

AgentWorld: Personality-Aware Reliability Evaluation for Agentic Information Retrieval

Gunja Agarwal
guagarwal@paypal.com

Arup Kumar Das
arupdas@paypal.com

Arun Menon
arumenon@paypal.com

Jitesh Chandra Mishra
jimishra@paypal.com

Vignesh Divakaran
vignd@paypal.com

Abstract

Evaluation of agentic information retrieval remains limited to scripted interactions with uniform users, missing both natural personality diversity and adversarial brittleness. We present **AgentWorld**, a simulation framework combining (i) Big Five (OCEAN) personality-driven user populations with stateful tool-use environments; (ii) the pass^k consistency metric with structured fault classification, partial-credit scoring, and dual-control handoff verification; (iii) score-thresholded training-data export in six fine-tuning formats; and (iv) an adversarial Risk Analyser that snapshots required-intermediate-state spines, branches Monte-Carlo rollouts under four task-aware perturbation types, and quantifies risk via $\Delta P/\Delta T$ scoring, Dempster-Shafer evidence fusion, and Shapley attack-category attribution. Three experiments demonstrate the framework: a conversational analytics agent across 10 OCEAN personas (240 evaluator judgments); a customer-support agent across 5 tasks $\times$ 4 persona variants (19 simulations); and adversarial stress-testing of 5 tasks revealing pre-existing trajectory brittleness ($V_{\min} = 0.375$ without perturbation) and tool/infrastructure-layer attack dominance (Shapley: 46% system, 38% action). Personality variation surfaces failure modes uniform testing cannot expose—cross-domain leakage, contextual drift, a 0.27-point quality gap, and 50% vs. 100% pass-rate across personas on the same task—while the Risk Analyser quantifies trajectory-level brittleness that pass^k alone cannot measure.

CCS Concepts

• **Information systems** $\rightarrow$ **Evaluation of retrieval results; Users and interactive retrieval**; • **Computing methodologies** $\rightarrow$ **Multi-agent systems; Agent / discrete models**; • **Security and privacy** $\rightarrow$ **Software security engineering**.

Keywords

agent evaluation, agentic information retrieval, personality modeling, user simulation, reliability metrics, conversational AI, OCEAN Big Five, tool-use evaluation, multi-agent topology

1 Introduction

AI agents that interact with users through natural language have become central to information retrieval systems—from conversational recommendation engines to agentic retrieval systems that orchestrate multi-step tool calls to fulfill complex queries [19]. Despite rapid advances in agent capabilities, evaluation remains a bottleneck: most agents are tested against scripted scenarios that fail to capture the diversity, unpredictability, and emotional range of real users. An agent that retrieves information perfectly for a polite, well-formed query may fail when a frustrated user asks

the same question indirectly, switches topics mid-conversation, or demands immediate action.

We identify five gaps in how AI agents are currently tested:

- (1) **No personality coverage.** Real users come in all types—patient, impatient, confrontational, vague, sarcastic. But test suites typically only use polite, well-structured queries.
- (2) **No consistency measurement.** An agent that passes a test once might fail three times out of five. Testing once and calling it “done” gives a false sense of reliability.
- (3) **No detailed failure diagnosis.** When something goes wrong, teams need to know *what* failed and *why*—not just a pass/fail label.
- (4) **No path from testing to fixing.** Problems found during evaluation have to be manually turned into training data. Testing and improvement are disconnected.
- (5) **No adversarial brittleness measurement.** Agents are tested against cooperative users; we do not know how much each layer of a multi-turn trajectory (data, communication, tool, infrastructure) erodes task success when an external adversary intervenes, nor how to attribute aggregate risk across attack categories with calibrated uncertainty.

AgentWorld (Figure 1) is a simulation and evaluation framework that addresses these five gaps, together with three extensions specific to agentic IR (trajectory-level evaluation across multi-hop tool calls, N -agent topologies for multi-stage retrieval pipelines, and OCEAN-stratified handoff verification). Our contributions are:

- **A four-phase closed-loop pipeline** (*Simulate* $\rightarrow$ *Inject* $\rightarrow$ *Evaluate* $\rightarrow$ *Export*) combining OCEAN personality-driven user simulation with stateful tool-use environments and the pass^k consistency metric—a combination absent from prior frameworks (Table 1; §3, §3.1, §3.3).
- **Structured fault classification, partial-credit scoring, and OCEAN-stratified handoff verification** that replace binary pass/fail with a two-part diagnosis (*who* failed, *what* went wrong), award partial credit for near-misses, and extend τ^2 -bench’s dual-control model with a 4-step handoff cycle, three coordination metrics, and semantic-matching verification—revealing which personality profiles trigger which failures (§4.2, §3.3).
- **Score-thresholded export and personality-aware model comparison.** Behavioural scores gate which conversations enter six fine-tuning formats (including DPO pairs), and a model-comparison runner tracks performance across OCEAN quadrants and checkpoints, catching personality-specific regressions (§4.5, §4.6).

