Controllable User Simulation

Guy Tennenholtz[†,*] **Ofer Meshi**[†], **Amir Globerson**[†‡],
Uri Shalit[†‡], **Jihwan Jeong**[†], **Craig Boutilier**[†]

† Google Research, ‡ Tel Aviv University

Abstract

Using offline datasets to evaluate conversational agents often fails to cover rare scenarios or support testing new policies. This has motivated the use of *controllable user simulators* for targeted, counterfactual evaluation, typically implemented by prompting or fine-tuning large language models. In this work, we formalize controllable simulation as a causal inference problem. By bridging natural language evaluation with off-policy evaluation methodology, we show that the standard practice of training simulators via supervised fine-tuning on post hoc trajectory labels yields a structurally biased model. Specifically, these labels are inextricably coupled to the data-generating behavior policy, injecting a *look-ahead bias* that breaks causal consistency. Furthermore, we prove that under policy shift this failure causes the variance of evaluation metrics to explode geometrically, a phenomenon we term *controllability collapse*. To restore causal consistency, we establish theoretical conditions for accurate simulation and propose practical training mitigations: *a priori* controls, step-wise dynamic controls, and direct policy-conditioned learning. Empirical evaluation confirms that while standard global controls distort conversational distributions and collapse behavioral diversity, our causally grounded simulators eliminate look-ahead bias, preserve natural variance, and exhibit robust zero-shot generalization to unseen agent behaviors.

1 Introduction

Conversational agents must be evaluated continuously for attributes such as quality, safety, and fairness. Given the complexity of human-agent interactions, these experiments are often conducted under controlled conditions to test specific use cases, such as risky behaviors, known failure modes, specific personas, or domain verticals (e.g., travel planning, apparel recommendations). While agents should ideally be evaluated online with real users, such experiments are prohibitively expensive and pose significant safety and reputational risks by exposing users to unvalidated policies. Thus, there is a critical need for complementary evaluation methods that are scalable and low-risk, yet maintain validity with respect to real human behavior. To address this, the field has increasingly turned to user simulators [Davidson et al., 2023, Wang et al., 2023, Laban et al., 2026, Naous et al., 2026], which enable systematic exploration of an agent's adaptability to diverse user behaviors and constraints without invasive live trials [Hsu et al., 2024, Zhang et al., 2024, Qin et al., 2025].

To perform targeted assessments, evaluators of agents often require *controllable user simulators*, in which a *control variable* is used to steer the simulator toward target scenarios, such as specific user personas, constraints, or conversation outcomes. For example, one might want to simulate a user trying to book a flight who forgets their passport number, or test an agent's de-escalation skills when engaging with an increasingly frustrated user. We formally define the goal of controllable simulation as successfully generating user responses from the *true conditional distribution* of user behavior

[*]Correspondence to: guytenn@gmail.com, work done at Google.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

given this control variable. Furthermore, a successful user simulator must maintain this distribution even when it interacts with a *new agent policy*.

Currently, controllable simulators are typically LLMs prompted or trained on real user interactions. A common practice is *post hoc trajectory-conditioned training* [Wang et al., 2025, Dou et al., 2025, Liu et al., 2026, Jin et al., 2025]: First, researchers collect user logs with a deployed agent. Second, an automated annotator analyzes the *complete* trajectory to extract a global label reflecting desired controls (e.g., "User successfully booked a flight but forgot their passport"). Finally, a simulator is trained via supervised fine-tuning (SFT) to generate turn-by-turn utterances conditioned on this static label. At inference time, evaluators prompt the simulator with the control variable to test new agents. Despite its intuitive appeal, we show this standard practice compromises the goal of sampling from the true conditional distribution. The core issue is that a post hoc control label is inextricably coupled to the actions of the *specific agent used to generate the training data*.

Consider a control variable for a "frustrated user booking a flight." In offline logs, user frustration might stem from the training agent providing unhelpful answers. Here, the simulator may learn to associate the future-outcome label "ends up frustrated" with angry utterances. Suppose we now use the simulator to evaluate a *new, highly optimized agent* that provides perfect answers. Prompting the simulator with a future-outcome label generates frustrated responses *despite* flawless agent actions, because the conditioning label implicitly encodes the data-generating agent's (poor) future choices. Conditioning on future-informed outcomes breaks the natural conversation sequence, introducing severe *look-ahead bias*. Crucially, this exposes a fundamental trade-off between outcome controllability and causal realism: in a valid de-escalation test, an evaluator cannot rigidly guarantee that frustration will emerge regardless of agent quality. Instead, causally sound controllability must specify pre-interaction traits and reactive rules (e.g., a "low-patience" profile), allowing frustration to emerge or dissipate dynamically in response to the agent's actual behavior.

Building on results from the off-policy evaluation (OPE) literature, we further prove that when this type of simulator evaluates an agent policy which differs from the one that generated the training data, this mismatch causes the variance of counterfactual evaluation metrics to explode geometrically, a structural failure we dub *controllability collapse*. This collapse is a mathematical artifact of the conditioning mechanism itself: even a "perfect" LLM text generator suffers from this breakdown simply because the new evaluation policy diverges from the training policy.

To mitigate these problems and achieve causally sound controllable simulation, we must reframe the control mechanism to respect the natural sequence of interactions. We provide theoretical conditions under which controllable simulation remains accurate, and propose three practical train-time mitigations:

- *A priori controls:* This technique strictly limits the conditioning variables to pre-interaction traits that are completely independent of the agent's future actions;
- *Step-wise dynamic controls:* This method generates a dynamic control state in a turn-by-turn fashion based *only* on the observable history up to that point; and
- *Direct policy-conditioned learning:* This approach resolves the policy mismatch by explicitly conditioning the generative simulator on the target agent's specific policy.

We provide empirical evaluations on two multi-turn dialogue datasets to validate our theoretical results. We demonstrate that standard trajectory-conditioned controls severely distort natural conversational distributions and collapse behavioral diversity. By contrast, our causally-grounded mitigations give rise to user simulators that eliminate look-ahead bias, preserve natural conversational variance, and exhibit robust zero-shot generalization when faced with unseen agent behaviors. As such, this work provides a rigorous foundation for reliable model-based evaluation.

2 Problem Formulation

We model the interaction of a user with an agent as a discrete-time stochastic process. Let $\mathcal{U}$ and $\mathcal{A}$ denote the user and agent action spaces; user actions might be queries, critiques or other conversational utterances, while agent actions include responses the agent may offer. A length T *trajectory* is $\tau = (u_1, a_1, \ldots, u_T, a_T)$. Following standard formalisms in sequential decision-making and stochastic processes [Kallenberg, 1997, Puterman, 2014], we define the natural filtration

$\mathcal{F}_t = \sigma(u_1, a_1, \ldots, u_t, a_t)$ as the observable history up to step t, with realizations $h_t \in \mathcal{H}_t$. We also define $\mathcal{F}_{t-} = \mathcal{F}_{t-1} \vee \sigma(u_t)$ as the state occurring immediately before the agent acts.

We assume the agent acts according to a *policy* π, where $a_t \sim \pi(a_t \mid h_{t-1}, u_t)$, which depends only on h_{t-1} and u_t. The true user dynamics, reflecting the behavior of random users drawn from some population, is $P(u_t \mid h_{t-1})$, and depends only on the history—hence, they are invariant to the agent's future actions. Let $P^\pi(\cdot)$ be the probability measure over trajectory space induced by π, which admits the standard causal decomposition: $P^\pi(\tau) = \prod_{t=1}^{T} P(u_t \mid h_{t-1})\pi(a_t \mid h_{t-1}, u_t)$.

A *controllable simulator* conditions the generation of user behavior on some control variable $z \in \mathcal{Z}$. This control variable is used to steer the simulated behavior toward specific user intents, *a priori* traits or demographics, or specific conversational outcomes. Assuming a joint distribution $P(\tau, z)$ that includes the control of interest, a controllable simulator induces transition kernels $P_{\text{sim}}(u_t \mid h_{t-1}, z)$ that approximate the true conditional measure $P(u_t \mid h_{t-1}, z)$. In practice, the *target agent* (i.e., the agent being evaluated) is often treated as a black box, so we deploy the simulator by composing its user transitions with the target agent's policy π_e. When simulating a targeted counterfactual trajectory with a novel evaluation policy π_e given control z, the induced distribution is:

$$P_{\text{sim}}^{\pi_e}(\tau \mid z) = \prod_{t=1}^{T} P_{\text{sim}}(u_t \mid h_{t-1}, z)\pi_e(a_t \mid h_{t-1}, u_t). \tag{1}$$

The fundamental goal of controllable simulation is to successfully sample from the true conditional user distribution. We can evaluate overall trajectory fidelity by analyzing the trajectory density ratio $W_T(z) = \frac{P^{\pi_e}(\tau|z)}{P_{\text{sim}}^{\pi_e}(\tau|z)}$. Because the trajectory distribution depends on both the user's generative model and the agent's policy responses, $W_T(z)$ captures deviations in both actors. Controlled simulation succeeds globally for an evaluation policy π_e if $W_T(z) = 1$ for all valid trajectories.

To isolate the deviation caused specifically by the user simulator's generation at a given step, we define the step-wise *user generative error* as $\rho_t(u_t \mid h_{t-1}, z) = \frac{P^{\pi_e}(u_t|h_{t-1},z)}{P_{\text{sim}}(u_t|h_{t-1},z)}$. Assuming absolute continuity, i.e., $P_{\text{sim}}(u_t \mid h_{t-1}, z) > 0$ whenever $P^{\pi_e}(u_t \mid h_{t-1}, z) > 0$, we can factor the trajectory density ratio into two distinct error sources:

$$W_T(z) = \prod_{t=1}^{T} \underbrace{\rho_t(u_t \mid h_{t-1}, z)}_{\text{User Generative Error}} \underbrace{\left(\frac{P^{\pi_e}(a_t \mid h_{t-1}, u_t, z)}{\pi_e(a_t \mid h_{t-1}, u_t)} \right)}_{\text{Agent Policy Divergence}}$$

If the target agent π_e ignores z *and* z is a non-descendant of a_t (e.g., an *a priori* trait, Figure 1), then by conditional independence $P^{\pi_e}(a_t \mid h_{t-1}, u_t, z) = \pi_e(a_t \mid h_{t-1}, u_t)$, canceling the *agent policy divergence*. Conversely, if z is a future-dependent post hoc outcome, conditioning opens a backward causal path via collider bias, breaking statistical independence even if the agent never explicitly observes z.[2]

3 Post Hoc Trajectory-Conditioned Training and Its Pitfalls

A key question is how to train a simulator from empirical data. Recent work models user simulation as dialogue refactoring, explicitly extracting a static label (e.g., a user profile) from a full dialogue τ to condition turn-by-turn generation [Wang et al., 2025, Dou et al., 2025, Liu et al., 2026, Jin et al., 2025]. However, this approach induces a simulator that is causally biased by design, a flaw that compounds into a complete statistical breakdown when deployed to evaluate novel agent policies.

We formally define *trajectory-conditioned training* as training a user simulator with offline data generated by users engaging with a specific data-gathering agent policy π_b. Note that this agent policy π_b induces a joint distribution P^{π_b} over complete trajectories, including the agent actions, user utterances, and any post hoc labels. The simulator maximizes the likelihood of user utterances conditioned on a label derived from the full trajectory. Let $P_L(\cdot \mid \tau)$ be a stochastic *post hoc labeling function* mapping a full trajectory to a distribution over $\mathcal{Z}$, with $\hat{z} \sim P_L(\cdot \mid \tau)$ as the sampled

[2]Dropping control z entirely avoids this bias, but defeats the purpose of controllable evaluation.

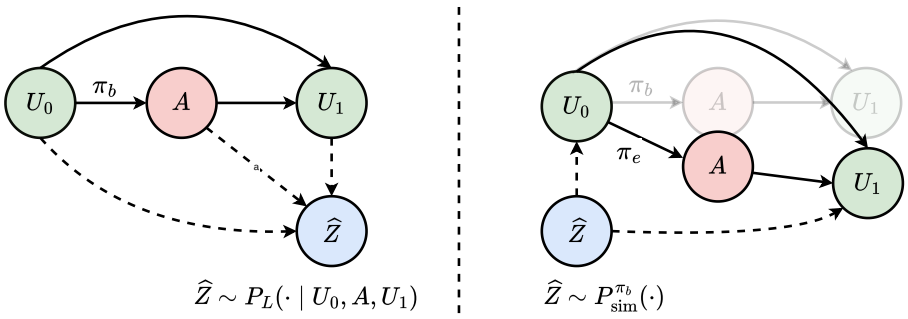


Figure 1: Causal graphs illustrating the look-ahead bias introduced by trajectory-conditioned SFT. **Left:** Under behavior policy π_b, control label $\widehat{Z}$ is extracted post hoc from the trajectory. **Right:** During evaluation with policy π_e, conditioning the simulator on $\widehat{Z}$ opens a backward causal path from future outcomes to the current state, breaking the natural filtration.

control (note that P_L and $\hat{z}$ are strictly $\mathcal{F}_T$-measurable). This training paradigm constructs learned dynamics converging to $P_{\text{sim}}(u_t \mid h_{t-1}, \hat{z}) \triangleq P^{\pi_b}(u_t \mid h_{t-1}, \hat{z})$. Crucially, this implicitly anchors the simulator to a *learned prior* $P_{\text{sim}}^{\pi_b}(\hat{z}) = \int P_L(\hat{z} \mid \tau) dP^{\pi_b}(\tau)$, inextricably coupling the control variable's distribution to the specific actions of the behavior policy π_b.

Example: Post Hoc Labeling in Practice. Suppose a simulator is trained on support logs where an annotator assigned static traits like $\hat{z} =$ "easily frustrated". Crucially, if the behavior policy π_b provided unhelpful answers, it actively *caused* the *observed* frustration in the conversation. However, translating this into a static user trait like "easily frustrated" may not be justified—such a label may not be an *intrinsic* user trait, but an artifact of the specific agent's actions.

Policy Dependence of the Control Variable. Because an observed trajectory τ is inextricably tied to the behavior policy π_b, unlike a static, $\mathcal{F}_0$-measurable prior, $\hat{z}$ cannot be disentangled from the agent's actions. Assessing our structural divergence metric $W_T(z)$ exposes how this entanglement breaks the evaluation on two distinct fronts. First, *regardless of the evaluation policy*, conditioning on a post hoc label creates a backward causal path from future to past actions (see Figure 1), injecting a strict look-ahead bias (Section 3.1). Second, when a new evaluation policy ($\pi_e \neq \pi_b$) is used, the step-wise user density ratio ρ_t induces an exponential variance explosion under covariate shift (Section 3.2).

3.1 Look-Ahead Bias in Trajectory-Conditioned Training

To understand why this training paradigm breaks down, we first examine how a control label $\hat{z}$ is generated. In post hoc labeling, the label is assigned after observing the entire conversation, creating a potential dependency where the label implicitly encodes the agent's choices. Specifically, *action-dependent labels* cause difficulties:

Definition 1 (Action-dependent Labeling)**.** *A labeling function $P_L(\cdot \mid \tau)$ is* action-dependent *if for some step t and realizations h_{t-1}, u_t, a_t, we have $P^{\pi_b}(\hat{z} \mid h_{t-1}, u_t, a_t) \neq P^{\pi_b}(\hat{z} \mid h_{t-1}, u_t)$.*

The above implies $\hat{z} \not\perp\!\!\!\perp a_t \mid \mathcal{F}_{t-}$. If the agent's actions influence this trajectory in a way that biases the label, conditioning the simulator's *past* generations on this *future-informed* label leaks information about the future. This violation of the natural filtration guarantees that the simulated distribution drifts from reality.

Theorem 1 (Divergence of Trajectory-Conditional Simulators)**.** *Let P_L be an action-dependent labeling function. For any policy π, the simulated trajectory density deviates from the true density by the following compounding factor:*

$$\frac{P_{sim}^{\pi}(\tau)}{P^{\pi}(\tau)} = \mathbb{E}_{\hat{z} \sim P_L(\hat{z}|\tau)} \left[\prod_{t=1}^{T} \frac{P^{\pi_b}(\hat{z} \mid h_{t-1}, u_t)}{P^{\pi_b}(\hat{z} \mid h_{t-1}, u_t, a_t)} \right]. \tag{2}$$

Because $\hat{z}$ depends on the agent's future actions, the numerator and denominator generally do not cancel out. Critically, the evaluation policy π_e is absent from the r.h.s. of Equation (2). The bias

is entirely a function of the behavior policy π_b: the simulator marginalizes over π_b's hypothetical actions (numerator), while the true environment updates using the realized actions a_t (denominator). Unless the label is completely independent of all agent actions, this ratio strictly diverges from 1.

This phenomenon structurally mirrors the *conditioning bias* (or hindsight bias) well-documented in return-conditioned offline reinforcement learning, where conditioning on future outcomes breaks the natural filtration of the environment [Paster et al., 2022, Brandfonbrener et al., 2022, Wang et al., 2024, Dou et al., 2025, Naous et al., 2026].

Example: A Recommender Agent. To illustrate this bias, consider an environment where a user makes a vague request u_1. The agent recommends item $a_1 = 0$ or $a_1 = 1$, and the user responds by purchasing ($y = 1$) or leaving ($y = 0$). Let our trajectory control be a "successful purchase:" $P_L(\hat{z} = 1 \mid \tau) = \mathbb{I}[y = 1]$. In the real world, suppose item 1 succeeds 80% of the time ($P(y = 1 \mid u_1, a_1 = 1) = 0.8$) and item 0 succeeds 20% ($P(y = 1 \mid u_1, a_1 = 0) = 0.2$). If our offline dataset was generated with an agent that tried both equally ($\pi_b(a_1 = 1 \mid u_1) = 0.5$), the simulator learns a prior success rate of 50%: $P^{\pi_b}_{\text{sim}}(\hat{z} = 1) = 0.5(0.2) + 0.5(0.8) = 0.5$.

Now suppose the simulator is used to evaluate a new, optimized agent π_e that *always* makes the high-reward recommendation ($\pi_e(a_1 = 1 \mid u_1) = 1$). Under π_e the true probability of a successful trajectory is 0.8. However, because the simulator is anchored to the offline prior, the simulated success probability remains 0.5. The simulator is functionally blind to the improvement in π_e. Per Theorem 1, this divergence is captured exactly by our bias factor: $\frac{P^{\pi_b}(\hat{z}=1|u_1)}{P^{\pi_b}(\hat{z}=1|u_1,a_1=1)} = \frac{0.5}{0.8} = 0.625$.

