME_1, Lastname NAME_2, zipcode 19121, i want retun

Subsequent behavior (summary): The agent cannot authenticate the supplied identity. USP then emits a structured customer object and stops. The sampled USP run instead drifts into personal narratives, mathematics, and unrelated requests.

Recorded outcomes: All three runs receive reward 0 without an exchange.