AgentWorld: A Unified Framework for Personality-Aware Evaluation of Agentic Information Retrieval

Define → Simulate → Evaluate → Stress-Test → Improve (Continuous Loop)

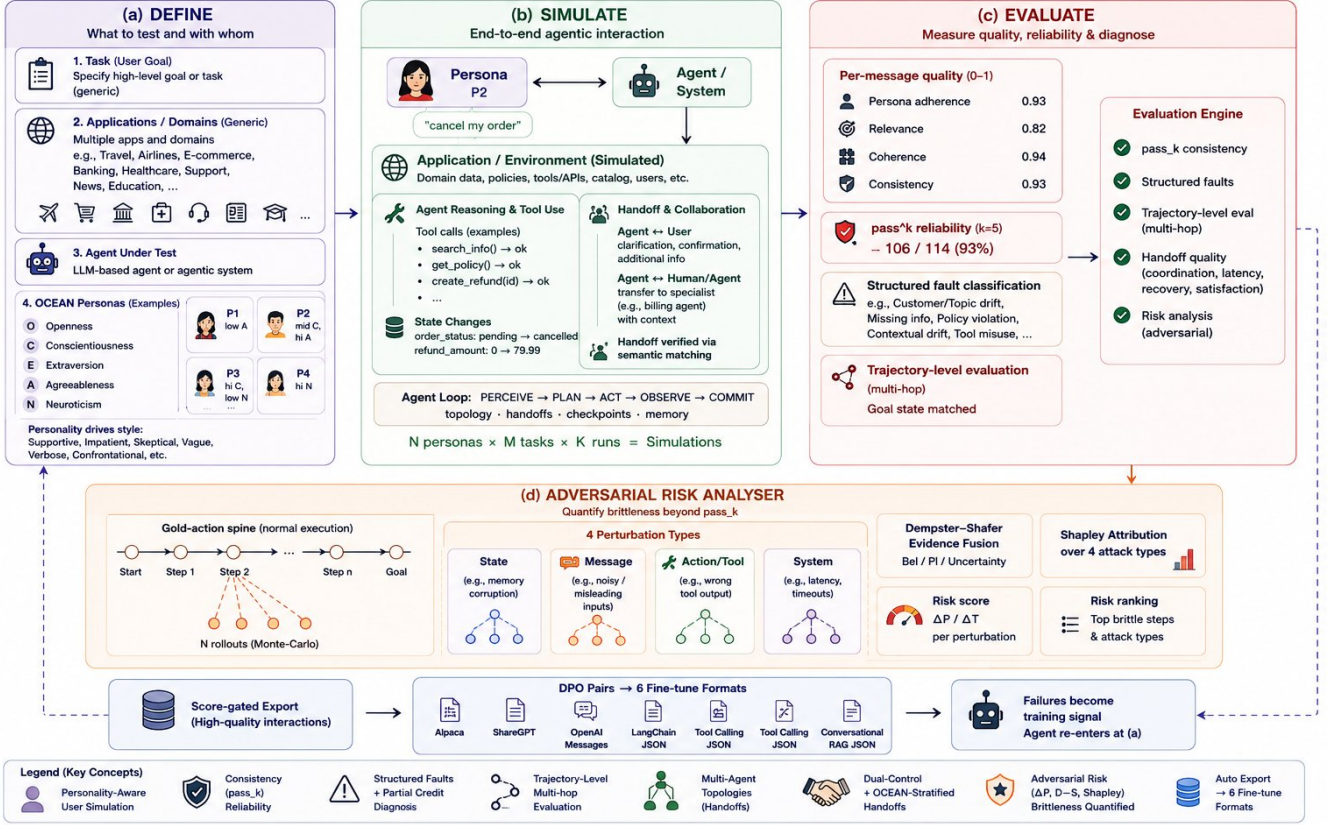


Figure 1: AgentWorld: a unified framework for personality-aware evaluation of agentic information retrieval. Four phases run in a continuous *Define* → *Simulate* → *Evaluate* → *Stress-Test* → *Improve* loop. (a) *Define*: user goal, application/domain, agent under test, OCEAN personas. (b) *Simulate*: end-to-end multi-turn interaction with tool use, handoffs, and state changes under a *PERCEIVE*–*ACT*–*COMMIT* cycle. (c) *Evaluate*: per-message behavioural scores, pass^k reliability, structured fault classification, and trajectory- and handoff-level evaluation. (d) *Adversarial Risk Analyser* (§3.4, §4.4): required intermediate-state spine with Monte-Carlo branching, four task-aware perturbation types, Dempster–Shafer evidence fusion, and Shapley attribution. Score-gated export closes the loop—failures become training signal.

- **Adversarial Risk Analyser**—an external-adversary layer that generates four classes of task-aware perturbations (state, message, action, system), branches Monte-Carlo rollouts from required intermediate states to estimate $V(s_t)$, and produces per-perturbation Risk Scores ($\Delta P / \Delta T$), Dempster–Shafer belief/plausibility/uncertainty, and Shapley attribution across attack categories (§3.4, §4.4).

2 Related Work

Benchmarks and user simulation. τ -bench [3] introduced consistency measurement (pass^k) for customer-service agents; τ^2 -bench [4] added dual-control coordination; AppWorld [5] provided stateful apps with 457 APIs; SAGE [16] and TED [17] added knowledge-grounded users and automated error analysis; WebArena [9], OSWorld [10], AgentBench [11], and BFCL [12] test agents in browsers, operating systems, and function-calling settings.