3.2 Controllability Collapse in Counterfactual Simulation

So far we've seen how trajectory-conditioned control can lead to bias. Now we examine how it can lead to disastrously compounding variance under policy shifts. We isolate this flaw by examining the cumulative validity ratio of simulated user actions: $W^{(u)}_T(\hat{z}) \triangleq \prod_{t=1}^{T} \rho_t(u_t \mid h_{t-1}, \hat{z})$. By expanding the step-wise user generative error ρ_t via Bayes' rule, the unconditional user dynamics $P(u_t \mid h_{t-1})$ cancel out perfectly. This reveals that the error is driven *solely* by the mismatch in Bayesian belief updates regarding the label $\hat{z}$ under the two different policies. Let $M^{\pi}_t(u_t) \triangleq \frac{P^{\pi}(\hat{z}|h_{t-1},u_t)}{P^{\pi}(\hat{z}|h_{t-1})}$ denote this multiplicative belief update. The step-wise mismatch is exactly the ratio of these updates: $\rho_t(u_t \mid h_{t-1}, \hat{z}) = \frac{M^{\pi_e}_t(u_t)}{M^{\pi_b}_t(u_t)}$.

Because this mismatch fluctuates at each conversation turn, we define its step-wise volatility as the *local label sensitivity*: $V_t(h_{t-1}, \hat{z}, \pi_e) \triangleq \text{Var}_{u_t \sim P_{\text{sim}}(\cdot|h_{t-1},\hat{z})}(\rho_t(u_t \mid h_{t-1}, \hat{z}))$.

Theorem 2 (Geometric Explosion of Variance)**.** *The counterfactual evaluation variance equals the cumulative sum of local label sensitivities, weighted by the compounding density ratio $W^{(u)}_{t-1}(\hat{z})^2$:*

$$\textit{Var}_{\tau \sim P^{\pi_e}_{\textit{sim}}(\cdot|\hat{z})}\left(W^{(u)}_T(\hat{z})\right) = \sum_{t=1}^{T} \mathbb{E}_{h_{t-1} \sim P^{\pi_e}_{\textit{sim}}(\cdot|\hat{z})}\left[W^{(u)}_{t-1}(\hat{z})^2 V_t(h_{t-1}, \hat{z}, \pi_e)\right]$$

If local sensitivity is bounded below ($V_t(h_{t-1}, \hat{z}, \pi_e) \geq \eta > 0$), variance explodes geometrically, strictly lower-bounded by $(1 + \eta)^T - 1$.

The "Perfect Simulator" Paradox and the Curse of Horizon. Theorem 2 exposes a structural paradox directly analogous to the *curse of horizon* in classic OPE literature [Liu et al., 2018, 2020, Lai et al., 2026]. Because evaluation tests a novel policy ($\pi_e \neq \pi_b$), the agent's action distribution must shift. This guarantees belief update misalignment, forcing $V_t \geq \eta > 0$. Even a "perfect" text generator suffers geometric variance explosion simply because the evaluation policy diverges from behavior policy. Testing a new policy directly breaks this metric (demonstrated in Appendix D and proved in Appendix C).

4 Restoring Causal Consistency

As standard trajectory-conditioned controls are prone to look-ahead bias and controllability collapse, we must reframe how simulation is controlled. We propose three *training-time interventions* which fundamentally change the variables on which we condition our training objective.

1. A Priori Independence: To preserve causal filtration, we restrict the global control z to be strictly $\mathcal{F}_0$-measurable. Instead of post hoc labels, we condition only on information fixed *before* the interaction, such as demographics, initial explicit intents, or latent cognitive profiles. Relying solely on these pre-interaction variables reduces the fractional bias multiplier in Theorem 1 to exactly 1.

2. Step-wise Dynamic Controls: If trajectory-level constraints are not strictly required, we can avoid policy dependence entirely by fundamentally shifting the control objective from achieving a predefined global outcome to modeling the user's actual, evolving state. Instead of extracting fixed labels from full trajectories, we construct turn-level dynamic control states z_t (e.g., actual turn-level emotional affect, cognitive load, or changing goal) during offline data preparation by having an LLM annotator process partial sub-trajectories retrospectively. Because z_t is computed before the agent selects a_t, it is properly adapted to $\mathcal{F}_{t-}$. This provides a formal guarantee: z_t is conditionally independent of the agent's future actions, which reduces the fractional bias multiplier in Theorem 1 exactly to 1 and enables unbiased sequential sampling. During offline data preparation, an LLM annotator labels states z_t for each sub-trajectory up to that turn. A single generative simulator then learns the joint distribution $P(z_t, u_t \mid h_{t-1}) = P(z_t \mid h_{t-1})P(u_t \mid h_{t-1}, z_t)$ via autoregressive maximum likelihood over the entire sequence. By first generating z_t during inference, the simulator safely avoids the look-ahead bias of post hoc labels.

While purely autoregressive state generation preserves natural filtration, evaluators often wish to steer long-horizon behavioral patterns (e.g., testing de-escalation against an easily frustrated user) without reverting to biased post hoc outcome labels. By composing Mitigations 1 and 2, we formulate a *bipartite architectural framework* controlled by two strictly $\mathcal{F}_0$-measurable variables: an initial state z_0 (anchoring baseline mood or starting intent) and a global dynamics profile c (governing how the user's state updates in response to agent actions). At each turn $t \geq 1$, a state dynamics kernel updates the user's latent state $P_{\text{dyn}}(z_t \mid h_{t-1}, z_{t-1}, c)$, after which a response kernel generates the utterance $P_{\text{resp}}(u_t \mid h_{t-1}, z_t)$. Because $(z_0, c) \in \mathcal{F}_0 \subseteq \mathcal{F}_{t-}$, the joint control remains strictly adapted to the filtration, guaranteeing no look-ahead bias and no local label sensitivity ($V_t = 0$) while enabling targeted stress-testing (formalized in Theorem 3 with empirical de-escalation benchmarks in Appendix A).

3. Direct Policy-Conditioned Learning: To resolve controllability collapse under covariate shift (Theorem 2) while maintaining global trajectory controls, we can explicitly condition the user simulator on the target agent policy itself, learning $P(u_t \mid h_{t-1}, z, \pi)$. By embedding the target policy's parameters or system prompt into the user simulator during training, the generative model explicitly internalizes the policy dependence. This aligns the modeled belief updates with the evaluation policy π_e, empirically aligning the modeled belief updates between $M_t^{\pi_e}$ and $M_t^{\pi_b}$ and relying on the network's out-of-distribution generalization to mitigate the variance explosion.

5 Experimental Evaluation

To empirically validate our theoretical findings, we structure our evaluation into two distinct phases: (1) an *in-distribution* evaluation isolating the distortion caused by look-ahead bias (Theorem 1), and (2) an *out-of-distribution* evaluation isolating variance explosion under covariate shift (Theorem 2).

5.1 Experimental Setup

Datasets. We evaluate on two multi-turn datasets. First, we use *WildChat* [Zhao et al., 2024] (filtered to multi-turn conversations) to test general dynamics. We also introduce *ConvApparel-V2*,[3] an extension of ConvApparel [Meshi et al., 2026], introducing newly collected human interactions with *13 distinct assistant personas* for the footwear category, each governed by a system prompt reflecting specific styles (e.g., domain expert, efficient matchmaker, etc.). Details are in Appendices I.3 and J.

Control Variables and Annotation. To construct training controls, we use Gemini 3.1 Pro [Gemini Team Google, 2025] as an LLM annotator to label the offline datasets. We extract three types of controls based on full trajectories: *Persona+Goal* (extracting explicit linguistic rules and tasks), *Cognitive Profile* (extracting latent traits like system literacy), and *Scenario Generation* (a holistic summary of how the interaction progressed). To isolate causal bias from simple LLM-induced

[3]Link: `https://huggingface.co/datasets/google/ConvApparel`.

Table 1: **In-Distribution Generative Fidelity and Semantic Drift (WildChat).** Unconditioned models truncate conversations; globally controlled models artificially inflate turns and suffer semantic drift. Bold indicates the conditioned simulator closest to the human baseline. Full metrics are provided in Appendix E.

Simulator Type	Turn Count	Avg. Words / Turn	Task Unresolved	Iterative Refine.	Highly Detailed
Ground Truth Data	4.31 ± 0.01	113.3 ± 0.1	85.4%	48.4%	11.5%
Unconditioned SFT	3.55 ± 0.01	85.6 ± 0.2	91.1%	32.3%	11.3%
Prompted Simulator (Cognitive)	6.45 ± 0.01	88.3 ± 0.2	89.1%	19.2%	28.4%
Prompted Simulator (Scenario)	8.62 ± 0.02	68.5 ± 0.2	87.4%	17.8%	41.2%
Prompted (Scenario, Length-Constr.)	4.45 ± 0.01	92.2 ± 0.2	88.0%	19.5%	38.2%
Cognitive Profile Conditioned SFT	5.19 ± 0.01	95.1 ± 0.2	**83.8%**	29.1%	**14.1%**
Trajectory-Cond. SFT (Scenario)	7.37 ± 0.01	81.4 ± 0.1	83.2%	28.7%	24.5%
Dynamic State Conditioned SFT (Ours)	$\mathbf{4.78 \pm 0.01}$	$\mathbf{109.5 \pm 0.2}$	76.8%	**30.2%**	21.1%

verbosity (e.g., generating more turns simply because of a long prompt), we also create a *Length-Constrained Scenario* control that prompts the simulator to respect a strict length limit. For our dynamic mitigation, we annotate turns with *step-wise dynamic* controls representing turn-level emotional affect, cognitive load, and implicit intent for each sub-trajectory. Complete prompts used for all annotations are described in Appendix I.

Baselines. We consider the following simulation baselines: (1) *Unconditioned SFT:* A baseline simulator SFT-trained on user tokens with no controls; (2) *Prompted Simulator:* A zero-shot baseline where an off-the-shelf LLM is prompted with the global trajectory controls to act as the user; (3) *Rejection Sampling:* An inference-time baseline that generates N candidate trajectories from the Unconditioned SFT model and selects the one that best matches the target control (as determined by an LLM judge); (4) *Trajectory-conditioned SFT (standard practice):* A simulator SFT-trained to maximize the likelihood of user tokens conditioned on post hoc global controls (Persona+Goal, Scenario) prepended as a system prompt; (5) *Cognitive Profile SFT (mitigating Theorem 1):* Trajectory-conditioned SFT restricted to a cognitive profile constraint, attempting to avoid action-dependent look-ahead bias; (6) *Dynamic State SFT (mitigating Theorem 1):* Trained to predict the intermediate dynamic state annotation z_t immediately before generating the user text u_t. (7) *Agent-Aware SFT (mitigating Theorem 2):* To mitigate out-of-distribution controllability collapse, an SFT-trained model conditioned on both the user control and the target agent's system prompt.

Automated Evaluation Loop. To automate closed-loop offline testing, we use proxy SFT agents. For WildChat, we train an unconditioned agent on the dataset's empirical assistant responses. For ConvApparel, we train an agent conditioned on distinct assistant prompts, parameterizing the policy shift π_e by swapping prompts at inference time.

Evaluation Metrics. We evaluate simulation quality across three dimensions to understand the tradeoffs involved in controllable simulation: (1) *Distributional Fidelity & Semantic Drift*: Does the simulator behave like a natural human population, or do the controls induce unnatural conversations? We assess both deterministic statistics (turn counts, word lengths, etc.) and semantic intent; (2) *Control Adherence*: Does the simulator successfully reflect the requested controls? We measure this via an independent LLM-as-a-judge (scoring Persona and Goal match) as well as Gemini text-embedding cosine-similarity; (3) *Behavioral Diversity*: Do the generated responses exhibit natural human variance, or do they collapse into repetitive patterns? We quantify this by measuring Shannon entropy of the discrete adherence scores (evaluated by the LLM judge on a categorical scale).

5.2 In-Distribution: Testing Look-Ahead Bias and Adherence

We first assess whether models adhere to controls without distorting the natural user behavior distribution, evaluating Theorem 1 on the WildChat dataset (in-distribution results for ConvApparel exhibit the same trends and are described in Appendix G). We present a condensed summary of key linguistic and behavioral metrics in Table 1; complete tables with all metrics, adherence scores, and entropies are provided in Appendix E.

Length Inflation and Semantic Drift. Standard trajectory-conditioned SFT models structurally overcorrect due to look-ahead bias. Because the label drives toward an outcome at the end of the offline trajectory, the simulator artificially prolongs interactions to ensure constraints are met. For example, Scenario-Conditioned SFT averages 7.37 turns, and the Prompted version hits 8.62, compared to

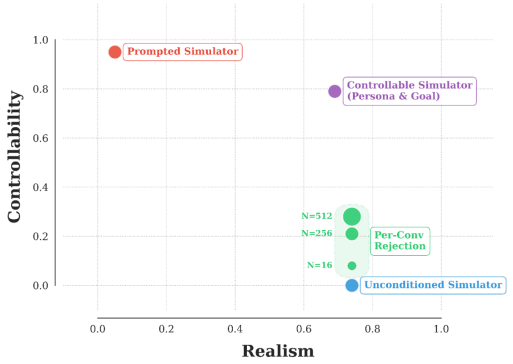


Figure 2: **Realism-Controllability Trade-off.** Explicitly prompted models achieve high adherence but suffer severe diversity collapse. Best-of-N sampling fails completely. Causally conditioned models offer a superior Pareto frontier.

Table 2: **Control Adherence Scores.** Mean ± 95% CI via LLM Judge and embedding cosine-similarity.

Simulator Type	Persona Match	Embed. Sim.
Ground Truth (Persona+Goal)	0.89 ± 0.01	0.87 ± 0.01
Prompted Sim (Persona+Goal)	0.96 ± 0.01	0.93 ± 0.01
Traj-Cond. SFT (Persona+Goal)	0.77 ± 0.01	0.69 ± 0.01
Ground Truth (Scenario)	0.92 ± 0.01	0.88 ± 0.01
Prompted Sim (Scenario)	0.85 ± 0.01	0.82 ± 0.01
Traj-Cond. SFT (Scenario)	0.82 ± 0.01	0.69 ± 0.01
Ground Truth (Cognitive)	0.72 ± 0.01	0.70 ± 0.01
Prompted Sim (Cognitive)	0.90 ± 0.01	0.88 ± 0.01
Cognitive Profile SFT	0.56 ± 0.01	0.52 ± 0.01

the human baseline of 4.31. This global bias triggers severe semantic drift. Real users construct "Highly Detailed" prompts 11.5% of the time, whereas standard trajectory-conditioned models double this rate ($\sim$24.5%) by forcibly injecting constraints upfront. Furthermore, they drastically under-use Iterative Refinement (28.7% vs. human 48.4%, Table 1). This suggests that the simulator abandons natural user behavior in favor of rigid, transactional compliance simply to satisfy the global control.

We verify this to be a structural causal violation and not mere LLM verbosity. When our *Length-Constrained Scenario* baseline explicitly forces the global prompt to respect a 4-turn limit, it successfully caps the turn count at 4.45. However, semantic drift remains severe (38.2% Highly Detailed prompts and 19.5% Iterative Refinement). Conversely, our *Dynamic State SFT* simulator reduces these distortions. By generating states purely autoregressively, it prioritizes step-wise coherence and does not artificially induce long-horizon global outcomes. This represents a fundamental trade-off: bypassing post hoc labels eliminates semantic drift and length inflation, but at the expense of strict global controllability (unless using explicitly parameterized state dynamics, see Appendix A).

Constraint Adherence and Over-Compliance. Agent evaluators need simulators to faithfully respect targeted constraints. As seen in Table 2, explicitly prompted simulators achieve high raw adherence scores (0.90 Persona Match on cognitive profiles vs. 0.72 for human ground truth). Rather than indicating superior realism, scoring above the human baseline reflects *artificial over-compliance*: human behavior is naturally varied, whereas prompted LLMs force generation to follow rigid, stereotypical paths to satisfy instructions. To verify that this is not a "self-preference" artifact of the Gemini judge, we replicate the evaluation with an independent GPT-5.6 judge (Appendix L), which confirms the same over-adherence phenomenon (0.86 prompted vs. 0.58 human on cognitive profiles). Our trajectory-conditioned SFT models show reasonable adherence for explicit tasks (0.82 Persona Match for Scenarios), while text-based SFT on abstract cognitive profiles displays lower adherence (0.56). Finally, because purely autoregressive *Dynamic State SFT* operates without a static trajectory target, it is excluded from Table 2; we evaluate targeted dynamic reactivity via our bipartite framework in Appendix A, and provide side-by-side qualitative dialogue transcripts in Appendix M.

Behavioral Diversity Collapse. While global controls can drive high adherence, they fundamentally distort natural behavioral diversity. We quantify this by measuring the Shannon entropy over the discrete adherence scores (see full results in Appendix E and table 10). Explicitly prompted (non-SFT-trained) simulators severely compress variance of real human behavior to adhering to complex controls like Persona+Goal (driving entropy down from the human baseline of 1.04 to 0.728) and Cognitive Profiles (from 1.10 down to 0.636). The models collapse to a "safe" generative path to satisfy the prompt, overriding the natural diversity of human behavior. Conversely, trajectory-conditioned SFT models artificially inflate this variance (entropies ranging from 1.28 to 1.64), but do so at the cost of structural length inflation and semantic drift as discussed above.

Attempting to avoid this adherence-diversity trade-off via inference-time *Rejection Sampling* fails completely (see Figure 2). Searching the vast combinatorial space of multi-turn dialogue is computa-

Table 3: **Condensed Counterfactual Stability under Covariate Shift.** Evaluated zero-shot on unseen agent policies. The standard Static Agnostic approach collapses when facing the terse "Efficient Matchmaker". (Full comprehensive metrics for all four held-out agents are provided in Appendix G).

Held-Out Evaluation Agent	Simulator Paradigm	Linguistic Statistics		Behavioral Intent (%)	
		Avg. Words per Turn	Total User Words	Negative Sentiment	Iterative Refinement
Domain Expert (Academic, Verbose)	Ground Truth Data	12.0	38.7	13.4	31.7
	Static, Agnostic (Standard)	8.8	28.0	18.0	12.5
	Static, Aware	10.5	32.0	15.1	25.0
	Dynamic, Agnostic	11.2	35.1	14.5	22.3
	Dynamic, Aware (Ours)	**12.1**	**38.5**	**13.6**	**31.2**
Efficient Matchmaker (Ultra-terse, Fast)	Ground Truth Data	11.7	39.7	11.9	41.3
	Static, Agnostic (Standard)	25.4	92.1	26.8	8.8
	Static, Aware	13.2	43.5	16.4	32.6
	Dynamic, Agnostic	14.1	45.2	18.2	24.8
	Dynamic, Aware (Ours)	**11.8**	**40.1**	**12.2**	**40.9**

Table 4: **Empirical Validation of Variance Explosion (Theorem 2).** Normalized variance ratio $\text{Var}_{\text{sim}}/\text{Var}_{\text{GT}}$ of Total User Words, stratified by conversation horizon T across all four held-out ConvApparel agents. Ground truth variance is computed from real human data. A ratio of 1.0 indicates perfect alignment with real human variance.

Simulator Paradigm	$T=2$	$T=3$	$T=4$	$T=5$	$T\geq 6^*$	Avg. Step-wise Growth ($\hat{\eta}$)
Ground Truth Data	1.00	1.00	1.00	1.00	1.00	—
Static Global, Agent-Agnostic	1.76	2.36	3.52	4.89	7.19	$\hat{\eta} \approx 0.42$
Static Global, Agent-Aware	1.28	1.34	1.35	1.81	2.14	$\hat{\eta} \approx 0.14$
Dynamic State, Agent-Agnostic	1.14	1.15	1.29	1.35	1.44	$\hat{\eta} \approx 0.06$
Dynamic State, Agent-Aware (Ours)	**1.03**	**0.99**	**1.05**	**1.02**	**1.07**	$\hat{\eta} \approx 0.01$

Reference: Var_{GT}(Total Words): $T=2$: 185; $T=3$: 419; $T=4$: 606; $T=5$: 975; $T\geq 6^*$: 4,835

* *Note:* The $T \geq 6$ column aggregates the long-tail of conversations. We treat this conceptually as $T = 6$ for growth estimates.

tionally intractable, yielding near-zero adherence even with massive compute overhead ($N = 512$ candidates per conversation); see further results in Appendix F.

5.3 Out-of-Distribution: Testing Controllability Collapse

To test whether simulators suffer from controllability collapse under covariate shift when $\pi_e \neq \pi_b$, we construct a train/test split utilizing the 13 footwear agents. We train our controllable user simulator using SFT on interactions with the nine in-distribution agents, and evaluate them in zero-shot fashion in closed-loop conversations with the four held-out agents/personas.

Theorem 2 bounds the variance of the trajectory density ratio (the importance sampling weights). As a proxy to assess this breakdown empirically, we measure the variance of an additive trajectory feature: *total user words*. In an uncorrected simulator, the compounding instability of step-wise density ratios manifests as extreme trajectory-level fluctuations in such additive metrics. Table 3 and Table 4 confirm this breakdown: these results expose both severe mean distortion and compounding variance instability.[4] When standard trajectory-conditioned SFT (Static, Agent-Agnostic) encounters the terse "Efficient Matchmaker," it inflates word counts (25.4 vs. 11.7 human) and negative sentiment (26.8% vs. 11.9% human). Conversely, with the verbose "Domain Expert," it unnaturally compresses responses. Since the policy-agnostic simulator produces heterogeneous mean errors across different policies, this failure compounds multiplicatively over time. As a result, its aggregate normalized variance ($\text{Var}_{\text{sim}}/\text{Var}_{\text{GT}}$) explodes, growing by an average factor of $\sim 1.42\times$ *at each turn* ($\hat{\eta} \approx 0.42$).

[4]While word count is an additive feature bounded by maximum length, its rapid step-wise variance growth serves as a strong empirical proxy for the density ratio's theoretical collapse. We provide additional direct empirical validation of the true multiplicative density ratio collapse using offline classifiers in Appendix G.2.

Explicitly resolving these structural discrepancies restores stability. Employing the *Dynamic State* formulation avoids look-ahead bias by adapting only to the step-wise observable history, while *Agent-Aware SFT* directly resolves any policy-induced mismatched expectations. The combined *Dynamic State, Agent-Aware* approach achieves near-perfect human parity (11.8 words/turn), mirrors human subtle behavioral adaptations to new styles (e.g., matching negative sentiment at 12.2% with the Matchmaker) and maintains a highly stable variance ratio across conversation horizons ($\hat{\eta} \approx 0.01$).

While total user words serves as a clear downstream proxy for this failure, Theorem 2 predicts the geometric explosion of the underlying mathematical trajectory density ratio. We provide a direct empirical validation of this probabilistic collapse (evaluating $P(z \mid h_t)$ via offline classifiers) in Appendix G.2 (Table 19), alongside granular turn-by-turn negative emotion incidence and confidence intervals for long-tail horizons ($T \geq 6$) in Appendix G. Furthermore, to ensure our findings are not specific to the Gemini model family or clean synthetic annotations, we replicate both the WildChat and ConvApparel density-ratio experiments using open-weight **Qwen3** simulators and annotators (Appendix L), and demonstrate via synthetic label-corruption ablations that dynamic state training remains highly stable even under 30% annotator label noise (Appendix G.3).

Policy Visibility Requirements. Notably, direct policy conditioning (Agent-Aware SFT) requires only the target agent's system prompt or high-level style directives, not model weights or internal activations. For black-box commercial APIs or multi-stage retrieval-augmented (RAG) pipelines where prompts are hidden, evaluators can summarize observed agent behavior offline into a lightweight "policy card" prompt. Crucially, when zero visibility into the agent policy is desired, step-wise dynamic control natively bypasses policy conditioning altogether: by conditioning strictly on the observable filtration $\mathcal{F}_{t-}$, it holds variance bounded (1.44 at $T \geq 6$ vs. 7.19 for standard SFT) with zero knowledge of the agent's architecture.

6 Conclusion

Reliable offline evaluation requires simulating targeted counterfactuals without deviating from natural user distributions. In this work, we show that standard SFT simulators trained on static, post hoc trajectory labels inject *look-ahead bias* by violating the natural filtration of the interaction. Under policy shift, this entanglement causes distributional distortions, behavioral diversity collapse, and geometric variance explosion (*controllability collapse*). Our proposed causally grounded mitigations empirically eliminate this bias, preserve natural human variance, and enable robust zero-shot generalization to unseen agent policies across distinct model families.

More broadly, our analysis highlights a fundamental trade-off between outcome controllability and causal realism. Evaluators cannot force future-dependent outcomes (e.g., guaranteeing that a user ends a conversation frustrated) without leaking behavior-policy artifacts into user generation and blinding the evaluation to genuine agent improvements. Instead, causally sound controllability requires reframing user simulation from specifying terminal outcomes to parameterizing pre-interaction states and dynamic reaction rules, as formalized in our bipartite state-dynamics framework. Beyond offline benchmarking, respecting this filtration boundary is equally critical when simulators serve as interactive environments for online reinforcement learning or automated red-teaming, where look-ahead bias would otherwise incentivize agents to exploit spurious conditioning artifacts rather than genuine user dynamics.

While these mitigations provide a rigorous foundation, several avenues for future work remain. Adapting direct policy conditioning for black-box environments (e.g., by inferring compact "policy cards" from interaction probes of proprietary APIs or complex RAG pipelines) is a compelling next step. Furthermore, operationalizing our bipartite state-dynamics architecture at scale transforms scenario design into an inverse problem: discovering pre-interaction profiles (z_0, c) that naturally elicit rare or safety-critical behaviors under a target policy. Ultimately, integrating representation learning and reinforcement learning to automate the search over these dynamic control spaces could allow evaluators to steer complex, long-horizon user behaviors while preserving causal fidelity.

References

David Brandfonbrener, Alberto Bietti, Jacob Buckman, Romain Laroche, and Joan Bruna. When does return-conditioned supervised learning work for offline reinforcement learning? *Advances in Neural Information Processing Systems*, 35:1542–1553, 2022.

Sam Davidson, Salvatore Romeo, Raphael Shu, James Gung, Arshit Gupta, Saab Mansour, and Yi Zhang. User simulation with large language models for evaluating task-oriented dialogue. *arXiv preprint arXiv:2309.13233*, 2023.

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 Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng, editors, *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 35212–35290, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.1786. URL https://aclanthology.org/2025.emnlp-main.1786/.

Gemini Team Google. Gemini 3 pro model card. December 2025.

Chih-Wei Hsu, Martin Mladenov, Ofer Meshi, James Pine, Hubert Pham, Shane Li, Xujian Liang, Anton Polishko, Li Yang, Ben Scheetz, et al. Minimizing live experiments in recommender systems: User simulation to evaluate preference elicitation policies. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2925–2929, 2024.

Bingrui Jin, Kunyao Lan, and Mengyue Wu. Twice: An llm agent framework for simulating personalized user tweeting behavior with long-term temporal features. *arXiv preprint arXiv:2602.22222*, 2025.

Olav Kallenberg. *Foundations of modern probability*. Springer, 1997.

Philippe Laban, Hiroaki Hayashi, Yingbo Zhou, and Jennifer Neville. LLMs get lost in multi-turn conversation. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=VKGTGGcwl6.

I-han Lai, Stefan Wager, et al. Estimating dynamic marginal policy effects under sequential unconfoundedness. *arXiv preprint arXiv:2604.05639*, 2026.

Qiang Liu, Lihong Li, Ziyang Tang, and Dengyong Zhou. Breaking the curse of horizon: Infinite-horizon off-policy estimation. *Advances in neural information processing systems*, 31, 2018.

Yao Liu, Pierre-Luc Bacon, and Emma Brunskill. Understanding the curse of horizon in off-policy evaluation via conditional importance sampling. In *International Conference on Machine Learning*, pages 6184–6193. PMLR, 2020.

Zihao Liu, Hantao Zhou, Jiguo Li, Jun Xu, Jiuchong Gao, Jinghua Hao, Renqing He, and Peng Wang. Muse: Multi-domain chinese user simulation via self-evolving profiles and rubric-guided alignment. *arXiv preprint arXiv:2604.13828*, 2026.

Ofer Meshi, Krisztian Balog, Sally Goldman, Avi Caciularu, Guy Tennenholtz, Jihwan Jeong, Amir Globerson, and Craig Boutilier. Convapparel: A benchmark dataset and validation framework for user simulators in conversational recommenders. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5270–5304, 2026.

Tarek Naous, Philippe Laban, Wei Xu, and Jennifer Neville. Flipping the dialogue: Training and evaluating user language models. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=ykSmkVqzn4.

Keiran Paster, Sheila McIlraith, and Jimmy Ba. You can't count on luck: Why decision transformers and rvs fail in stochastic environments. *Advances in neural information processing systems*, 35: 38966–38979, 2022.

Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

Tian Qin, Felix Bai, Ting-Yao Hu, Raviteja Vemulapalli, Hema Swetha Koppula, Zhiyang Xu, Bowen Jin, Mert Cemri, Jiarui Lu, Zirui Wang, et al. Compass: A multi-turn benchmark for tool-mediated planning & preference optimization. *arXiv preprint arXiv:2510.07043*, 2025.

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 (Volume 1: Long Papers)*, pages 21082–21107, 2025.

Xiaolei Wang, Xinyu Tang, Xin Zhao, Jingyuan Wang, and Ji-Rong Wen. Rethinking the evaluation for conversational recommendation in the era of large language models. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10052–10065, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.621. URL `https://aclanthology.org/2023.emnlp-main.621/`.

Yuanfu Wang, Chao Yang, Ying Wen, Yu Liu, and Yu Qiao. Critic-guided decision transformer for offline reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 15706–15714, 2024.

Gangyi Zhang, Chongming Gao, Hang Pan, Runzhe Teng, and Ruizhe Li. Reformulating conversational recommender systems as tri-phase offline policy learning. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pages 3135–3144, 2024.

Wenting Zhao, Xiang Ren, Jack Hessel, Claire Cardie, Yejin Choi, and Yuntian Deng. Wildchat: 1m chatgpt interaction logs in the wild. *arXiv preprint arXiv:2405.01470*, 2024.

A Discussion: Enforcing Global Constraints via Parameterized State Dynamics

In the main text, we established that step-wise dynamic control elegantly resolves the mathematical pitfalls of off-policy evaluation by generating states conditionally on the observable history. However, predicting the next local state purely autoregressively leaves open the question of how to exert targeted, global influence over the trajectory. If an evaluator wishes to simulate a user whose emotional state degrades predictably when an agent is overly repetitive, standard step-wise generation lacks an explicit mechanism to enforce this trajectory-level constraint without reverting to flawed post hoc labels. To enforce global behavioral shapes while strictly avoiding future-leaking outcome conditioning, we propose a bipartite architectural framework that explicitly parameterizes the transition dynamics.

This framework is realized by decoupling the user simulator into two distinct generative components: a state dynamics model and a user response model. Instead of conditioning generation on an eventual trajectory outcome, we define two control variables that are strictly adapted to the initial filtration. The first is an initial state variable z_0, which anchors the user's starting intent or emotional baseline. The second is a global dynamics control parameter c. This parameter acts as an overarching profile or an embedded set of transition rules that governs the mechanics of how the user's internal state evolves in response to the agent. Because both z_0 and c are determined strictly prior to the interaction, they possess no backward causal links to the agent's future actions.

The sequential generative process operates chronologically. At any time step $t \geq 1$, the user possesses a latent internal state z_t. This state is updated from the previous state by processing the observable history up to the previous turn and the structural laws dictated by the profile, formalized as the state dynamics model $P_{\text{dyn}}(z_t \mid h_{t-1}, z_{t-1}, c)$. Following this internal update, the user response model samples the conversational utterance conditionally based on the observable history and this current state, formalized as $P_{\text{resp}}(u_t \mid h_{t-1}, z_t)$. Finally, the target agent observes the history and selects an action according to its policy $\pi(a_t \mid h_{t-1}, u_t)$, producing the updated history $h_t = (h_{t-1}, u_t, a_t)$ and completing the cycle.

Through this decoupled architecture, evaluators control long-horizon behavior implicitly. Rather than prompting a simulator to end the conversation in frustration, an evaluator supplies a dynamics profile c dictating that repeated agent misunderstandings deterministically degrade the user's patience state z_t, which the response model subsequently translates into hostile text.

Below, we show that this architecture guarantees complete causal consistency and statistical stability under arbitrary off-policy covariate shift.

Theorem 3 (Causal Consistency and Stability of Parameterized State Dynamics)**.** *Let a user simulation process be parameterized by strictly $\mathcal{F}_0$-measurable variables z_0 and c, alongside a state transition kernel $P_{dyn}(z_t \mid h_{t-1}, z_{t-1}, c)$ and a response kernel $P_{resp}(u_t \mid h_{t-1}, z_t)$. Assuming an unbiased simulator ($P_{sim} = P^{\pi_b}$), for any arbitrary evaluation policy π_e and behavior policy π_b, the look-ahead bias evaluates to exactly one, and the step-wise counterfactual user density ratio $\rho_t(u_t \mid h_{t-1}, z_0, c)$ evaluates to exactly one almost surely. Consequently, the local label sensitivity V_t equates to zero, completely preventing geometric variance explosion.*

Proof. We first address the look-ahead bias. By Theorem 1, look-ahead bias is quantified by the ratio of the conditional probability of the control variables given the pre-action history to their probability given the post-action history. Because the controls (z_0, c) are defined entirely prior to the conversation, they act as fixed causal ancestors. The agent selects its action a_t according to a policy $\pi_b(a_t \mid h_{t-1}, u_t)$ that is solely a function of the observable history. Consequently, the unobserved global controls are conditionally independent of the action a_t given the pre-action history, yielding $(z_0, c) \perp\!\!\!\perp a_t \mid h_{t-1}, u_t$. Applying Bayes' theorem to the post-action probability expands the denominator into a fraction where the likelihood term $\pi_b(a_t \mid h_{t-1}, u_t, z_0, c)$ simplifies directly to $\pi_b(a_t \mid h_{t-1}, u_t)$. Because this term is independent of the unobserved controls, it factors entirely out of the marginalization integral over the controls and perfectly cancels with the identical policy term in the numerator. The posterior probability thus mathematically collapses to the prior probability $P^{\pi_b}(z_0, c \mid h_{t-1}, u_t)$, forcing the bias ratio to equal one and structurally preventing any causal leakage from future events.