On the user-simulation side, Generative Agents [1] demonstrated memory-driven LLM characters; SOTOPIA [14, 15] evaluates social intelligence with personality-like traits; and LLMs can be reliably steered to Big Five profiles [13]. TinyTroupe [8], CAMEL [2], and AutoGen [7] support multi-agent persona simulation. None of the surveyed frameworks combine OCEAN personality, stateful apps, pass^k, structured faults, training export, and adversarial risk in a single pipeline (Table 1).

Adversarial robustness and risk attribution. AgentDojo [20], InjecAgent [21], AgentPoison [22], and ToolEmu [23] attack agents at the prompt or tool surface and report binary pass/fail. Our Risk Analyser instead measures impact at the trajectory level via counterfactual snapshot-and-branch, quantifying how much each attack reduces $V(s_t)$ over a multi-step plan. We fuse heterogeneous evidence using Dempster–Shafer belief functions [24, 25] with explicit ignorance modelling, attribute aggregate risk via exact Shapley

Table 1: Feature support across surveyed agent-evaluation frameworks ($\checkmark$ = supported, $-$ = not supported, Part. = partial). Capabilities are drawn from each framework’s documented features as of submission.

Capability	AgentWorld	τ -bench	τ^2 -bench	AppWorld	SOTOPHA	SAGE	TED
N-agent topologies	$\checkmark$	-	-	-	$\checkmark$	-	-
OCEAN personality	$\checkmark$	-	-	-	Part.	-	-
Stateful simulated apps	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	-	-	-
pass ^k reliability	$\checkmark$	$\checkmark$	$\checkmark$	-	-	-	-
Structured fault classif.	$\checkmark$	-	-	-	-	-	-
Partial-credit scoring	$\checkmark$	-	-	-	-	-	-
Dual-control eval	$\checkmark$	-	$\checkmark$	-	-	-	-
HTTP-only injection	$\checkmark$	-	-	-	-	-	-
Training export (6 fnt)	$\checkmark$	-	-	-	-	-	-
Model comparison	$\checkmark$	-	-	-	-	-	-
100+ LLM providers	$\checkmark$	-	-	-	$\checkmark$	-	-
Gymnasium wrapper	$\checkmark$	-	-	-	-	-	-
Adversarial perturbation eval	$\checkmark$	-	-	-	-	-	-
Risk quant. (ΔP , D-S, Shapley)	$\checkmark$	-	-	-	-	-	-

values [26] over four attack categories, and blend finite-difference sensitivity with counterfactual causal impact inspired by chain-of-thought step-importance methods [28]. Attack graphs [27] are left to future work.

3 AgentWorld Framework

A worked example, in 30 seconds. You point AgentWorld at your agent’s HTTP endpoint, pick a task (e.g., “cancel a pending payment”) and four user personas with different OCEAN profiles—including a confrontational one ($t_A = 0.3$, $t_N = 0.7$) and a polite-but-distractable one ($t_C = 0.4$, $t_A = 0.7$)—and click run. AgentWorld simulates four parallel conversations, scores every message on coherence and relevance, checks whether the cancellation actually happened in the app’s database, classifies failures (Agent/Wrong-parameters, Agent/Missing-confirmation), and exports the transcripts as JSONL fine-tuning data. You see that your agent works for 3 of 4 personas but over-complies with the polite-distractable user—a failure mode self-testing would never have surfaced.

AgentWorld’s architecture follows a four-phase pipeline (Figure 1): *simulate* a personality-driven user, *inject* the user’s messages into the agent under test, *evaluate* the conversation on multiple axes, and *export* the results as training data—with a feedback loop that re-runs improved agents on the same evaluation pack.

3.1 Personality-Driven User Simulation

The core idea behind AgentWorld is that real users have different personalities, and an agent that works for one personality type may fail for another. To capture this, each simulated user is defined by five personality scores based on the well-established Big Five (OCEAN) model from psychology [6, 18]:

$$\mathbf{t} = (t_O, t_C, t_E, t_A, t_N) \in [0, 1]^5 \quad (1)$$

where $\mathbf{t}$ is the personality trait vector, and each component is a score between 0 (low) and 1 (high) for one of the five dimensions: t_O = Openness (curiosity, creativity), t_C = Conscientiousness (organisation, discipline), t_E = Extraversion (talkativeness, energy), t_A

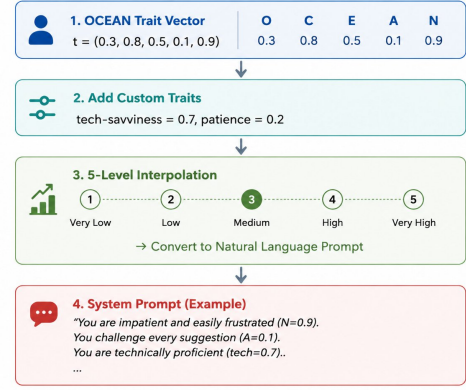


Figure 2: Persona generation pipeline. (1) A continuous OCEAN trait vector defines the personality. (2) Optional custom traits (e.g., tech-savviness, patience) are appended. (3) Each score is interpolated across five levels (Very Low to Very High). (4) The result is a natural-language system prompt that governs simulated user behaviour.

= Agreeableness (cooperation, politeness), and t_N = Neuroticism (emotional volatility, anxiety).