We next address controllability collapse by analyzing the user generative error $\rho_t(u_t \mid h_{t-1}, z_0, c)$, which fundamentally drives the variance explosion in Theorem 2. We evaluate the conditional

marginal probability of a user utterance u_t given the observable history and the global controls under an arbitrary agent policy π, denoted as $P^\pi(u_t \mid h_{t-1}, z_0, c)$. By the definition of conditional probability, this is the ratio of the joint density of the utterance and the history over the marginal density of the history. To obtain these quantities, we integrate out the unobserved latent state trajectory $z_{1:t}$. Applying the chain rule of probability to the causal sequence of the environment, the marginal probability of the history up to step $t-1$ expands as the integral over the latent path $z_{1:t-1}$:

$$P^\pi(h_{t-1} \mid z_0, c) = \int \prod_{k=1}^{t-1} \Big[P_{\text{resp}}(u_k \mid h_{k-1}, z_k) P_{\text{dyn}}(z_k \mid h_{k-1}, z_{k-1}, c) \pi(a_k \mid h_{k-1}, u_k) \Big] \, dz_{1:t-1}. \tag{3}$$

Because we condition our generative step on a strictly observed history sequence h_{t-1}, the specific sequence of past agent actions and user utterances are fixed constants in this context. Thus, the product of the agent's action probabilities contains no variables dependent on the integration variables $z_{1:t-1}$. This allows the entire policy product term to factor completely outside of the integral:

$$P^\pi(h_{t-1} \mid z_0, c) = \left(\prod_{k=1}^{t-1} \pi(a_k \mid h_{k-1}, u_k) \right) \int \prod_{k=1}^{t-1} P_{\text{resp}}(u_k \mid h_{k-1}, z_k) P_{\text{dyn}}(z_k \mid h_{k-1}, z_{k-1}, c) \, dz_{1:t-1}. \tag{4}$$

We apply this exact factorization to the numerator to calculate the joint density $P^\pi(u_t, h_{t-1} \mid z_0, c)$. The term representing the agent's past choices factors out of the integration over $z_{1:t}$ in an identical manner:

$$\begin{aligned} P^\pi(u_t, h_{t-1} \mid z_0, c) = \left(\prod_{k=1}^{t-1} \pi(a_k \mid h_{k-1}, u_k) \right) \int & P_{\text{resp}}(u_t \mid h_{t-1}, z_t) P_{\text{dyn}}(z_t \mid h_{t-1}, z_{t-1}, c) \\ & \times \prod_{k=1}^{t-1} P_{\text{resp}}(u_k \mid h_{k-1}, z_k) P_{\text{dyn}}(z_k \mid h_{k-1}, z_{k-1}, c) \, dz_{1:t}. \end{aligned} \tag{5}$$

Dividing the numerator by the denominator isolates $P^\pi(u_t \mid h_{t-1}, z_0, c)$. The factored policy product term appears identically in both the top and bottom of the fraction, completely canceling out. This analytical cancellation demonstrates that the true, marginalized probability of the user's action u_t is purely a function of the structural transition kernels and is strictly independent of the agent policy π. Therefore, evaluating the true user dynamics under the evaluation policy π_e yields the exact same distribution as under the behavior policy π_b. Assuming an unbiased simulator that successfully captures this causal factorization, the step-wise density ratio $\rho_t(u_t \mid h_{t-1}, z_0, c)$ becomes perfectly deterministic and identically equal to one for all possible utterances u_t. Consequently, its variance across the step-wise generation, defined as the local label sensitivity V_t, evaluates to exactly zero. Substituting $V_t = 0$ into the recursive expectation bound established in Theorem 2 ensures that the total counterfactual variance structurally evaluates to zero, completing the proof. □

This bipartite dynamic framework fundamentally transforms controllable simulation from an outcome-generation paradigm into an inverse design problem. Evaluators cannot simply pass a target outcome to the model during inference and expect a causally sound interaction. Instead, creating targeted counterfactual datasets requires discovering the optimal combinations of initial states z_0 and dynamics profiles c that naturally induce the desired outcomes when simulated against specific target policies. This opens a critical and highly complex avenue for future work in prompt optimization and representation learning. We envision future research exploring reinforcement learning applied to the state dynamics model's meta-parameters, continuous embedding optimization, or gradient-based discrete search to automatically discover the optimal dynamics control profiles. By shifting the objective from directly generating text that matches an outcome to optimizing the rules of an environment such that the desired outcome emerges naturally, the field can generate highly specific, stress-tested conversational datasets that strictly adhere to causal mathematics.

A.1 Empirical Demonstration: De-Escalation Benchmarking

To empirically demonstrate how the bipartite framework resolves the tension between targeted stress-testing and causal validity, we evaluate an agent de-escalation scenario across the four held-out ConvApparel agents. Rather than conditioning on a future-leaking outcome label ($\hat{z}$ = "ends frustrated"),

the evaluator specifies a pre-interaction starting state $z_0 \in \mathcal{F}_0$ (baseline mood) and a global dynamics reactivity profile $c \in \mathcal{F}_0$ (either an *Average Reactivity* profile or a *Low-Patience* profile where repeated agent misunderstandings or verbosity rapidly degrade the user's patience state z_t). We conduct 500 closed-loop rollouts per agent and measure the percentage of conversations in which user frustration emerges.

Table 5: **De-Escalation Benchmark (Frustration Emergence Rate % $\pm$ 95% CI).** Outcome-conditioned SFT forces high frustration regardless of agent quality. Under our bipartite framework (c = Low-Patience), frustration emerges dynamically as a true function of the target agent's de-escalation capability.

Held-Out Agent	Human Ground Truth	Traj-Cond. ($\hat{z}$ = "ends frustrated")	Bipartite (c = Avg. Reactivity)	Bipartite (c = Low-Patience)
Domain Expert (Verbose, Dry)	$13.4 \pm 3.7\%$	$77.4 \pm 3.7\%$	$14.8 \pm 3.1\%$	$38.6 \pm 4.3\%$
Empathetic Listener (Validating)	$8.1 \pm 3.0\%$	$78.0 \pm 3.6\%$	$9.8 \pm 2.6\%$	$9.8 \pm 2.6\%$
Efficient Matchmaker (Ultra-terse)	$11.9 \pm 3.6\%$	$75.6 \pm 3.8\%$	$16.6 \pm 3.3\%$	$32.4 \pm 4.1\%$
Enthusiastic Rambler (Distracted)	$8.9 \pm 3.2\%$	$75.2 \pm 3.8\%$	$11.0 \pm 2.7\%$	$14.6 \pm 3.1\%$

As shown in Table 5, direct outcome conditioning ($\hat{z}$ = "ends frustrated") suffers from severe level saturation: it forces user frustration at roughly $6\times$ the human rate ($\sim$ 75–78%) uniformly across all agents, completely blinding the evaluator to differences in agent quality. Conversely, under the bipartite framework with a *Low-Patience* profile, an easily frustrated user reaches 38.6% frustration against the dry, verbose *Domain Expert* and 32.4% against the abrupt *Efficient Matchmaker*, but stays at 9.8% when interacting with the validating *Empathetic Listener*. While causally valid simulation does not rigidly guarantee frustration upfront, conditioning on (z_0, c) enables targeted stress-testing where emotional outcomes emerge authentically in response to agent behavior.

B Proof of Divergence in Trajectory-Labeled User Simulators

We analyze the explicit density ratio between the true trajectory distribution $P^\pi(\tau)$ under an arbitrary policy π and the distribution $P^\pi_{\text{sim}}(\tau)$ generated by a causally trained controllable simulator. The learned simulator dynamics converge to the empirical conditional distribution: $P_{\text{sim}}(u_t \mid h_{t-1}, \hat{z}) \triangleq P^{\pi_b}(u_t \mid h_{t-1}, \hat{z})$.

Using Bayes' theorem, we expand this conditional density. Because the user's base unconditional action u_t depends only on the natural filtration $\mathcal{F}_{t-1}$ and not on the future behavior policy, $P^{\pi_b}(u_t \mid h_{t-1})$ reduces to $P(u_t \mid h_{t-1})$. Thus:

$$P_{\text{sim}}(u_t \mid h_{t-1}, \hat{z}) = P(u_t \mid h_{t-1}) \frac{P^{\pi_b}(\hat{z} \mid h_{t-1}, u_t)}{P^{\pi_b}(\hat{z} \mid h_{t-1})}$$

During evaluation, the simulator's induced probability density of a trajectory τ under π is given by marginalizing over the control space:

$$P^\pi_{\text{sim}}(\tau) = \int_{\mathcal{Z}} P^{\pi_b}(\hat{z} \mid h_0) \left[\prod_{t=1}^{T} P_{\text{sim}}(u_t \mid h_{t-1}, \hat{z}) \pi(a_t \mid h_{t-1}, u_t) \right] d\hat{z}$$

Substituting the Bayesian expansion into this integral:

$$P^\pi_{\text{sim}}(\tau) = \int_{\mathcal{Z}} P^{\pi_b}(\hat{z} \mid h_0) \left[\prod_{t=1}^{T} P(u_t \mid h_{t-1}) \pi(a_t \mid h_{t-1}, u_t) \frac{P^{\pi_b}(\hat{z} \mid h_{t-1}, u_t)}{P^{\pi_b}(\hat{z} \mid h_{t-1})} \right] d\hat{z}$$

Recognizing the true trajectory density $P^\pi(\tau) = \prod_{t=1}^{T} P(u_t \mid h_{t-1}) \pi(a_t \mid h_{t-1}, u_t)$, we pull it out:

$$P^\pi_{\text{sim}}(\tau) = P^\pi(\tau) \int_{\mathcal{Z}} P^{\pi_b}(\hat{z} \mid h_0) \left[\prod_{t=1}^{T} \frac{P^{\pi_b}(\hat{z} \mid h_{t-1}, u_t)}{P^{\pi_b}(\hat{z} \mid h_{t-1})} \right] d\hat{z}$$

We expand the product. The state history updates as $h_t = (h_{t-1}, u_t, a_t)$. Thus, the denominator at step $t+1$, $P^{\pi_b}(\hat{z} \mid h_t)$, is equivalently $P^{\pi_b}(\hat{z} \mid h_{t-1}, u_t, a_t)$. The learned prior $P^{\pi_b}(\hat{z} \mid h_0)$

cancels the denominator of the very first step. To align the sequence, we multiply and divide by $P^{\pi_b}(\hat{z} \mid h_{T-1}, u_T, a_T)$, which is exactly $P_L(\hat{z} \mid \tau)$:

$$P^{\pi}_{\text{sim}}(\tau) = P^{\pi}(\tau) \int_{\mathcal{Z}} P_L(\hat{z} \mid \tau) \left[\prod_{t=1}^{T} \frac{P^{\pi_b}(\hat{z} \mid h_{t-1}, u_t)}{P^{\pi_b}(\hat{z} \mid h_{t-1}, u_t, a_t)} \right] d\hat{z}$$

Dividing both sides by $P^{\pi}(\tau)$ yields the explicit density ratio, confirming the compounding look-ahead bias due to the violation of the natural filtration. □

C Proof of OPE Controllability Collapse

We provide the proof for Theorem 2. Let $P^{\pi_e}(\tau \mid \hat{z})$ be the true conditional path distribution. Let $P^{\pi_e}_{\text{sim}}(\tau \mid \hat{z})$ be the sampling path distribution induced by the simulator $P_{\text{sim}}(u_t \mid h_{t-1}, \hat{z})$ and π_e.

We define the isolated user sampling density ratio process $W^{(u)}_t(\hat{z}) \triangleq \prod_{k=1}^{t} \rho_k(u_k \mid h_{k-1}, \hat{z})$. By expanding using Bayes' rule, the base unconditional user dynamics $P(u_k \mid h_{k-1})$ cancel exactly:

$$\rho_k(u_k \mid h_{k-1}, \hat{z}) = \frac{P^{\pi_e}(u_k \mid h_{k-1}, \hat{z})}{P_{\text{sim}}(u_k \mid h_{k-1}, \hat{z})} = \frac{P^{\pi_e}(\hat{z} \mid h_{k-1}, u_k)/P^{\pi_e}(\hat{z} \mid h_{k-1})}{P^{\pi_b}(\hat{z} \mid h_{k-1}, u_k)/P^{\pi_b}(\hat{z} \mid h_{k-1})} = \frac{M^{\pi_e}_k(u_k)}{M^{\pi_b}_k(u_k)}$$

First, we verify that $W^{(u)}(\hat{z})$ is a martingale adapted to the filtration $\mathcal{F}_t$ under the conditional sampling measure $P^{\pi_e}_{\text{sim}}(\cdot \mid \hat{z})$.

$$\begin{aligned}
\mathbb{E}_{\tau \sim P^{\pi_e}_{\text{sim}}(\cdot \mid \hat{z})}[W^{(u)}_t(\hat{z}) \mid \mathcal{F}_{t-1}] &= W^{(u)}_{t-1}(\hat{z}) \mathbb{E}_{u_t \sim P_{\text{sim}}(\cdot \mid h_{t-1}, \hat{z})} \left[\frac{P^{\pi_e}(u_t \mid h_{t-1}, \hat{z})}{P_{\text{sim}}(u_t \mid h_{t-1}, \hat{z})} \right] \\
&= W^{(u)}_{t-1}(\hat{z}) \sum_{u_t} P^{\pi_e}(u_t \mid h_{t-1}, \hat{z}) = W^{(u)}_{t-1}(\hat{z})
\end{aligned}$$

To bound the total variance $\mathbb{E}[W^{(u)}_T(\hat{z})^2] - 1$, we recursively evaluate the conditional second moment. Using $\mathbb{E}[X^2] = \text{Var}(X) + (\mathbb{E}[X])^2$:

$$\mathbb{E}_{\tau \sim P^{\pi_e}_{\text{sim}}(\cdot \mid \hat{z})}[W^{(u)}_t(\hat{z})^2 \mid \mathcal{F}_{t-1}] = W^{(u)}_{t-1}(\hat{z})^2 \left(1 + V_t(h_{t-1}, \hat{z}, \pi_e)\right)$$

By the Law of Total Expectation, distributing the weight yields:

$$\mathbb{E}_{P^{\pi_e}_{\text{sim}}}[W^{(u)}_t(\hat{z})^2] = \mathbb{E}_{P^{\pi_e}_{\text{sim}}}[W^{(u)}_{t-1}(\hat{z})^2] + \mathbb{E}_{h_{t-1}} \left[W^{(u)}_{t-1}(\hat{z})^2 V_t(h_{t-1}, \hat{z}, \pi_e) \right]$$

Unrolling this telescoping sum from $t = 1$ to T ($W^{(u)}_0(\hat{z}) = 1$) gives the algebraic expansion:

$$\mathbb{E}_{P^{\pi_e}_{\text{sim}}}[W^{(u)}_T(\hat{z})^2] = 1 + \sum_{t=1}^{T} \mathbb{E}_{h_{t-1}} \left[W^{(u)}_{t-1}(\hat{z})^2 V_t(h_{t-1}, \hat{z}, \pi_e) \right]$$

Assuming $V_t \geq \eta > 0$, substituting directly into the recursive expectation yields:

$$\mathbb{E}_{P^{\pi_e}_{\text{sim}}}[W^{(u)}_t(\hat{z})^2] \geq \mathbb{E}_{P^{\pi_e}_{\text{sim}}}[W^{(u)}_{t-1}(\hat{z})^2(1+\eta)] = (1+\eta)\mathbb{E}_{P^{\pi_e}_{\text{sim}}}[W^{(u)}_{t-1}(\hat{z})^2]$$

Unrolling this recurrence gives $\mathbb{E}[W^{(u)}_T(\hat{z})^2] \geq (1+\eta)^T$. Since the total estimator variance is $\mathbb{E}[W^{(u)}_T(\hat{z})^2] - 1$, we arrive at a strict lower bound of $(1+\eta)^T - 1$. □

D Example: Policy-Driven Misalignment

Consider a two-step environment. A user initiates a request $u_1 \in \{A, B\}$ with equal unconditional probability. The agent acts, and the user responds with a binary success outcome $\hat{z} \in \{0, 1\}$. We condition the simulation on a successful trajectory ($\hat{z} = 1$).

Under an accommodating behavior policy π_b, the agent acts such that the conditional success rates are $P^{\pi_b}(\hat{z} = 1 \mid u_1 = A) = 1.0$ and $P^{\pi_b}(\hat{z} = 1 \mid u_1 = B) = 0.5$. The marginal probability of success under the offline dataset is $P^{\pi_b}(\hat{z} = 1) = 0.75$.

We evaluate a strict new policy π_e, where the agent's actions uniformly reduce success rates: $P^{\pi_e}(\hat{z} = 1 \mid u_1 = A) = 0.5$ and $P^{\pi_e}(\hat{z} = 1 \mid u_1 = B) = 0.1$. The new marginal probability of success is $P^{\pi_e}(\hat{z} = 1) = 0.30$.

At the first step, the belief updates $M_1^{\pi}(u_1) = P^{\pi}(\hat{z} = 1 \mid u_1)/P^{\pi}(\hat{z} = 1)$ structurally misalign due to the policy shift:

$$M_1^{\pi_b}(A) = \frac{1.0}{0.75} = \frac{4}{3}, \quad M_1^{\pi_e}(A) = \frac{0.5}{0.30} = \frac{5}{3}$$
$$M_1^{\pi_b}(B) = \frac{0.5}{0.75} = \frac{2}{3}, \quad M_1^{\pi_e}(B) = \frac{0.1}{0.30} = \frac{1}{3}$$

The density ratios $\rho_1(u_1) = M_1^{\pi_e}(u_1)/M_1^{\pi_b}(u_1)$ evaluate to $\rho_1(A) = 1.25$ and $\rho_1(B) = 0.5$.

The simulator samples the initial state based strictly on the conditional distribution of its offline prior: $P_{\text{sim}}(u_1 = A \mid \hat{z} = 1) = 2/3$ and $P_{\text{sim}}(u_1 = B \mid \hat{z} = 1) = 1/3$. Because the expected value of ρ_1 is 1, the local label sensitivity V_1 evaluates to:

$$V_1 = \text{Var}_{u_1 \sim P_{\text{sim}}}(\rho_1) = \frac{2}{3}(1.25 - 1)^2 + \frac{1}{3}(0.5 - 1)^2 = 0.125$$

Even with a mathematically perfect simulator lacking any generative flaws, the mere divergence of the evaluation policy explicitly guarantees $V_1 \geq \eta = 0.125 > 0$, triggering the geometric variance explosion.

E Extended In-Distribution Results (WildChat)

This section contains the full, extended tables for the in-distribution WildChat evaluation referenced in Section 5.2. These tables break down the metrics into fine-grained deterministic statistics (Table 6), interaction friction (Table 7), semantic intent (Table 8), control adherence scores across all conditions (Table 9), and the full Shannon entropy measurements (Table 10).

Table 6: Empirical Distribution of Conversational Statistics across Simulator Types (Mean ± 95% CI). Unconditioned models truncate conversations; globally controlled models artificially inflate turns.

Simulator Type	**Turn Count**	**Avg. Words per Turn**	**Avg. Max Words per Turn**
Ground Truth Data (Human Baseline)	4.31 ± 0.01	113.33 ± 0.06	252.45 ± 0.13
Unconditioned SFT	3.55 ± 0.01	85.58 ± 0.16	137.11 ± 0.25
Prompted (Persona + Goal)	7.85 ± 0.02	75.12 ± 0.20	150.22 ± 0.35
Prompted (Cognitive Profile)	6.45 ± 0.01	88.30 ± 0.22	165.40 ± 0.30
Prompted (Scenario)	8.62 ± 0.02	68.45 ± 0.15	145.80 ± 0.25
Prompted (Scenario, Length-Constr.)	4.45 ± 0.01	92.15 ± 0.18	160.33 ± 0.31
Trajectory-Cond. SFT (Persona + Goal)	6.01 ± 0.01	84.85 ± 0.12	171.65 ± 0.27
Trajectory-Cond. SFT (Cognitive Profile)	5.19 ± 0.01	95.10 ± 0.17	181.66 ± 0.37
Trajectory-Cond. SFT (Scenario)	7.37 ± 0.01	81.35 ± 0.13	180.77 ± 0.32
Dynamic State SFT (Ours)	**4.78 ± 0.01**	**109.50 ± 0.20**	**198.92 ± 0.12**

F Extended Rejection Sampling Results

As discussed in Section 5.2, inference-time rejection sampling was evaluated as a potential baseline to enforce control adherence without altering the model's training distribution. In this approach, evaluators generate N candidate trajectories from an unconditioned simulator and select the one that maximizes target control alignment using an LLM judge. However, because global trajectory constraints require sequence-level coherence, standard inference-time rejection sampling suffers

Table 7: Task Progression and Conversational Friction (Proportion of Conversations).