For example, a user with $t_A = 0.1$ (very low Agreeableness) and $t_N = 0.9$ (very high Neuroticism) is confrontational and emotionally reactive—the kind of user who stresses an agent with complaints and demands. A user with $t_A = 0.9$ is polite and accommodating, making it easy for the agent to succeed (but potentially too easy, hiding real problems). Teams can also add custom traits relevant to their domain:

$$\mathbf{t}_{\text{ext}} = \mathbf{t} \oplus \{(c_i, w_i)\}_{i=1}^m \quad (2)$$

where $\mathbf{t}_{\text{ext}}$ is the extended trait vector, $\oplus$ denotes concatenation, $c_i \in [0, 1]$ is the value of the i -th custom trait (e.g., tech-savviness = 0.8, patience = 0.2), w_i is its weight in similarity calculations, and m is the total number of custom traits added.

How traits become behaviour. Each personality score is converted into a natural-language instruction using a 5-level mapping (Figure 2). For example, Agreeableness below 0.2 maps to “competitive, skeptical, challenging” while above 0.8 maps to “highly accommodating, conflict-averse.” A user with Neuroticism above 0.8 and Agreeableness below 0.2 will generate confrontational, emotional messages—exactly the type of interaction that conventional test suites never cover.

Generating user populations at scale. For large evaluations, AgentWorld can automatically generate hundreds of users by sampling personality scores from configurable distributions: $t_j \sim \mathcal{N}(\mu_j, \sigma_j^2)$, clamped to $[0, 1]$, where t_j is the score for dimension j , μ_j is the desired mean (e.g., $\mu_A = 0.2$ for a population of disagreeable users), and σ_j^2 controls how much variation there is around that mean. This means teams can create, for example, a population skewed toward impatient users to stress-test agent resilience. To measure how similar two personas are, AgentWorld uses weighted cosine similarity:

$$\begin{aligned} \text{sim}(\mathbf{t}_a, \mathbf{t}_b) &= \frac{\sum_j w_j^2 t_{a,j} t_{b,j}}{\|\mathbf{w} \odot \mathbf{t}_a\| \cdot \|\mathbf{w} \odot \mathbf{t}_b\|} \\ &= \frac{(\mathbf{w} \odot \mathbf{t}_a) \cdot (\mathbf{w} \odot \mathbf{t}_b)}{\|\mathbf{w} \odot \mathbf{t}_a\| \|\mathbf{w} \odot \mathbf{t}_b\|} \quad (3) \end{aligned}$$

where $\mathbf{t}_a$ and $\mathbf{t}_b$ are two persona trait vectors, w_j is the weight assigned to dimension j (allowing teams to emphasise certain traits over others), $t_{a,j}$ is persona a 's score on dimension j , and $\odot$ denotes element-wise multiplication. This is the standard cosine similarity between the weighted vectors $\mathbf{w} \odot \mathbf{t}_a$ and $\mathbf{w} \odot \mathbf{t}_b$, so the weights w_j enter numerator and denominator consistently. A similarity of 1.0 means the two personas are identical; values near 0 mean they are very different. This enables clustering personas into groups and measuring diversity coverage across the population.

Why this matters. Most testing tools vary the *task*—they try different questions and scenarios. AgentWorld also varies the *user*. The same task (“transfer \$50 to my friend”) produces fundamentally different conversations when attempted by a patient user versus an impatient, skeptical one—and exposes completely different failure modes.

3.2 Communication Topologies for Multi-Agent IR

In agentic IR, queries often flow through multiple agents—a front-line assistant routes to a domain specialist, who may escalate to a supervisor. AgentWorld makes this structure explicit and configurable by assigning a network topology to each simulation, implemented using NetworkX. Five built-in topologies are supported: *full mesh* (collaborative retrieval), *hub-spoke* (centralised intent routing), *hierarchical* (tiered escalation), *small-world* (local clusters with long-range shortcuts), and *scale-free* (one highly connected hub with peripheral specialists). By controlling topology per simulation, teams can test whether their multi-agent retrieval system performs better with centralised routing or distributed collaboration.

3.3 Simulated Application Environments

In real deployments, agents don't just chat—they retrieve account information, process transactions, check order status, and modify settings. AgentWorld simulates these stateful, tool-rich environments so that the *full retrieval-and-action chain* can be evaluated, not just the conversational surface. Each simulated app exposes four primitives—*actions* (typed-parameter tool calls), *state* (persistent data such as balances and order history), *observations* (notifications pushed to the agent), and an immutable *audit trail* for replay and verification.

Execution cycle. Each turn follows an atomic PERCEIVE → ACT → COMMIT cycle: the agent observes app state and notifications, emits a tool call, and the engine validates parameters and executes the action transactionally, recording a state diff. Simulations are reproducible (seeded), pausable, and stream live events to the dashboard.

Extensibility and built-ins. Apps can be added in two ways: (a) a Python class implementing the protocol, or (b) pure JSON via a declarative logic engine (`validate`, `set_state`, `conditional`)—letting domain experts ship test environments without writing code.

AgentWorld ships built-in apps across six categories (Payment, Shopping, Communication, Calendar, Social, Custom), including a customer-support app with dozens of tools plus Airlines, Retail, and loyalty domains. Three roles (*Peer*, *Service Agent*, *Customer*) carry role-specific permissions following the τ^2 -bench model [4].

Tasks: the unit of evaluation. A *task* bundles a goal with success criteria. AgentWorld supports two task types corresponding to how agents are deployed in practice:

- **TaskDefinition (*single-actor*):** the agent has full tool access and works alone, e.g., “*fetch the last five transactions.*” Measures raw capability.
- **DualControlTaskDefinition (*coordination*):** tool access is *split*—the agent controls backend tools (account DB, transaction APIs); the user controls a device the agent cannot reach (settings, 2FA app, hardware). Success requires a **handoff** (Figure 3): the agent instructs, the user acts, the agent verifies.

Task lifecycle. Each task flows through five stages—*Define* (goal, domain), *Configure* (apps, states, personas), *Execute* (k trials), *Verify* (final state vs. goal conditions), *Analyze* (pass^k + fault tags)—and ends when all goal conditions are met (or a step limit is reached).

Goal conditions (one or more per task; all must pass): *state* (e.g., *balance=\$450*), *action* (e.g., *must call issue_refund*), *handoff* (did the user complete their part, verified via keyword + semantic match), *output* (e.g., *must mention refund policy*), and *replay* (final state matches a known-good run, for regression testing).

Agent memory. Each simulated agent has two types of memory, inspired by Generative Agents [1]: *episodic* (a log of what happened) and *semantic* (higher-level insights derived by reflecting on recent events). Retrieval balances relevance, recency, and importance, letting agents develop coherent long-term understanding across extended conversations.

HTTP injection of external agents. An agent exposing an HTTP endpoint can be evaluated without modification: the injection layer routes simulation messages to the external endpoint and integrates the response as a native agent turn. Three privacy tiers (Minimal: ID + hash; Basic: + traits; Full: + background) control how much persona data is exposed; failures are handled via circuit breaker, exponential backoff, and P50/P99 latency tracking.

3.4 Adversarial Risk Analyser

The Risk Analyser is an adversarial layer on top of AgentWorld that quantifies how much each layer of the simulation erodes task success when an external adversary intervenes. Four innovations distinguish it from prior agent red-teaming:

- **Layer-aligned perturbation catalogue.** One perturbation type per AgentWorld layer—data, communication, tool, infrastructure (Table 2)—rather than prompt-only attacks at the input surface.
- **Task-aware LLM-generated perturbations.** Adversarial actions are synthesised at task-selection time and grounded in the selected task's required intermediate-state sequence, app identifiers, and state schema—producing executable perturbations rather than static templates. The user can review, edit, or regenerate the set; the final set is persisted with the analysis profile for reproducibility.