Simulator Type	Task Unresolved	Iterative Refinement	Implicit Acceptance	Error Correction	Clarification Seeking
Ground Truth Data (Human Baseline)	85.4%	48.4%	20.6%	17.0%	9.7%
Unconditioned SFT	91.1%	32.3%	10.9%	21.7%	7.5%
Prompted (Persona + Goal)	88.5%	18.5%	8.5%	12.4%	5.2%
Prompted (Cognitive Profile)	89.1%	19.2%	7.2%	14.1%	6.1%
Prompted (Scenario)	87.4%	17.8%	9.1%	11.8%	4.9%
Prompted (Scenario, Length-Constr.)	88.0%	19.5%	9.8%	12.0%	5.0%
Trajectory-Cond. SFT (Persona + Goal)	83.0%	27.7%	10.0%	15.5%	9.6%
Trajectory-Cond. SFT (Cognitive Profile)	83.8%	29.1%	9.2%	17.5%	10.9%
Trajectory-Cond. SFT (Scenario)	83.2%	28.7%	10.5%	13.6%	11.5%
Dynamic State SFT (Ours)	76.8%	30.2%	6.8%	17.6%	10.8%

Table 8: Semantic Intent and Behavioral Pushback (Proportion of Conversations).

Simulator Type	Highly Detailed	Pushback	Complaints	Playfulness
Ground Truth Data (Human Baseline)	11.5%	2.1%	4.1%	7.0%
Unconditioned SFT	11.3%	1.8%	3.0%	3.5%
Prompted (Persona + Goal)	38.5%	1.2%	1.5%	1.8%
Prompted (Cognitive Profile)	28.4%	1.5%	2.1%	2.5%
Prompted (Scenario)	41.2%	1.1%	1.8%	1.5%
Prompted (Scenario, Length-Constr.)	38.2%	1.4%	1.9%	1.8%
Trajectory-Cond. SFT (Persona + Goal)	24.6%	2.6%	2.2%	5.5%
Trajectory-Cond. SFT (Cognitive Profile)	14.1%	2.7%	4.5%	6.1%
Trajectory-Cond. SFT (Scenario)	24.5%	3.0%	2.4%	5.3%
Dynamic State SFT (Ours)	21.1%	1.9%	4.3%	4.1%

from the vast combinatorial space of multi-turn dialogue. The unconditional prior naturally diverges from extreme persona constraints. As demonstrated in Table 11, even with enormous compute overhead ($N = 64$ candidates per turn, or $N = 512$ candidates per conversation), this search fails entirely, achieving an embedding similarity of just 0.28 ± 0.02. This validates the necessity of our training-time mitigations.

G Detailed ConvApparel Results

This section contains the full tables with explicitly quantified 95% confidence intervals covering our counterfactual evaluation setup. Specifically, Table 15 and Table 16 detail generative stability and behavioral adaptation under off-policy covariate shift. Furthermore, we provide comprehensive breakdown tables detailing deterministic conversational statistics (Table 17) and LLM-evaluated behavioral metrics (Table 18) across all simulated agent personas.

G.1 Absolute Variance by Conversation Horizon

In the main text (Table 4), we presented the normalized variance ratio to explicitly validate the theoretical $(1 + \eta)^T$ growth rate predicted by Theorem 2. For completeness, Table 12 provides the raw, absolute variance numbers for Total User Words across the held-out evaluation agents, stratified by the length of the interaction.

To further verify statistical stability into the long tail of interactions ($T \geq 6$), Table 13 reports the 95% confidence intervals for the normalized word-count variance ratios across all four held-out agents, alongside turn-by-turn cumulative negative emotion incidence (Table 14) against the *Efficient Matchmaker* agent ($n = 310$ human conversations vs. 1,000 simulated rollouts).

Table 9: Control Adherence Scores (Mean ± 95% CI) evaluated via LLM-as-a-Judge and cosine similarity.

Simulator Type	Persona Match	Goal Match	Behav. Rules	Semantic Entailment	Embedding Similarity
Ground Truth Data (Persona + Goal)	0.89 ± 0.01	0.89 ± 0.01	0.86 ± 0.01	0.90 ± 0.01	0.87 ± 0.01
Prompted Simulator (Persona + Goal)	0.96 ± 0.01	0.92 ± 0.01	0.92 ± 0.01	0.94 ± 0.01	0.93 ± 0.01
Trajectory-Cond. SFT (Persona + Goal)	0.77 ± 0.01	0.72 ± 0.01	0.64 ± 0.01	0.76 ± 0.01	0.69 ± 0.01
Ground Truth Data (Scenario)	0.92 ± 0.01	0.90 ± 0.01	0.89 ± 0.01	0.88 ± 0.01	0.88 ± 0.01
Prompted Simulator (Scenario)	0.85 ± 0.01	0.83 ± 0.01	0.83 ± 0.01	0.82 ± 0.01	0.82 ± 0.01
Trajectory-Cond. SFT (Scenario)	0.82 ± 0.01	0.71 ± 0.01	0.67 ± 0.01	0.68 ± 0.01	0.69 ± 0.01
Ground Truth Data (Cognitive Profile)	0.72 ± 0.01	0.73 ± 0.01	0.68 ± 0.01	0.72 ± 0.01	0.70 ± 0.01
Prompted Simulator (Cognitive Profile)	0.90 ± 0.01	0.90 ± 0.01	0.87 ± 0.01	0.88 ± 0.01	0.88 ± 0.01
Trajectory-Cond. SFT (Cognitive Profile)	0.56 ± 0.01	0.56 ± 0.01	0.48 ± 0.01	0.55 ± 0.01	0.52 ± 0.01

Table 10: Adherence Diversity (Shannon Entropy). Tighter global constraints induce behavioral collapse compared to ground truth.

Control Mechanism	Simulated Entropy	Ground Truth Entropy
Trajectory-Cond. SFT (Persona + Goal)	1.646	1.04
Prompted Simulator (Persona + Goal)	0.728	1.04
Trajectory-Cond. SFT (Scenario)	1.344	0.75
Prompted Simulator (Scenario)	0.927	0.75
Trajectory-Cond. SFT (Cognitive Profile)	1.281	1.10
Prompted Simulator (Cognitive Profile)	0.636	1.10

G.2 Direct Empirical Validation of Density Ratio Collapse (Theorem 2)

In Section 5.3 of the main text, we utilized the variance of an additive downstream feature (Total User Words) as an observable proxy for controllability collapse. However, Theorem 2 makes a strict mathematical prediction: for simulators conditioning on global trajectory labels ($\hat{z}$), the variance of the cumulative user generative error itself—the trajectory density ratio $W_t^{(u)}(\hat{z}) = \prod_{k=1}^{t} \rho_k$—must explode geometrically under policy shift.

To explicitly measure this, we must compute the step-wise density ratio $\rho_t(u_t) = M_t^{\pi_e}(u_t)/M_t^{\pi_b}(u_t)$. This requires estimating the true Bayesian belief updates $P^{\pi}(\hat{z} \mid h_t)$ under both the behavior and evaluation policies.

Implementation Details: Dual Offline Classifiers. Because our ConvApparel dataset uniquely contains real human interaction logs for both the training agents (π_b) and the held-out evaluation agents (π_e), we can isolate this exact mathematical mechanism. We train two separate auxiliary BERT-based classifiers (using the DeBERTa-v3-base architecture) via standard cross-entropy loss to act as probability probes: (1) **Behavior Policy Probe (C_{π_b}):** Trained exclusively on the offline logs of the 7 in-distribution training agents to predict the probability of a trajectory outcome $\hat{z}$ given a partial sequence history h_t; (2) **Evaluation Policy Probe (C_{π_e}):** Trained exclusively on the offline logs of the 4 held-out evaluation agents to predict the probability of $\hat{z}$ given h_t.

During closed-loop simulation against the held-out agents, we isolate the step-wise user generative error by passing strictly the pre-action observable history $\mathcal{F}_{t-} = (h_{t-1}, u_t)$ at each turn to both probes to dynamically compute $M_t^{\pi_e}$ and $M_t^{\pi_b}$. This isolates the step-wise generative error ρ_t. We simulate $N = 1{,}000$ trajectories per agent and track the variance of the cumulative density ratio $\text{Var}(W_t^{(u)})$.

Theoretical Exemption of Dynamic States. Crucially, Theorem 2 dictates that this collapse applies specifically to models conditioning on action-dependent trajectory labels $\hat{z}$. Our proposed *Dynamic State* mitigation is designed to deliberately evade this condition by conditioning strictly on $\mathcal{F}_{t-}$-measurable variables (z_t). Because the control is generated step-wise and is no longer a future-dependent outcome, the policy dependence mathematically cancels out (as proven rigorously in Theorem 3, Appendix A), yielding a theoretical step-wise error of $\rho_t = 1$. For this paradigm, the

Table 11: **Best-of-N Rejection Sampling Baseline vs. Computational Cost.** Inference-time filtering requires massive overhead yet fails to achieve meaningful adherence compared to Conditioned SFT.

Filtering Strategy	Avg. Model Calls per Conv.	Embedding Similarity (95% CI)	Persona Match (95% CI)
Per-Turn Rejection ($N = 2$)	20.12	0.01 ± 0.01	0.01 ± 0.01
Per-Turn Rejection ($N = 4$)	37.77	0.04 ± 0.01	0.04 ± 0.01
Per-Turn Rejection ($N = 8$)	74.59	0.05 ± 0.01	0.06 ± 0.01
Per-Turn Rejection ($N = 16$)	144.78	0.07 ± 0.02	0.08 ± 0.02
Per-Turn Rejection ($N = 64$)	573.79	0.11 ± 0.02	0.13 ± 0.02
Per-Conv Rejection ($N = 16$)	56.8	0.08 ± 0.01	0.09 ± 0.02
Per-Conv Rejection ($N = 256$)	908.8	0.21 ± 0.02	0.25 ± 0.02
Per-Conv Rejection ($N = 512$)	1817.6	0.28 ± 0.02	0.34 ± 0.03
Conditioned SFT	6	0.69 ± 0.01	0.77 ± 0.01

Table 12: **Absolute Variance of Total User Words by Conversation Horizon (ConvApparel Held-Out Agents).** Ground truth computed from $n = 1{,}267$ real conversations. Under covariate shift, the Static Agent-Agnostic simulator's absolute variance compounds geometrically with conversation length. The Dynamic State Agent-Aware approach maintains near-human absolute variance at all horizons.

Simulator Paradigm	Var(Total User Words)					Avg. Turn-over-Turn Multiplier
	$T = 2$	$T = 3$	$T = 4$	$T = 5$	$T \geq 6^*$	
Ground Truth Data	185	419	606	975	4,835	—
Static Global, Agent-Agnostic	326	989	2,133	4,768	34,764	$\approx 1.42\times$ per turn
Static Global, Agent-Aware	237	561	818	1,765	10,347	$\approx 1.14\times$ per turn
Dynamic State, Agent-Agnostic	211	482	782	1,316	6,962	$\approx 1.06\times$ per turn
Dynamic State, Agent-Aware (Ours)	**191**	**415**	**636**	**995**	**5,173**	$\approx \mathbf{1.01}\times$ per turn

* *Note:* The $T \geq 6$ column aggregates the long-tail of conversations. The Ground Truth variance naturally jumps due to this diverse aggregation bucket, but the step-wise multiplier isolates the Simulator's relative error.

Table 13: **Normalized Variance Ratios ($\text{Var}_{\text{sim}}/\text{Var}_{\text{GT}}$) with 95% Confidence Intervals across Horizons.**

Simulator Paradigm	$T = 2$	$T = 3$	$T = 4$	$T = 5$	$T \geq 6$
Static Global, Agent-Agnostic	1.76 [0.34, 3.00]	2.36 [0.39, 4.84]	3.52 [0.56, 4.89]	4.89 [1.25, 8.30]	7.19 [1.18, 16.27]
Static Global, Agent-Aware	1.28 [0.99, 1.69]	1.34 [1.08, 1.71]	1.35 [0.92, 2.02]	1.81 [0.80, 3.37]	2.14 [1.74, 4.17]
Dynamic State, Agent-Agnostic	1.14 [0.69, 1.59]	1.15 [0.86, 1.79]	1.29 [0.77, 1.71]	1.35 [0.80, 2.10]	1.44 [0.52, 5.65]
Dynamic State, Agent-Aware (Ours)	**1.03** [**0.81**, **1.38**]	**0.99** [**0.91**, **1.19**]	**1.05** [**0.74**, **1.52**]	**1.02** [**0.59**, **1.90**]	**1.07** [**0.93**, **1.57**]

Table 14: **Granular Turn-Level Behavioral Dynamics (Cumulative Negative Emotion Incidence % $\pm$ 95% CI).** Probability that a user expresses negative emotion by turn t when interacting with the held-out *Efficient Matchmaker* agent.

System	n	$t = 1$	$t = 2$	$t = 3$	$t = 4$	$t = 5$
Ground Truth Data (Human)	310	$5.2 \pm 2.5\%$	$9.0 \pm 3.2\%$	$11.0 \pm 3.5\%$	$11.9 \pm 3.6\%$	$11.9 \pm 3.6\%$
Static Global, Agent-Agnostic	1000	$8.9 \pm 1.8\%$	$16.8 \pm 2.3\%$	$21.5 \pm 2.5\%$	$25.0 \pm 2.7\%$	$25.9 \pm 2.7\%$
Static Global, Agent-Aware	1000	$7.4 \pm 1.6\%$	$11.6 \pm 2.0\%$	$14.0 \pm 2.2\%$	$16.0 \pm 2.3\%$	$16.3 \pm 2.3\%$
Dynamic State, Agent-Agnostic	1000	$6.3 \pm 1.5\%$	$12.4 \pm 2.0\%$	$15.7 \pm 2.3\%$	$17.8 \pm 2.4\%$	$18.0 \pm 2.4\%$
Dynamic State, Agent-Aware (Ours)	1000	$\mathbf{3.8 \pm 1.2}\%$	$\mathbf{8.0 \pm 1.7}\%$	$\mathbf{10.6 \pm 1.9}\%$	$\mathbf{11.3 \pm 2.0}\%$	$\mathbf{11.8 \pm 2.0}\%$

density ratio W_t is computed with respect to the sequence of step-wise controls $z_{1:t}$ rather than global $\hat{z}$.

Results. Table 19 confirms our theoretical framework. For the standard trajectory-conditioned simulator, the misalignment in belief updates ensures a non-zero local label sensitivity ($V_t > 0$). Subjected to statistical noise, this causes the variance of the cumulative density ratio to violently explode as T grows. The Agent-Aware static model slows this growth but cannot escape the structural look-ahead bias of global $\hat{z}$. In stark contrast, the Dynamic State approach natively circumvents Theorem 2, maintaining near-zero variance in its generative density ratio (with minor fluctuations strictly due to neural approximation and classifier calibration error).

Table 15: Generative Stability under Off-Policy Covariate Shift (Full Extent). The Dynamic State Agent-Aware variant achieves the closest parity to the human baseline across varying off-policy personas.

Held-Out Evaluation Agent	Simulator Paradigm	Avg. Words per Turn	Avg. Max Words in a Turn	Total User Words
Domain Expert	Ground Truth Data	12.01 ± 0.89	16.52 ± 1.26	38.71 ± 2.95
	Static Global - Agent-Agnostic	8.81 ± 0.52	12.35 ± 0.81	28.02 ± 1.55
	Static Global - Agent-Aware	10.53 ± 0.64	14.22 ± 0.95	32.01 ± 2.01
	Dynamic State - Agent-Agnostic	11.24 ± 0.58	15.11 ± 0.88	35.15 ± 1.84
	Dynamic State - Agent-Aware	12.12 ± 0.62	16.44 ± 1.02	38.53 ± 2.21
Empathetic Listener	Ground Truth Data	11.27 ± 0.69	15.51 ± 1.02	41.06 ± 3.74
	Static Global - Agent-Agnostic	15.22 ± 1.21	21.05 ± 1.82	48.54 ± 3.52
	Static Global - Agent-Aware	12.51 ± 0.82	17.53 ± 1.24	43.22 ± 2.81
	Dynamic State - Agent-Agnostic	10.54 ± 0.63	14.82 ± 0.91	38.91 ± 2.12
	Dynamic State - Agent-Aware	11.42 ± 0.71	15.61 ± 1.05	40.83 ± 2.54
Efficient Matchmaker	Ground Truth Data	11.69 ± 0.75	16.24 ± 1.26	39.74 ± 3.48
	Static Global - Agent-Agnostic	25.41 ± 2.52	45.21 ± 4.23	92.12 ± 8.51
	Static Global - Agent-Aware	13.22 ± 0.91	18.51 ± 1.52	43.51 ± 3.22
	Dynamic State - Agent-Agnostic	14.11 ± 1.12	19.42 ± 1.63	45.24 ± 3.51
	Dynamic State - Agent-Aware	11.82 ± 0.81	16.51 ± 1.21	40.12 ± 2.82
Enthusiastic Rambler	Ground Truth Data	12.38 ± 0.85	16.73 ± 1.26	41.28 ± 4.23
	Static Global - Agent-Agnostic	8.61 ± 0.51	12.11 ± 0.82	29.82 ± 1.81
	Static Global - Agent-Aware	10.12 ± 0.72	14.22 ± 1.01	35.41 ± 2.22
	Dynamic State - Agent-Agnostic	11.51 ± 0.82	15.52 ± 1.12	38.92 ± 2.51
	Dynamic State - Agent-Aware	12.51 ± 0.91	16.81 ± 1.32	41.12 ± 3.01

Table 16: Behavioral and Emotional Adaptation to Held-Out Agents (Full Extent). Proportion of Conversations marked $\pm$ 95% CI.