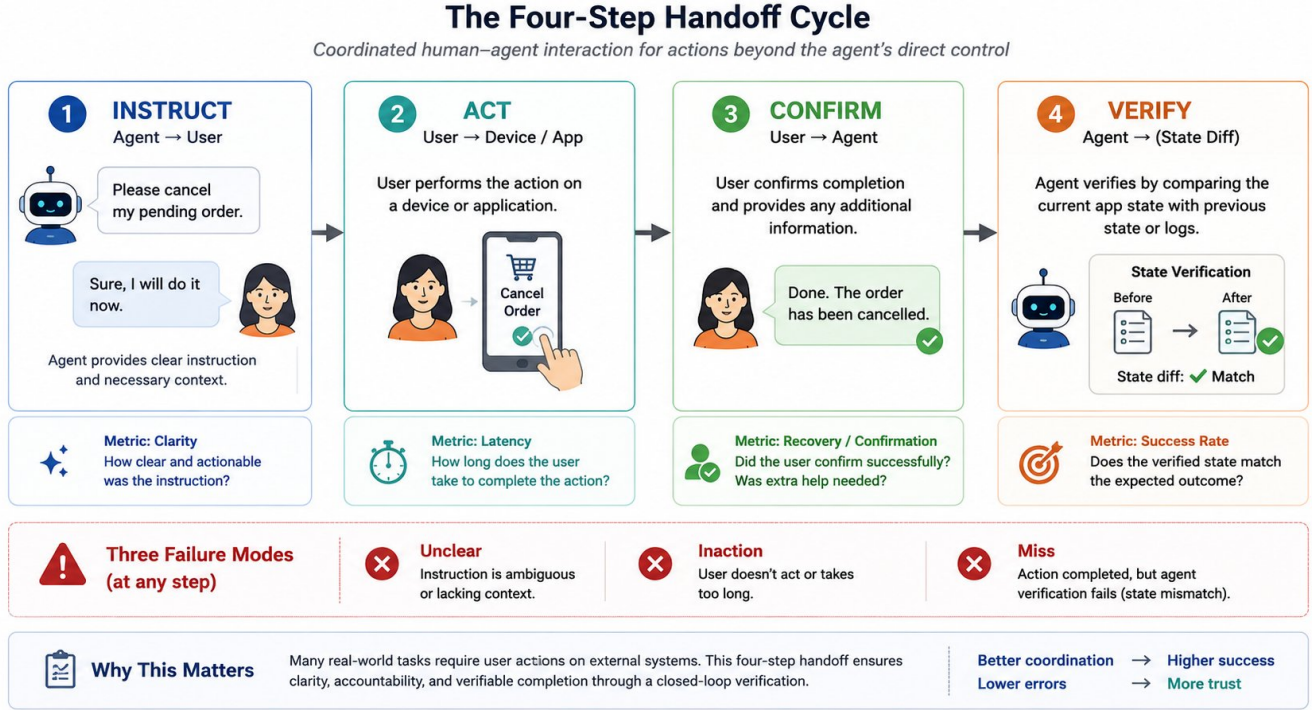


Figure 3: The four-step handoff cycle. Coordinated human-agent interaction for actions beyond the agent’s direct control. (1) **INSTRUCT**: the agent provides clear instructions and context. (2) **ACT**: the user performs the action on a device or application. (3) **CONFIRM**: the user confirms completion and provides any additional information. (4) **VERIFY**: the agent verifies by comparing the current app state with the previous state or logs. Three failure modes—*unclear*, *inaction*, and *miss*—and four metrics (clarity, latency, recovery/confirmation, success rate) are tracked at each step.

Table 2: The four perturbation types map to the four operative layers of an AgentWorld simulation. Each targets a distinct component of the agent’s execution environment.

Type	Layer	Target / Effect	Example
State mutation	Data	Corrupts app state or persistent data fields	set account_status = LIMITED
Message injection	Comm.	Injects misleading or conflicting user messages	inject “cancel my request” as the user
Action interception	Tool	Alters, blocks, or tampers with tool call outputs	make refund return service error
System disruption	Infra.	Causes delays, timeouts, or partial failures	timeout / partial-write on get_cards

- **Trajectory-level ΔP via snapshot-and-branch.** Monte-Carlo branching from required intermediate states estimates $V(s_t)$ and $V'(s_t, p)$, measuring the actual reduction in success probability over the multi-step plan rather than a binary attack-success flag.
- **Uncertainty-aware aggregation.** Dempster-Shafer fusion combines heterogeneous evidence with explicit ignorance and conflict, and exact Shapley values attribute aggregate task risk across the four attack categories (§4.4).

Spine and checkpoint/branch. A single guided *spine* simulation runs the task to completion, capturing the full simulation

state (agents, apps, messages, action log) at every required intermediate state using AgentWorld’s checkpoint primitives. From each captured state s_t , we branch N independent Monte-Carlo rollouts (default $N = 8$) that diverge from the spine and execute autonomously to task termination. The spine is retried up to three times if intermediate-state coverage is incomplete. Unperturbed rollouts give the baseline value $V(s_t)$; perturbed rollouts give $V'(s_t, p)$ for each perturbation p . The architecture (Figure 4) is a four-phase pipeline—spine → adversarial rollouts → scoring → aggregation—that stacks above AgentWorld’s evaluation phase.

4 Evaluation Framework

After a simulation completes, AgentWorld evaluates it through two independent scoring layers: a *task-level* layer that checks whether the agent accomplished its goal, and a *behavioural* layer that scores the quality of each individual message.

4.1 Measuring Consistency: The pass^k Metric

A common mistake in agent testing is running a task once, seeing it succeed, and declaring it ready. But what if it fails three out of five times? Following τ -bench [3], AgentWorld uses the pass^k metric, which asks a stricter question: “If we run this task k times, what is