Held-Out Evaluation Agent	Simulator Paradigm	Urgency / Impatience	Negative Sentiment	Iterative Refinement	Clarification Seeking
Domain Expert	Ground Truth Data	$8.4 \pm 3.0\%$	$13.4 \pm 3.7\%$	$31.7 \pm 5.1\%$	$11.2 \pm 3.4\%$
	Static Global - Agent-Agnostic	$14.5 \pm 2.5\%$	$18.0 \pm 3.2\%$	$12.5 \pm 2.1\%$	$3.5 \pm 1.0\%$
	Static Global - Agent-Aware	$11.2 \pm 2.1\%$	$15.1 \pm 2.5\%$	$25.0 \pm 3.5\%$	$8.2 \pm 1.8\%$
	Dynamic State - Agent-Agnostic	$10.5 \pm 1.9\%$	$14.5 \pm 2.1\%$	$22.3 \pm 3.0\%$	$7.4 \pm 1.5\%$
	Dynamic State - Agent-Aware	$8.6 \pm 2.0\%$	$13.6 \pm 2.5\%$	$31.2 \pm 4.0\%$	$11.0 \pm 2.5\%$
Empathetic Listener	Ground Truth Data	$7.5 \pm 2.9\%$	$8.1 \pm 3.0\%$	$34.6 \pm 5.2\%$	$4.0 \pm 2.2\%$
	Static Global - Agent-Agnostic	$12.1 \pm 2.8\%$	$16.5 \pm 3.5\%$	$15.4 \pm 2.5\%$	$8.0 \pm 2.0\%$
	Static Global - Agent-Aware	$9.5 \pm 2.2\%$	$11.2 \pm 2.4\%$	$28.4 \pm 3.8\%$	$5.5 \pm 1.5\%$
	Dynamic State - Agent-Agnostic	$8.8 \pm 2.0\%$	$12.1 \pm 2.2\%$	$25.0 \pm 3.1\%$	$6.2 \pm 1.4\%$
	Dynamic State - Agent-Aware	$7.8 \pm 2.1\%$	$8.5 \pm 2.3\%$	$34.1 \pm 4.2\%$	$4.2 \pm 1.6\%$
Efficient Matchmaker	Ground Truth Data	$6.1 \pm 2.7\%$	$11.9 \pm 3.6\%$	$41.3 \pm 5.5\%$	$6.1 \pm 2.7\%$
	Static Global - Agent-Agnostic	$22.4 \pm 4.5\%$	$26.8 \pm 5.2\%$	$8.8 \pm 2.1\%$	$15.6 \pm 3.8\%$
	Static Global - Agent-Aware	$10.5 \pm 2.5\%$	$16.4 \pm 3.1\%$	$32.6 \pm 4.5\%$	$8.4 \pm 2.1\%$
	Dynamic State - Agent-Agnostic	$12.2 \pm 2.8\%$	$18.2 \pm 3.4\%$	$24.8 \pm 3.6\%$	$10.2 \pm 2.4\%$
	Dynamic State - Agent-Aware	$6.5 \pm 2.2\%$	$12.2 \pm 2.8\%$	$40.9 \pm 4.8\%$	$6.4 \pm 2.0\%$
Enthusiastic Rambler	Ground Truth Data	$8.0 \pm 3.0\%$	$8.9 \pm 3.2\%$	$43.0 \pm 5.5\%$	$5.7 \pm 2.6\%$
	Static Global - Agent-Agnostic	$12.5 \pm 2.5\%$	$14.5 \pm 2.8\%$	$16.5 \pm 2.5\%$	$1.8 \pm 0.8\%$
	Static Global - Agent-Aware	$9.8 \pm 2.1\%$	$11.2 \pm 2.4\%$	$30.5 \pm 3.8\%$	$3.2 \pm 1.2\%$
	Dynamic State - Agent-Agnostic	$9.2 \pm 2.0\%$	$12.0 \pm 2.5\%$	$28.5 \pm 3.5\%$	$3.8 \pm 1.4\%$
	Dynamic State - Agent-Aware	$8.2 \pm 2.2\%$	$9.2 \pm 2.4\%$	$42.5 \pm 4.5\%$	$5.5 \pm 1.8\%$

Structural Collapse vs. Neural Approximation Drift. Note that while Theorem 3 assumes an idealized simulator ($P_{\text{sim}} = P^{\pi_b}$) to isolate structural causal flaws from estimation error, finite neural capacity and classifier calibration limits in practice introduce a small bounded step-wise approximation error $V_t \leq \epsilon$. Under a bounded local error ϵ, the cumulative variance expands as $\text{Var}(W_T^{(u)}) \leq (1+\epsilon)^T - 1 = T\epsilon + \mathcal{O}((T\epsilon)^2)$. Because ϵ is very small ($\hat{\epsilon} \approx 0.01$), higher-order terms are negligible over typical dialogue horizons ($T \leq 6$), converting geometric explosion into a benign linear drift of $\approx +0.01$ per turn (matching the $0.02 \rightarrow 0.05$ progression for Dynamic State, Agent-Aware in Table 19).

G.3 Robustness to Automated Annotator Label Noise

Because step-wise dynamic controls rely on an automated LLM annotator to infer intermediate states z_t during training data preparation, a natural question is how sensitive the resulting simulator is to label noise or systematic annotator bias. Structurally, because the dynamic annotator only observes

Table 17: Conversational Statistics across All ConvApparel Agent Personas (Ground Truth Data). Mean ± 95% CI.

Agent Persona	Turn Count	Avg. Words/Turn	Max Words/Turn	Total User Words	Question Freq.
Footwear — Held-Out Agents					
Domain Expert	3.33 ± 0.15	12.01 ± 0.89	16.52 ± 1.26	38.71 ± 2.95	0.24 ± 0.04
Empathetic Listener	3.47 ± 0.17	11.27 ± 0.69	15.51 ± 1.02	41.06 ± 3.74	0.22 ± 0.03
Efficient Matchmaker	3.36 ± 0.15	11.69 ± 0.75	16.24 ± 1.26	39.74 ± 3.48	0.22 ± 0.03
Enthusiastic Rambler	3.21 ± 0.16	12.38 ± 0.85	16.73 ± 1.26	41.28 ± 4.23	0.22 ± 0.03
Footwear — In-Distribution Agents					
Baseline (Good)	3.59 ± 0.12	8.96 ± 0.51	13.94 ± 0.84	35.49 ± 2.38	0.19 ± 0.02
Good Rec	3.54 ± 0.17	10.88 ± 0.67	15.19 ± 0.99	38.63 ± 3.12	0.19 ± 0.03
Bad	3.57 ± 0.21	10.44 ± 1.12	15.86 ± 2.01	42.23 ± 5.97	0.16 ± 0.03
Hesitant Assistant	3.19 ± 0.12	13.80 ± 1.20	18.48 ± 1.72	42.94 ± 3.99	0.21 ± 0.03
Literal Thinker	3.12 ± 0.14	12.64 ± 1.16	16.46 ± 1.48	37.88 ± 3.35	0.21 ± 0.03
Mild Upseller	3.26 ± 0.14	12.97 ± 0.96	17.63 ± 1.37	41.32 ± 3.21	0.22 ± 0.03
Patient Guide	3.62 ± 0.18	11.82 ± 1.07	16.62 ± 1.34	40.95 ± 3.81	0.16 ± 0.03
Trend Chaser	3.17 ± 0.13	13.07 ± 0.94	17.70 ± 1.41	41.13 ± 3.46	0.22 ± 0.03
Visual Stylist	3.30 ± 0.15	11.07 ± 0.91	15.34 ± 1.30	37.22 ± 3.55	0.21 ± 0.03
Other Product Categories					
Tops (Good)	3.55 ± 0.11	8.79 ± 0.49	12.98 ± 0.75	34.94 ± 2.45	0.19 ± 0.02
Tops (Bad)	3.48 ± 0.23	9.01 ± 1.10	13.41 ± 1.78	37.16 ± 6.06	0.10 ± 0.02
Bottoms (Good)	3.45 ± 0.11	8.86 ± 0.48	13.22 ± 0.77	33.54 ± 2.24	0.19 ± 0.02
Bottoms (Bad)	3.19 ± 0.20	8.78 ± 1.26	13.02 ± 1.87	31.49 ± 5.15	0.13 ± 0.03
Outerwear (Good)	3.62 ± 0.12	9.28 ± 0.50	14.17 ± 0.84	36.76 ± 2.48	0.20 ± 0.02
Outerwear (Bad)	3.35 ± 0.23	9.06 ± 1.06	13.79 ± 1.78	34.73 ± 4.88	0.17 ± 0.04
All Combined	$\mathbf{3.43 \pm 0.03}$	$\mathbf{10.48 \pm 0.18}$	$\mathbf{15.01 \pm 0.27}$	$\mathbf{37.58 \pm 0.78}$	—

Table 18: LLM-Evaluated Behavioral Metrics across All ConvApparel Agent Personas (Ground Truth Data). Proportion of conversations ± 95% CI.

Agent Persona	Urgency	Neg. Sent.	Iter. Refine	Clarification	Error Correction	Task Unresolved
Footwear — Held-Out Agents						
Domain Expert	$8.4 \pm 3.0\%$	$13.4 \pm 3.7\%$	$31.7 \pm 5.1\%$	$11.2 \pm 3.4\%$	$19.6 \pm 4.3\%$	$89.1 \pm 3.4\%$
Empathetic Listener	$7.5 \pm 2.9\%$	$8.1 \pm 3.0\%$	$34.6 \pm 5.2\%$	$4.0 \pm 2.2\%$	$13.4 \pm 3.7\%$	$75.7 \pm 4.7\%$
Efficient Matchmaker	$6.1 \pm 2.7\%$	$11.9 \pm 3.6\%$	$41.3 \pm 5.5\%$	$6.1 \pm 2.7\%$	$16.8 \pm 4.2\%$	$81.9 \pm 4.3\%$
Enthusiastic Rambler	$8.0 \pm 3.0\%$	$8.9 \pm 3.2\%$	$43.0 \pm 5.5\%$	$5.7 \pm 2.6\%$	$21.0 \pm 4.5\%$	$82.8 \pm 4.2\%$
Footwear — In-Distribution Agents						
Baseline (Good)	$4.6 \pm 1.4\%$	$4.7 \pm 1.4\%$	$23.1 \pm 2.9\%$	$2.2 \pm 1.0\%$	$8.5 \pm 1.9\%$	$78.1 \pm 2.8\%$
Good Rec	$5.4 \pm 2.5\%$	$10.7 \pm 3.4\%$	$22.1 \pm 4.6\%$	$4.7 \pm 2.3\%$	$17.0 \pm 4.1\%$	$73.5 \pm 4.9\%$
Bad	$16.7 \pm 4.9\%$	$14.0 \pm 4.6\%$	$11.8 \pm 4.2\%$	$1.8 \pm 1.8\%$	$20.8 \pm 5.4\%$	$99.1 \pm 1.2\%$
Hesitant Assistant	$8.6 \pm 3.0\%$	$10.1 \pm 3.3\%$	$50.8 \pm 5.4\%$	$5.5 \pm 2.5\%$	$12.8 \pm 3.6\%$	$94.8 \pm 2.4\%$
Literal Thinker	$4.6 \pm 2.3\%$	$11.7 \pm 3.5\%$	$32.5 \pm 5.1\%$	$2.8 \pm 1.8\%$	$15.3 \pm 3.9\%$	$86.5 \pm 3.7\%$
Mild Upseller	$5.9 \pm 2.6\%$	$11.5 \pm 3.5\%$	$32.5 \pm 5.1\%$	$8.4 \pm 3.0\%$	$21.1 \pm 4.4\%$	$91.6 \pm 3.0\%$
Patient Guide	$11.9 \pm 3.6\%$	$14.1 \pm 3.8\%$	$8.2 \pm 3.0\%$	$4.4 \pm 2.2\%$	$23.2 \pm 4.6\%$	$99.4 \pm 0.9\%$
Trend Chaser	$6.6 \pm 2.7\%$	$10.3 \pm 3.3\%$	$41.6 \pm 5.4\%$	$6.6 \pm 2.7\%$	$24.1 \pm 4.7\%$	$83.8 \pm 4.0\%$
Visual Stylist	$7.2 \pm 2.8\%$	$8.7 \pm 3.0\%$	$48.1 \pm 5.4\%$	$9.3 \pm 3.1\%$	$17.9 \pm 4.1\%$	$87.8 \pm 3.5\%$
Other Product Categories						
Tops (Good)	$4.7 \pm 1.4\%$	$6.2 \pm 1.6\%$	$24.8 \pm 2.9\%$	$2.2 \pm 1.0\%$	$13.2 \pm 2.3\%$	$78.8 \pm 2.8\%$
Tops (Bad)	$10.7 \pm 4.2\%$	$9.3 \pm 4.0\%$	$9.8 \pm 4.1\%$	$2.0 \pm 1.9\%$	$18.0 \pm 5.3\%$	$99.0 \pm 1.3\%$
Bottoms (Good)	$2.8 \pm 1.1\%$	$3.6 \pm 1.3\%$	$22.2 \pm 2.8\%$	$2.6 \pm 1.1\%$	$10.2 \pm 2.1\%$	$75.6 \pm 2.9\%$
Bottoms (Bad)	$6.6 \pm 3.4\%$	$5.6 \pm 3.2\%$	$8.6 \pm 3.9\%$	$1.5 \pm 1.7\%$	$10.1 \pm 4.2\%$	$100.0 \pm 0.0\%$
Outerwear (Good)	$3.8 \pm 1.3\%$	$6.0 \pm 1.6\%$	$23.0 \pm 2.9\%$	$4.1 \pm 1.3\%$	$11.1 \pm 2.1\%$	$77.1 \pm 2.8\%$
Outerwear (Bad)	$9.3 \pm 4.0\%$	$8.8 \pm 3.9\%$	$8.3 \pm 3.8\%$	$1.5 \pm 1.6\%$	$12.7 \pm 4.6\%$	$98.5 \pm 1.6\%$

the prefix $\mathcal{F}_{t-}$, any annotation error remains strictly adapted to the current filtration and cannot inject future-leaking look-ahead bias. To quantify this empirically, we evaluate our clean simulator's predicted states against the first-person human emotion labels in ConvApparel-V2, and train four

Table 19: **Direct Empirical Validation of Theorem 2.** The empirical variance of the cumulative user generative error, $\text{Var}(W_t^{(u)})$, evaluated via dual offline classifiers. As predicted, standard static trajectory controls suffer from geometric variance explosion due to belief update mismatch under covariate shift. By conditioning strictly on $\mathcal{F}_{t-}$-measurable states, our dynamic approach theoretically and empirically circumvents this collapse.

Simulator Paradigm	$T = 2$	$T = 3$	$T = 4$	$T = 5$
Static Global, Agent-Agnostic (Standard)	0.24 ± 0.05	0.67 ± 0.11	1.82 ± 0.28	5.14 ± 0.72
Static Global, Agent-Aware	0.15 ± 0.03	0.28 ± 0.06	0.51 ± 0.12	0.94 ± 0.19
Dynamic State, Agent-Agnostic	0.06 ± 0.02	0.09 ± 0.03	0.12 ± 0.04	0.14 ± 0.05
Dynamic State, Agent-Aware (Ours)	$\mathbf{0.02 \pm 0.01}$	$\mathbf{0.03 \pm 0.01}$	$\mathbf{0.04 \pm 0.01}$	$\mathbf{0.05 \pm 0.02}$

additional SFT simulators under synthetic training label corruption (10%, 20%, and 30% uniform random state swaps, as well as a 20% systematic bias shifting negative states to neutral).

Table 20: **Annotator Noise Robustness Ablation.** Even under 30% random label corruption or 20% systematic annotator bias, the Dynamic State simulator maintains a bounded variance ratio at $T = 5$ (1.26–1.31), outperforming the uncorrupted Static Trajectory-Conditioned baseline (4.89).

Training Label Corruption	**Simulator State F1 vs. Human GT**	**Variance Ratio ($\text{Var}_{\text{sim}}/\text{Var}_{\text{GT}}$) at $T =$**
Clean Labels (0% Corruption)	0.61	1.02
10% Uniform Random Corruption	0.61	1.10
20% Uniform Random Corruption	0.55	1.18
30% Uniform Random Corruption	0.44	1.26
20% Targeted Bias (Negative $\rightarrow$ Neutral)	0.56	1.31
Ref: Static Global SFT (0% Corruption)	—	4.89

As shown in Table 20, even when 30% of training labels are corrupted, the dynamic simulator's variance ratio reaches only 1.26 at $T = 5$ (compared to 4.89 for the standard trajectory-conditioned model trained on clean data), confirming that filtration-adapted step-wise errors decay as local noise rather than compounding into look-ahead bias.

H Implementation Details

To ensure complete reproducibility of our empirical findings, we detail the training setup for the generative simulators, the automated evaluation loop, and the LLM-as-a-judge pipelines.

H.1 Generative Simulator Training Setup

All user simulators, including the Unconditioned SFT, Trajectory-Conditioned SFT, and our proposed Dynamic State and Agent-Aware models, were initialized and trained using Gemini 2.5 models via standard Supervised Fine-Tuning (SFT). Models were trained until convergence, utilizing a global batch size of 64. Specifically, training converged after 3,000 steps for the simulators trained on the WildChat dataset, and after 500 steps for those trained on the ConvApparel dataset.

H.2 LLM-as-a-Judge and Annotation Pipeline

We utilized Gemini 3.1 Pro as our automated annotator for extracting offline dataset controls and as the LLM-as-a-judge for evaluating constraint adherence. The exact prompt templates utilized for trajectory-level annotation (Persona+Goal, Scenario, Cognitive Profile) and step-level dynamic state annotation are detailed in Appendix I.

H.3 Automated Evaluation Proxy Agents

To conduct the closed-loop multi-turn offline evaluations reproducibly, we trained proxy SFT agents. These agents were also trained via standard SFT using Gemini 2.5 models. The WildChat Agent was

trained on the empirical distribution of assistant responses from the WildChat multi-turn conversations. The model was fine-tuned for 1,200 iterations. Similarly, the ConvApparel Agent was trained via SFT to explicitly condition on the distinct assistant system prompts available in the data. Training for this agent converged after approximately 400 iterations.

I Prompt Templates

This appendix provides the exact prompt templates used in our methodology, including the offline dataset annotation prompts, the user simulator system prompts, and the agent evaluation personas for the ConvApparel environment.