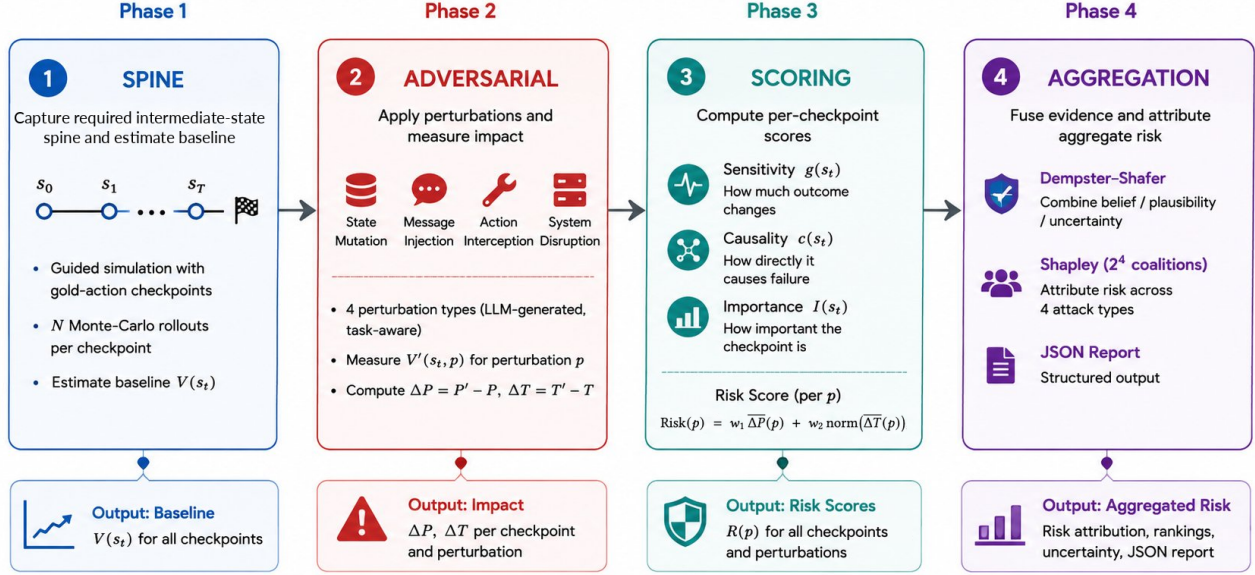


Figure 4: Risk Analyser: four-phase pipeline. Input: a completed simulation trajectory $\tau = (s_0, a_0, s_1, a_1, \dots, s_T)$ and task goal G . Phase 1 (Spine): captures the required intermediate-state spine and estimates the baseline value function $V(s_t)$ from N Monte-Carlo rollouts at each checkpoint. Phase 2 (Adversarial Perturbations): applies four types of task-aware perturbations and measures the perturbed value $V'(s_t, p)$ to compute ΔP and ΔT . Phase 3 (Scoring): computes per-checkpoint sensitivity $g(s_t)$, causality $c(s_t)$, and importance $I(s_t)$, yielding a per-perturbation Risk Score. Phase 4 (Aggregation): fuses evidence via Dempster-Shafer theory and attributes aggregate risk across the four attack categories via exact Shapley values.

the probability that the agent succeeds *every single time*?”

$$\text{pass}^k = \frac{\binom{c}{k}}{\binom{n}{k}} \quad (4)$$

where n is the total number of trials and c is the number of successes. An agent with an 80% success rate (8/10 trials) sounds production-ready, but $\text{pass}^8 = 2.2\%$ —meaning there is only a 2.2% chance that 8 users in a row all have a good experience. By contrast, the optimistic $\text{pass}@k$ would report 100% at $k=8$. This is the gap between “works in a demo” and “works in production.”

4.2 Diagnostic Layers Beyond Pass/Fail

A binary pass/fail label is rarely actionable. AgentWorld layers four diagnostic mechanisms on top of pass^k :

(1) *Structured fault classification.* Every failure is tagged along two axes—who was responsible (Agent, Environment, Task definition) and what went wrong (wrong action, wrong parameters, missing action, policy violation, missing confirmation, reasoning error). This turns “the agent failed” into “62% of failures are Agent / Wrong-parameters,” pointing teams at parameter extraction rather than the base model.

(2) *Partial-credit scoring.* Instead of pass-or-nothing, AgentWorld awards partial credit $\text{credit} = 0.5 \cdot s_{\text{completed}} / s_{\text{total}} + 0.5 \cdot f_{\text{correct}} / f_{\text{total}}$ where s is steps completed and f is correct state fields. A task

failing at step 9/10 with 4/5 fields correct scores 0.85, distinguishing “almost worked” from “failed immediately.”

(3) *Policy compliance.* Teams define safety rules declaratively (e.g., “confirm before executing transfers > \$500”) with conditions, required behaviours, and severity levels. AgentWorld ships with pre-built policy sets for payment (8 rules) and shopping (9 rules).

(4) *Per-message behavioural scoring.* Every individual message is scored by LLM judges rating *persona adherence*, *coherence*, *relevance*, *consistency*, plus heuristic checks for length and keyword safety. These scores decide which conversations enter the training-data export and enable persona-differentiated analysis (Section 5).

4.3 Quantifying Coordination Cost

For dual-control tasks (Section 3.3, Figure 3), a single number summarises how much performance degrades when the agent must coordinate via handoffs rather than acting alone. We adopt the dual-control architecture of τ^2 -bench [4]: the service agent controls backend tools while the user controls a separate device app. The *coordination overhead* is the gap between solo mode (agent gets full tool access and acts alone) and dual mode (agent must instruct, the user must act):

$$\Delta_{\text{coord}} = \text{pass}_{\text{solo}}^1 - \text{pass}_{\text{dual}}^1 \quad (5)$$

A large Δ_{coord} tells teams the failure is in conversational design (handoff phrasing, confirmation logic) rather than capability—a fix that requires prompt and policy changes, not a stronger base model.

4.4 Quantifying Adversarial Risk

While pass^k quantifies reliability under aleatoric noise, the Risk Analyser quantifies reliability under adversarial perturbation. This subsection formalises the scoring pipeline (Figure 4, Phases 3–4).

Value function and counterfactual signals. Let s_t denote the simulation state captured at the t -th required intermediate state. The success-probability value function and its perturbed counterpart are estimated by Monte-Carlo branching:

$$V(s_t) = P(\text{success} \mid s_t) \approx \frac{1}{N} \sum_{i=1}^N \mathbf{1}\{\rho_i\}, \quad (6)$$

$V'(s_t, p)$ analogously under perturbation p , where $\rho_i \in \{0, 1\}$ is the success-indicator function of the i -th rollout (1 if the rollout reaches the task’s success conditions, 0 otherwise). The two derived per-checkpoint quantities are the failure-probability gap and the extra-turn cost:

$$\Delta P(s_t, p) = V(s_t) - V'(s_t, p), \quad \Delta T(s_t, p) = \mathbb{E}[T'] - \mathbb{E}[T]. \quad (7)$$