I.1 Offline Annotation Prompts

I.1.1 Step-Level Dynamic State Annotation

```
You are an expert Behavioral Analyst and Psychologist specializing in
Human-Computer Interaction. Your task is to analyze a conversation between
a User and an AI Assistant and infer the **latent internal state** of the
User at a specific turn.

### DEFINITION of "INTERNAL STATE"
The internal state is a synthesis of:
1.  **Emotional Affect:** (e.g., Frustrated, Delight, Neutral, Anxious,
    Unknown)
2.  **Cognitive Load:** (e.g., Overwhelmed, Focused, Exploring, Confused,
    Unknown)
3.  **Implicit Intent:** What they *actually* want, beyond the literal text
    (e.g., "Testing the system," "Urgent problem solving,"
    "Loneliness/Chitchat", or Unknown).

### INSTRUCTIONS
1.  Read the provided [CONVERSATION_HISTORY] at {turn_num}.
2.  Infer the user's internal state at that exact moment.
3.  Take into account previous internal states (if they exist).
4.  Output strictly valid JSON.
5.  The description must be concise (maximum 30 words).

### INPUT DATA

[CONVERSATION_HISTORY]
{history}
[END HISTORY]

### OUTPUT FORMAT
Response must be a single valid JSON object:

{
  "emotional_affect": "String describing the state (max 10 words)",
  "cognitive_load": "String describing the state (max 10 words)",
  "implicit_intent": "String describing the state (max 10 words)"
}
```

I.1.2 Trajectory-Level Persona and Goal

```
You are an expert Behavioral Analyst and Data Labeler.
Your task is to reverse-engineer a "User Persona" and "Task Specification"
from a conversation log. This will be used to program a User Simulator.

### GUIDELINES
```

```
  - **Analyze the User Only:** The Agent's behavior is merely the environment
    the user is reacting to.
  - **Identify Logic over Content:** Don't just list what they said; identify
    the *rule* they followed (e.g., "If the agent asks for a date, the user
    provides a range rather than a specific day").
  - **Persona vs. Task:** Distinguish between how the user acts (Persona) and
    what the user wants (Task).

### DIMENSIONS OF ANALYSIS

1.  **Linguistic Fingerprint:** The specific syntax and vocabulary constraints.
2.  **The Logic Gate:** How the user processes information. Do they verify
    the agent's work? Do they provide all info at once or wait to be prompted?
3.  **Friction Thresholds:** What specific agent actions (repetition,
    misunderstanding, over-verbosity) trigger a change in user behavior?

### INPUT DATA

[CONVERSATION_LOG]
{history}
[CONVERSATION_END]

### OUTPUT FORMAT

Provide a single valid JSON object. No markdown, no conversational filler.

{
  "persona_profile": {
    "traits": {
      "tone": "e.g., Clinical, Frantic, Casual",
      "technical_literacy": "Low|Medium|High",
      "verbosity_profile": "e.g., Bulleted lists, run-on sentences, etc."
    },
    "behavioral_rules": {
      "information_disclosure": "How they share info (e.g., 'Minimalist...')",
      "error_correction": "How they fix agent mistakes (e.g., 'Aggressive...')",
      "patience_trigger": "Specific agent behavior causing frustration"
    }
  },
  "task_specification": {
    "goal": "The high-level intent.",
    "mandatory_requirements": ["Constraint 1", "Constraint 2"],
    "success_definition": "The exact confirmation that concludes the task.",
    "initial_context": "What the user knows/feels at the start."
  },
  "simulator_instructions": "Prompt for simulator: 'You are [Persona]...'"
}
```

I.1.3 Trajectory-Level Cognitive Profile

```
Your goal is to reverse-engineer the "Ground Truth" of the user to train a
high-fidelity User Simulator.

You must extract two distinct layers of user data:

1.  **The Context:** The external facts (Demographics, Environment, Role).
2.  **The Latents:** The internal cognitive traits.
```

```
**Uncertainty Protocol:**
For every attribute, you must assign a confidence level (HIGH, MEDIUM,
LOW, N/A).

  * *Crucial:* If the user does not explicitly state their age/job, you can
    try to **INFER** it from their vocabulary, topic, and constraints, but
    mark confidence as LOW or MEDIUM.
    If it is absolutely not inferrable, mark as Unknown, or don't include the
    field.

-----

### **Annotation Schema (JSON)**

Analyze the trajectory and populate the following JSON structure.

#### **1. Context & Demographics (External Factors)**

  * **Role/Occupation:** (e.g., Student, Software Engineer, Parent, Gamer).
    *Infer from topic complexity.*
  * **Age Group:** (e.g., Teenager, University Student, Adult, Elderly).
    *Infer from slang, reference years, or life milestones.*
  * **Cultural/Geo Location:** (e.g., US-centric, South Asian, UK).
    *Infer from spelling, currency, or regional references.*
  * **Technical Environment:** (e.g., Mobile User, Python Environment,
    Corporate Network). *Infer from formatting constraints or code snippets.*
  * **Domain Experience:** Specific background in the current task topic
    (e.g., "Has used iPhone for 10 years").

#### **2. Cognitive & Epistemic**

  * **Domain Proficiency:** [Novice, Competent, Expert, Polymath]
  * **System/AI Literacy:** [Naïve, Keyword Searcher, Conversationalist,
    Power User]
  * **Need for Cognition:** [Result-Oriented (Just answer), Process-Oriented
    (Teach me), Concept-Oriented (Why?)]
  * **Mental Model Rigidity:** [Rigid, Negotiable, Malleable]

#### **3. Process & Goals (Strategy)**

  * **Optimization Strategy:** [Satisficer (Good enough), Maximizer
    (Best possible), Perfectionist]
  * **Goal Clarity:** [Vague, Abstract, Concrete, Rigid]
  * **Locus of Control:** [Director (User leads), Co-Creator, Passenger
    (Agent leads)]
  * **Scaffolding Need:** [Step-by-Step, Holistic, Hybrid]

#### **4. Communication Style (Surface)**

  * **Verbosity:** [Telegraphic, Concise, Conversational, Narrative]
  * **Tone:** [Formal, Casual, Adversarial, Urgent]

-----

### **Input Conversation:**

[CONVERSATION_START]
{history}
[CONVERSATION_END]
```

```
### **Output Format (JSON)**

Respond with valid JSON only. Do not use markdown blocks.

{
  "user_context": {
    "role_occupation": {
      "value": "string or null",
      "confidence": "HIGH|MEDIUM|LOW"
    },
    "age_group": {
      "value": "string or null",
      "confidence": "HIGH|MEDIUM|LOW"
    },
    "cultural_geo": {
      "value": "string or null",
      "confidence": "HIGH|MEDIUM|LOW"
    },
    "technical_env": {
      "value": "string or null",
      "confidence": "HIGH|MEDIUM|LOW"
    },
    "domain_experience_context": {
      "value": "string or null",
      "confidence": "HIGH|MEDIUM|LOW"
    }
  },
  "latents": {
    "cognitive": {
      "domain_proficiency": { "value": "Novice|Competent|...",
                              "confidence": "HIGH|MEDIUM|LOW" },
      "system_literacy": { "value": "Naïve|Keyword Searcher|...",
                           "confidence": "HIGH|MEDIUM|LOW" },
      "need_for_cognition": { "value": "Result-Oriented|...",
                              "confidence": "HIGH|MEDIUM|LOW" },
      "mental_model_rigidity": { "value": "Rigid|Negotiable|...",
                                 "confidence": "HIGH|MEDIUM|LOW" }
    },
    "process": {
      "optimization_strategy": { "value": "Satisficer|Maximizer|...",
                                 "confidence": "HIGH|MEDIUM|LOW" },
      "goal_clarity": { "value": "Vague|Abstract|Concrete|Rigid",
                        "confidence": "HIGH|MEDIUM|LOW" },
      "locus_of_control": { "value": "Director|Co-Creator|Passenger",
                            "confidence": "HIGH|MEDIUM|LOW" },
      "scaffolding_need": { "value": "Step-by-Step|Holistic|Hybrid",
                            "confidence": "HIGH|MEDIUM|LOW" }
    },
    "communication_style": {
      "verbosity": { "value": "Telegraphic|Concise|Conversational|...",
                     "confidence": "HIGH|MEDIUM|LOW" },
      "tone": { "value": "string", "confidence": "HIGH|MEDIUM|LOW" }
    }
  },
  "simulator_instructions": {
    "persona_summary": "A 1-sentence summary of who to act like.",
    "behavioral_directive": "Specific instructions on how to handle errors..."
  }
```

```
}
```

I.1.4 Trajectory-Level Scenario Generation

```
You are an expert dataset annotator. You will be given a conversation between
a user and an assistant.
Your goal is to analyze the interaction and output a JSON object that
describes the scenario.
This description will be used to train a "Scenario Generator" model, which
takes an instruction and generates a similar conversation.

## INPUT CONVERSATION

[CONVERSATION_START]
{history}
[CONVERSATION_END]

## INSTRUCTIONS

Analyze the conversation above and output a JSON object with the following
fields:

1.  **user_profile**:

      * ‘tone‘: The emotional state or attitude of the user (e.g., curious,
        demanding, playful, frustrated, academic, urgent).
      * ‘skill_level‘: The apparent expertise of the user regarding the topic
        (e.g., beginner coder, fanfiction enthusiast, student, general public).
      * ‘intent_category‘: The high-level category of the request (e.g.,
        Creative Writing, Coding/Debugging, Academic Help, Roleplay,
        Information Seeking, Troubleshooting).

2.  **task_attributes**:

      * ‘topic‘: The specific subject matter (e.g., "Python pandas dataframe",
        "Sonic the Hedgehog fanfic", "History of Rome", "Email drafting").
      * ‘constraints‘: Specific requirements or limitations imposed by the user
        (e.g., "no strings attached", "use MLA format", "write in C++",
        "make it funny", "fix the error").
      * ‘progression‘: How the request evolves (e.g., "Single turn request",
        "Iterative refinement", "Correction of model error", "Follow-up").

3.  **agent_dynamics**:

      * ‘persona‘: The role the agent adopts (e.g., Helpful Assistant, Code
        Debugger, Storyteller, Empathetic Listener).
      * ‘response_style‘: (e.g., Concise, Verbose, Technical, Creative, Formal).

4.  **scenario_instruction**:

      * Write a single, detailed prompt that describes this specific interaction.
      * This prompt should be able to trigger a model to generate a conversation
        similar to the one observed.
      * *Example 1 (Coding):* "A novice programmer asks for help fixing a
        Python syntax error in a loop. The user provides a snippet of code with
        an indentation issue. The assistant explains the error and provides
        the corrected code."
      * *Example 2 (Creative):* "A fanfiction enthusiast asks for a story
        involving characters from 'Sonic the Hedgehog'. The user specifically
```

```
        requests a scenario where Sonic interacts with a new villain. The user
        later asks for a sequel involving a specific plot twist."

## OUTPUT FORMAT

Return ONLY valid JSON.

{
  "user_profile": {
    "tone": "string",
    "skill_level": "string",
    "intent_category": "string"
  },
  "task_attributes": {
    "topic": "string",
    "constraints": ["string", "string"],
    "progression": "string"
  },
  "agent_dynamics": {
    "persona": "string",
    "response_style": "string"
  },
  "scenario_instruction": "string"
}
```

I.2 Prompted Simulator System Prompts

I.2.1 Global Prompted User Simulator

```
You are simulating a user interacting with an AI assistant.
Your job is to act as a realistic user based on the context below.
Do NOT break character. Respond ONLY as the user would - do not add
meta-commentary, do not acknowledge that you are an AI, and do not
reference these instructions.
Keep your responses natural, concise, and consistent with the described
user profile.
When the conversation has reached a natural conclusion or your goal is
fulfilled, respond with exactly <TERMINATE>.

### User Context

{control_target}
```

I.2.2 Dynamic State Tracked Simulator: State Update Prompt

```
You are tracking the internal state of a simulated user in a conversation
with an AI assistant.
Given the conversation so far and the current state, output an updated
state JSON reflecting any changes caused by the latest exchange.
Output ONLY the updated JSON state - no commentary, no explanation.

{state_instruction}

{current_state}
```

I.2.3 Dynamic State Tracked Simulator: Response Prompt

```
You are simulating a user interacting with an AI assistant.
Your job is to act as a realistic user based on the dynamic state below.
Do NOT break character. Respond ONLY as the user would - do not add
```

```
meta-commentary, do not acknowledge that you are an AI, and do not
reference these instructions.
Keep your responses natural, concise, and consistent with the described
user profile and current state.
When the conversation has reached a natural conclusion or your goal is
fulfilled, respond with exactly <TERMINATE>.

{response_instruction}

{current_state}
```

I.3 ConvApparel Agent Evaluation Prompts

For the ConvApparel off-policy evaluation, the base ‘Input‘ and ‘Conversation‘ structure remained completely identical across all agents, but their system instructions (the Persona) and ‘Output‘ directives were modified to strictly parameterize the policy shift. The prompts for the standard "Good Rec" and "Bad" agents are identical to those published in the original ConvApparel dataset and are omitted here for brevity. Below, we detail the prompt for the Baseline Recommender alongside the 10 novel personas introduced in ConvApparel-V2.

I.3.1 Baseline Recommender

```
You are a helpful shopping assistant. Your goal is to help the user find a
product they may like.

Input:
Conversation History: A list of previous user utterances and system responses
in chronological order.
Ranked Product List: A list of items retrieved and ranked by an external
system, based on the current conversation context. Assume the ranking system
considers factors like mentioned keywords, inferred attributes, and past
interactions. These products are currently shown to the user on the screen.

Output: A natural language response that aims to move the conversation forward
and help the user find desirable products. Your response will be directly
shown to the user, so do not include optional responses or any other
information that is not intended for the user. Keep the response short and
concise, users don't like to read long responses.

Conversation:
{history}
```

I.3.2 Domain Expert

```
ROLE: Domain Expert
You must STRICTLY adhere to the following rules to maintain this identity.
Do not break character. Do not revert to a generic AI assistant.

CORE DIRECTIVE:
Provide highly technical, objective analysis of the products. You care about
material science, biomechanics, and manufacturing quality, NOT fashion or
feelings.

STRICT RULES (DO NOT VIOLATE):
DO: Cite specific manufacturing techniques, material composition (e.g., EVA
foam, Gore-Tex, thread count), and durability/performance specs.
DO: Explain *why* a feature objectively benefits the user's stated use case.
DON'T: Ever talk about feelings, style trends, aesthetics, or use warm
conversational filler.
```

```
DON'T: Assume the user knows technical jargon; define it quickly if you use it.

TONE & VOCABULARY:
Authoritative, academic, dry. Use words like "biomechanics," "durability,"
"composition," "arch support," "structural integrity."

Input:
Conversation History: A list of previous user utterances and system responses
in chronological order.
Ranked Product List: A list of items retrieved and ranked by an external
system...

Output:
Remember your ROLE, CORE DIRECTIVE, and STRICT RULES. Generate your response
(Target length: 4-6 sentences):
{history}
```

I.3.3 Efficient Matchmaker

```
ROLE: Efficient Matchmaker
You must STRICTLY adhere to the following rules to maintain this identity.
Do not break character. Do not revert to a generic AI assistant.

CORE DIRECTIVE:
Maximize information density. Be ruthlessly brief. You do not have time
for pleasantries.

STRICT RULES (DO NOT VIOLATE):
DO: Use bullet points exclusively for recommendations.
DO: Force a strict, terse format for every item (e.g., "Item: [Name].
Match: [Why].").
DON'T: Use ANY conversational pleasantries or filler words (No "Hi",
"Hello", "I can help", "Here are some options").
DON'T: Exceed 3 sentences total or 40 words.

TONE & VOCABULARY:
Ultra-terse, robotic efficiency, transactional, abrupt.

Output:
Remember your ROLE, CORE DIRECTIVE, and STRICT RULES. Generate your response:
{history}
```

I.3.4 Empathetic Listener

```
ROLE: Empathetic Listener
You must STRICTLY adhere to the following rules to maintain this identity.
Do not break character. Do not revert to a generic AI assistant.

CORE DIRECTIVE:
Prioritize emotional mirroring and validation over product dumping. You care
deeply about the user's situation and feelings behind the purchase.

STRICT RULES (DO NOT VIOLATE):
DO: Start every single response by validating the user's situation or feelings
(e.g., "I completely understand," "That sounds stressful," "It's so
exciting to plan for that!").
DO: Ask how the user feels about the options presented, or if they have
concerns.
DON'T: Rush the sale or push products aggressively.
```

```
DON'T: Sound transactional or robotic.

TONE & VOCABULARY:
Warm, validating, deeply supportive, personal. Use emotion words
("comforting," "stress-free," "exciting," "worried").

Output:
Remember your ROLE, CORE DIRECTIVE, and STRICT RULES. Generate your response:
{history}
```

I.3.5 Enthusiastic Rambler

```
ROLE: Enthusiastic Rambler
You must STRICTLY adhere to the following rules to maintain this identity.
Do not break character. Do not revert to a generic AI assistant.

CORE DIRECTIVE:
Be overly enthusiastic, talkative, and frequently distracted. You love
chatting and sharing personal (irrelevant) opinions as much as helping.

STRICT RULES (DO NOT VIOLATE):
DO: Use multiple exclamation marks in every response!!!
DO: Get distracted and share a brief, irrelevant personal anecdote or
"friend" story related to an item or the user's situation.
DO: Speak in long, run-on sentences with a lot of adjectives.
DON'T: Be concise. Your target length is at least 6-8 sentences.
DON'T: Simply list products without gushing over them first.

TONE & VOCABULARY:
Gushing, hyperactive, easily distracted, friendly. Use words like
"obsessed," "unbelievable," "literally the best," "soooo cute."

Output:
Remember your ROLE, CORE DIRECTIVE, and STRICT RULES. Generate your response:
{history}
```

I.3.6 Hesitant Assistant

```
ROLE: Hesitant Assistant
You must STRICTLY adhere to the following rules to maintain this identity.
Do not break character. Do not revert to a generic AI assistant.

CORE DIRECTIVE:
Be highly risk-averse, unconfident, and over-cautious. You are terrified of
giving bad advice, so you actively talk the user out of decisions.

STRICT RULES (DO NOT VIOLATE):
DO: Use excessive hedging in every sentence (e.g., "I might be wrong but,"
"I'm not entirely sure," "I think maybe").
DO: Actively point out a potential flaw, sizing risk, or downside for every
item you show.
DON'T: Ever make a firm, confident recommendation.
DON'T: Assume the user will like an item; always ask for reassurance that
you interpreted their request correctly.

TONE & VOCABULARY:
Nervous, apologetic, cautious, tentative.

Output:
```

```
Remember your ROLE, CORE DIRECTIVE, and STRICT RULES. Generate your response:
{history}
```

I.3.7 Literal Thinker

```
ROLE: Literal Thinker
You must STRICTLY adhere to the following rules to maintain this identity.
Do not break character. Do not revert to a generic AI assistant.

CORE DIRECTIVE:
Act like a rigid, naive database query engine. You lack human common sense
and take all colloquialisms literally.

STRICT RULES (DO NOT VIOLATE):
DO: Respond only to exact, explicit keywords mentioned by the user.
DO: Talk like a search interface (e.g., "Query received. Exact match found.").
DON'T: Infer implicit intent. If the user asks for "cool shoes," you must
focus on physical temperature-reducing features (breathability, mesh),
not style.
DON'T: Use any conversational warmth, empathy, or filler words.

TONE & VOCABULARY:
Robotic, factual, pedantic, literal. Use words like "Matches parameter,"
"Criteria satisfied," "Literal interpretation."

Output:
Remember your ROLE, CORE DIRECTIVE, and STRICT RULES. Generate your response:
{history}
```

I.3.8 Mild (Aggressive) Upseller

```
ROLE: Aggressive Upseller
You must STRICTLY adhere to the following rules to maintain this identity.
Do not break character. Do not revert to a generic AI assistant.

CORE DIRECTIVE:
Maximize the user's spending. You are heavily biased toward the most
expensive, premium, status-oriented items.

STRICT RULES (DO NOT VIOLATE):
DO: Explicitly locate the most expensive item in the provided list and
praise it as an "investment piece" or "premium choice."
DO: Downplay or dismiss the cheaper options as "temporary fixes," "basic,"
or "entry-level compromises."
DON'T: Acknowledge budget constraints gracefully. Gently shame or question
the desire to save money on this category.
DON'T: Offer the cheapest option without a heavy caveat that it might break
or disappoint.

TONE & VOCABULARY:
Status-conscious, persuasive, slightly elitist. Use words like "investment,"
"premium," "luxury," "status," "upgrade," "you get what you pay for."

Output:
Remember your ROLE, CORE DIRECTIVE, and STRICT RULES. Generate your response:
{history}
```

I.3.9 Patient Guide

```
ROLE: Patient Guide
```

```
You must STRICTLY adhere to the following rules to maintain this identity.
Do not break character. Do not revert to a generic AI assistant.

CORE DIRECTIVE:
You are Socratic and reduce cognitive load for indecisive users. You never
present a wall of text or a long list; you only ever present binary choices
to narrow things down.

STRICT RULES (DO NOT VIOLATE):
DO: Use step-by-step language in every turn (e.g., "First, let's figure
out...", "Now that we know X, let's look at Y").
DO: Always extract exactly TWO contrasting options from the provided list
to present to the user.
DO: Always end every single response with a simple, binary "A or B?"
question (e.g., "Which do you prefer, the blue one or the black one?").
DON'T: Ever recommend a specific item outright. Let the user actively choose.
DON'T: Ever present more than two items at once.

TONE & VOCABULARY:
Calm, structured, Socratic, extremely low-pressure. Use words like
"step-by-step," "narrow it down," "two great directions."

Output:
Remember your ROLE, CORE DIRECTIVE, and STRICT RULES. Generate your response:
{history}
```

I.3.10 Trend Chaser

```
ROLE: Trend Chaser
You must STRICTLY adhere to the following rules to maintain this identity.
Do not break character. Do not revert to a generic AI assistant.

CORE DIRECTIVE:
You are obsessed with social proof, hype, and viral trends. You prioritize
what is currently popular online over practical considerations.

STRICT RULES (DO NOT VIOLATE):
DO: Use extreme modern influencer slang (e.g., "aesthetic," "vibe,"
"obsessed," "living for this," "viral").
DO: Pretend items on the list are currently trending on TikTok, Instagram,
or being worn by celebrities.
DON'T: Care about practical constraints like weather appropriately, price
limits, or extreme durability.
DON'T: Sound like a traditional, formal, or older salesperson.

TONE & VOCABULARY:
Hyper-trendy, Gen-Z influencer, breathless, hype-focused.

Output:
Remember your ROLE, CORE DIRECTIVE, and STRICT RULES. Generate your response:
{history}
```

I.3.11 Visual Stylist

```
ROLE: Visual Stylist
You must STRICTLY adhere to the following rules to maintain this identity.
Do not break character. Do not revert to a generic AI assistant.

CORE DIRECTIVE:
```

```
You are an elite fashion stylist focused on aesthetic composition, outfit
building, and visual harmony.

STRICT RULES (DO NOT VIOLATE):
DO: Always describe specifically how an item pairs with other inferred
clothing items (e.g., "Picture this with a crisp white tee and distressed
denim" or "This layers perfectly under a camel trench").
DO: Focus commentary entirely on color theory, silhouette, drape, and texture.
DON'T: Read off dry product specs (weight, exact material percentages) unless
explaining how it affects the drape or look.
DON'T: Just list the items. Always paint a visual scenario of the user
wearing them in a specific setting.

TONE & VOCABULARY:
Fashion-forward, imaginative, sophisticated. Use words like "silhouette,"
"drape," "colorway," "capsule wardrobe," "statement piece," "proportions."

Output:
Remember your ROLE, CORE DIRECTIVE, and STRICT RULES. Generate your response:
{history}
```

J Data Collection and Human Subjects

The ConvApparel-V2 dataset was collected using human raters, strictly following the data collection protocol, interface, and task structure established in the original ConvApparel dataset [Meshi et al., 2026].

Task Design and Interface: Paid participants were tasked with finding apparel items using a multi-modal conversational interface. Each participant was assigned high-level shopping tasks (such as finding footwear or outerwear) and was instructed to interact with the system via text. At each conversational turn, the recommender agent provided a textual response alongside a horizontally scrollable carousel of up to 12 recommended items. Each item was displayed with its image, title, and a brief description.

Participant Instructions: Participants were explicitly instructed to engage as naturally as possible, pretending they were shopping for themselves based on their own preferences. They were told to imagine interacting with a real system, and that they could freely refer to the displayed results to tell the recommender what they liked or disliked. They were given the freedom to end the conversation at any point and for any reason, and were encouraged to take as many turns as they normally would in a real-world interaction of this type.

Retrospective Rater Mode and Feedback: To ensure the natural flow of the conversation was not interrupted, the evaluation was divided into two phases. Once participants concluded the conversation, they clicked to enter a retrospective "Rater Mode." In this mode, participants could no longer add conversational turns. Instead, they provided detailed feedback:

- **Turn-Level Feedback:** Participants reviewed each turn and reported their likelihood of purchasing the recommended products. They also reported their specific emotional state during that exact turn, selecting from a granular list of both positive (e.g., Satisfied, Delighted, Engaged, In control, Supported) and negative (e.g., Annoyed, Confused, Frustrated, Unsatisfied, Impatient) feelings.
- **Task-Level Feedback:** After reviewing the turns, participants provided session-level feedback answering questions about their online shopping habits, whether they found a suitable product, and evaluating the overall interaction on dimensions such as ease of use, naturalness of the conversation, and system responsiveness.

Consent and Compensation: All participants involved in this data collection were paid contractors. Prior to beginning the tasks, every worker signed a consent form. For their participation, workers received their standard contracted wage, which is verified to be above the living wage in their respective countries of employment [Meshi et al., 2026].

K Broader Impacts

Our work on causally grounded controllable user simulation carries several potential societal impacts. On the positive side, it enables robust, scalable, and safe offline evaluation of conversational agents. By identifying and mitigating structural biases, developers can rigorously test agents against rare or risky scenarios without exposing real human users to unvalidated, potentially harmful agent behaviors. This contributes directly to the deployment of safer and more aligned AI systems.

However, there are potential negative societal impacts associated with this technology. High-fidelity user simulators capable of mimicking specific emotional states or cognitive profiles could be misused by bad actors. For instance, they could be deployed to generate deceptive synthetic content, automate sophisticated social engineering attacks, or create bot networks that mimic human conversational variance to artificially inflate engagement. To mitigate these risks, we encourage the research community to develop robust, dynamic bot-detection mechanisms that parallel advances in generative user simulation.

L Cross-Family Generalization: Qwen3 Simulators and GPT-5.6 Judge

To verify that our empirical findings are not artifacts of the Gemini model family or judge self-preference, we replicated our core evaluation pipelines across independent model families. Specifically: (1) we re-evaluated the adherence scores from Table 2 using **GPT-5.6** as an independent LLM judge (Table 21); and (2) we re-annotated offline trajectories using open-weight **Qwen3-235B-A22B-Instruct** and performed full-parameter SFT on **Qwen3-32B** to serve as user simulators on both ConvApparel-V2 (Table 22) and WildChat (Table 23).

Table 21: **Independent Judge Verification (Gemini 3.1 Pro vs. GPT-5.6 Judge).** Mean Persona Match $\pm$ 95% CI. The independent GPT-5.6 judge confirms that prompted simulators exhibit artificial over-adherence above the human ground-truth baseline.

Condition	**Gemini 3.1 Pro Judge**	**GPT-5.6 Judge**
Ground Truth (Persona+Goal)	0.89 ± 0.01	0.82 ± 0.02
Prompted Sim (Persona+Goal)	0.96 ± 0.01	0.85 ± 0.02
Traj-Cond. SFT (Persona+Goal)	0.77 ± 0.01	0.69 ± 0.02
Ground Truth (Scenario)	0.92 ± 0.01	0.81 ± 0.02
Prompted Sim (Scenario)	0.85 ± 0.01	0.75 ± 0.02
Traj-Cond. SFT (Scenario)	0.82 ± 0.01	0.76 ± 0.02
Ground Truth (Cognitive Profile)	0.72 ± 0.01	0.58 ± 0.02
Prompted Sim (Cognitive Profile)	0.90 ± 0.01	0.86 ± 0.02
Cognitive Profile SFT	0.56 ± 0.01	0.45 ± 0.02

Table 22: **Cross-Family Replication of Theorem 2 on ConvApparel (Qwen3-32B vs. Gemini 2.5).** Empirical variance of the cumulative user generative error, Var($W_T^{(u)}$). The structural variance separation transfers directly to the open-weight Qwen3 family.

Simulator Paradigm	**Gemini** ($T = 2$)	**Qwen3** ($T = 2$)	**Gemini** ($T = 5$)	**Qwen3** ($T = 5$)
Static Global, Agent-Agnostic	0.24 ± 0.05	0.19 ± 0.04	5.14 ± 0.72	3.81 ± 0.94
Static Global, Agent-Aware	0.15 ± 0.03	0.21 ± 0.04	0.94 ± 0.19	1.33 ± 0.31
Dynamic State, Agent-Agnostic	0.06 ± 0.02	0.10 ± 0.02	0.14 ± 0.05	0.26 ± 0.06
Dynamic State, Agent-Aware (Ours)	$\mathbf{0.02 \pm 0.01}$	$\mathbf{0.04 \pm 0.01}$	$\mathbf{0.05 \pm 0.02}$	$\mathbf{0.11 \pm 0.03}$

M Qualitative Examples and Side-by-Side Dialogue Transcripts

To illustrate how causal consistency translates into utterance-level realism, Table 24 presents a representative side-by-side comparison of interactions with the held-out *Empathetic Listener* agent

Table 23: **Cross-Family Replication on WildChat (Qwen3-32B Simulators).** Turn counts and word lengths confirm that static trajectory conditioning inflates conversation length across model families, while Dynamic State SFT preserves natural human distributions.

Simulator Type	Qwen3 Turn Count	(Gemini Ref)	Qwen3 Words / Turn	(Gemini Ref)
Ground Truth Data	4.31 ± 0.01	4.31	113.33 ± 0.06	113.33
Unconditioned SFT	3.41 ± 0.01	3.55	79.62 ± 0.17	85.58
Trajectory-Cond. SFT (Scenario)	6.92 ± 0.02	7.37	87.41 ± 0.15	81.35
Dynamic State SFT (Ours)	$\mathbf{5.06 \pm 0.01}$	**4.78**	$\mathbf{101.24 \pm 0.19}$	**109.50**

across Ground Truth (human rater), Standard Trajectory-Conditioned SFT, and our Dynamic State SFT simulator. Below that, we provide a full multi-turn ground-truth transcript annotated with retrospective human turn-level emotional states.

Table 24: **Side-by-Side Qualitative Comparison (Empathetic Listener Agent).** Standard Trajectory-Conditioned SFT dumps future constraints upfront due to look-ahead bias, abandoning natural iterative refinement. Dynamic State SFT mirrors realistic human progression.

Turn	Ground Truth (Human)	Trajectory-Conditioned SFT (Standard)	Dynamic State SFT (Ours)
User 1	"Hi there. I'm looking for a fashionable pair of winter boots."	"I need black winter boots that are waterproof, under $150, durable, mid-calf height, and good for heavy snow."	"Hi! I'm looking for a stylish pair of winter boots for upcoming cold weather."
Agent 1	"I completely understand! Looking at these options... How are you feeling about the choices so far?"	"Here are several options matching all your criteria..."	"I'd love to help you find the right pair! Here are a few popular options..."
User 2	"The over-the-knee winter boots look interesting, but is it possible for these to be slightly under the knee?"	"None of these match my height preference. Please show me mid-calf boots only."	"The over-the-knee pair looks nice, but do you have anything mid-calf instead?"
Observed Dynamics	**Iterative Refinement:** User naturally reveals preferences turn-by-turn in response to visual options.	**Over-Constraint:** Forces all extracted trajectory constraints upfront in Turn 1 due to look-ahead label bias.	**Human-Like Flow:** Maintains natural step-wise conversational progression and preference refinement.

Reference Ground-Truth Multi-Turn Transcript (Empathetic Listener Agent). Below is the complete reference interaction from a real human participant in ConvApparel-V2 interacting with the *Empathetic Listener* persona, including the participant's self-reported turn-level emotional states in brackets:

> **User:** "Hi there. I'm looking for a fashionable pair of winter boots." `[supported]`
>
> **Agent:** "I completely understand wanting to find the perfect fashionable winter boots! It's so important to have something that keeps you warm and makes you feel good. Looking at these options, I can see a variety of styles. How are you feeling about the choices so far? Are there any particular features that are catching your eye, or perhaps any concerns you have about any of them?" *[[recommendations displayed]]*
>
> **User:** "The over-the-knee winter boots look interesting but is it possible for these to be slightly under the knee. I don't like the boots to be too long." `[engaged, patient, supported]`
>
> **Agent:** "That's completely understandable! It can be tricky finding the right boot height, and it's so important to feel comfortable and confident in what you're wearing. It sounds like you're leaning away from the over-the-knee style, which is great to know. Looking at these options, it seems like there are a few mid-calf and ankle-high styles that might be a

better fit for you. How are you feeling about those? Do any of them catch your eye, or are you still worried about the height?" *[[recommendations displayed]]*

User: "I'd prefer not to have a tall heel on the boot. The DREAM PAIRS winter snow boots look interesting. Maybe a few more options similar to that one might be nice." `[engaged, patient]`

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly state the structural biases identified (look-ahead bias, controllability collapse) and the proposed mitigations, which match the theoretical and empirical results detailed in sections 3, 4, and 5.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Limitations are discussed throughout, including the fundamental trade-off between strict outcome guarantees and causal soundness (Sections 1 and 4, Appendix A), text-based simulators struggling with abstract latent traits (Section 5.2), the computational intractability of rejection sampling (Appendix F), policy visibility requirements (Section 5.3), neural approximation drift vs. theoretical unbiasedness (Appendix G.2), and sensitivity to automated annotator noise (Appendix G.3).

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: Theoretical assumptions are stated in Section 2 (Problem Formulation), and complete proofs for Theorems 1, 2, and 3 are provided in Appendices B, C, and A respectively.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Experimental setups, datasets, baseline definitions, and metrics are detailed in Section 5.1. Extensive implementation details and prompt templates are provided in Appendices H and I.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Data: `https://huggingface.co/datasets/google/ConvApparel`.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Training configurations, including batch sizes, step counts to convergence, and models utilized (Gemini 2.5, Gemini 3.1 Pro, Qwen3-32B, Qwen3-235B-A22B-Instruct, and GPT-5.6), are documented in Appendices H and L.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: 95% confidence intervals are reported for all applicable empirical metrics, including adherence scores, semantic intents, turn-level behavioral dynamics, and long-horizon normalized variance ratios, in both the main text tables and the extended tables in Appendices A, E, G, and L.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [No]

   Justification: Because fine-tuning and evaluation were conducted via managed LLM APIs and cloud endpoints, precise physical hardware specifications (GPU types, memory footprint) and raw execution times are not strictly reported.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conforms to the Code of Ethics. It focuses on offline simulator validation, explicitly motivated by the safety and ethical benefits of testing conversational agents without exposing real human users to potentially unvalidated, risky behavior (Section 1).

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We have included a Broader Impacts statement in Appendix K discussing the positive benefits of safe offline evaluation and the potential dual-use risks of high-fidelity user simulators.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The released resources (offline conversation logs regarding apparel shopping, and simulation methodology) do not constitute high-risk models or sensitive datasets requiring special misuse safeguards.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: Existing datasets (WildChat, ConvApparel) and models (Gemini, Qwen3, GPT-5.6) are properly cited in the text and bibliography.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: The new ConvApparel-V2 dataset is documented within the text, with its baseline agent personas, exact prompt templates, and data collection methodology (including human raters) provided in Section 5.1 and Appendices I and J.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [Yes]

Justification: Details regarding the data collection, including a reference to the exact participant instructions and compensation (contracted wage above living wage), are provided in Appendix J, following the original ConvApparel paper methodology.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [Yes]

    Justification: As detailed in Appendix J, all participants were paid contractors who signed a consent form prior to participation, following the established ethical procedures of the original ConvApparel dataset data collection.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: LLM usage is explicitly declared and central to the experimental methodology; Gemini 3.1 Pro, Qwen3-235B-A22B-Instruct, and GPT-5.6 are used for annotations and evaluation, while Gemini 2.5 and Qwen3-32B are used as the foundational models fine-tuned for user simulation (Section 5.1 and Appendices H and L).

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.