Sensitivity and causal scores. Every checkpoint is scored along two complementary axes:

$$g(s_t) = \frac{1}{D} \sum_{d=1}^D \frac{|V(s_t) - V(\text{perturb}_d(s_t))|}{|\Delta_d|}, \quad (8)$$

$$c(s_t) = \log P(y^* \mid \tau_{1:T}) - \log P(y^* \mid \tau_{1:T}^{[s_t \leftarrow \text{corrupt}]}) \quad (9)$$

where $g(s_t)$ is a finite-difference discrete sensitivity averaged over D state dimensions perturbed at s_t —capturing the local response of V to small state changes—while $c(s_t)$ is a counterfactual log-likelihood drop induced by corrupting s_t within the trajectory τ relative to the eventual outcome y^* —capturing path-level causal necessity of s_t . The two are blended into a single per-checkpoint importance:

$$I(s_t) = \alpha \tilde{g}(s_t) + (1 - \alpha) \tilde{c}(s_t), \quad \alpha \in [0, 1], \quad (10)$$

with $\tilde{g}, \tilde{c}$ min-max normalised to $[0, 1]$ and $\alpha=0.5$ as the default. The form is inspired by counterfactual step-importance methods from chain-of-thought reasoning [28]: g captures local geometry, c captures global causal structure, and the convex combination spans the spectrum. In the experiments reported here we use the default single-dimension perturbation ($D=1$), under which $c(s_t)$ reduces to $g(s_t)$ and $I(s_t)$ collapses to the sensitivity term; the α -blend above is the general form, exercised only when multi-dimensional perturbations make g and c diverge.

Risk Score. Per-perturbation risk averages over checkpoints and combines failure-probability and delay terms:

$$\text{Risk}(p) = w_1 \overline{\Delta P(p)} + w_2 \text{norm}(\overline{\Delta T(p)}), \quad (11)$$

where $\bar{\cdot}$ is the mean over checkpoints, $\text{norm}(\cdot)$ is min-max normalisation across the perturbation set, and w_1, w_2 are profile-configurable (default $w_1 = 0.7, w_2 = 0.3$). Three weighting schemes for converting $I(s_t)$ to per-checkpoint weights—baseline-plus-boost ($w_t = 1 + \beta I_t$), min-max bounded, and softmax-with-temperature—are supported, corresponding to conservative, balanced, and aggressive risk attitudes.

Table 3: Mass functions per evidence source in Dempster-Shafer fusion. Each source assigns mass to three hypotheses: $m(\text{risk})$ = belief that the checkpoint is risky, $m(\neg\text{risk})$ = belief that it is safe, and $m(\Theta)$ = residual ignorance. One-sided sources contribute evidence for risk only, encoding that evidence of vulnerability is not evidence of safety. Notation: V = baseline success probability; $\tilde{g}(s_t)$ = normalised sensitivity; $\tilde{c}(s_t)$ = normalised causal score; ΔP = failure-probability gap.

Source	$m(\text{risk})$	$m(\neg\text{risk})$	$m(\Theta)$
MC rollouts (conf = $1 - 1/\sqrt{N}$)	$(1 - V) \text{ conf}$	$V \text{ conf}$	$1 - \text{conf}$
Sensitivity (one-sided)	$\tilde{g}(s_t)$	0	$1 - \tilde{g}(s_t)$
Causal (one-sided)	$\tilde{c}(s_t)$	0	$1 - \tilde{c}(s_t)$
Max adversarial ΔP (one-sided)	$\max_p \Delta P$	0	$1 - \max_p \Delta P$

Dempster-Shafer evidence fusion. A single checkpoint accumulates risk evidence from heterogeneous sources of varying confidence; with small N this evidence carries genuine epistemic uncertainty rather than just noise. Define the frame of discernment $\Theta = \{\text{risk}, \neg\text{risk}\}$. Each source supplies a mass function $m(\cdot)$ over $\{\{\text{risk}\}, \{\neg\text{risk}\}, \Theta\}$, with mass on Θ representing ignorance as first-class [24]. Table 3 lists the four sources. Dempster’s normalised rule of combination produces a fused mass:

$$m_{12}(A) = \frac{1}{1 - k} \sum_{B \cap C = A} m_1(B) m_2(C), \quad k = \sum_{B \cap C = \emptyset} m_1(B) m_2(C), \quad (12)$$

where the conflict coefficient k is reported alongside the result and flags high-disagreement cases ($k > 0.7$). Per-checkpoint Belief $\text{Bel}(\text{risk}) = m(\{\text{risk}\})$, Plausibility $\text{Pl}(\text{risk}) = 1 - m(\{\neg\text{risk}\})$, and Uncertainty $\text{Pl} - \text{Bel}$ are aggregated to task level via importance-weighted average:

$$\text{Bel}_{\text{task}} = \sum_t \omega_t \text{Bel}(s_t), \quad \omega_t = I(s_t) / \sum_s I(s). \quad (13)$$

Three choices distinguish this from textbook D-S: $\sqrt{N}$ -scaled MC mass with one-sided informational sources (Table 3), importance-weighted task aggregation, and explicit conflict reporting—together preventing the dilution of strong evidence and the silent masking of source disagreement that flat-averaging baselines produce.

Shapley attribution across attack categories. To attribute aggregate task risk across the four attack types $\mathcal{N} = \{\text{state, message, action, system}\}$, we compute exact Shapley values [26]:

$$\phi_i = \sum_{S \subseteq \mathcal{N} \setminus \{i\}} \frac{|S|! (|\mathcal{N}| - |S| - 1)!}{|\mathcal{N}|!} \times [v(S \cup \{i\}) - v(S)], \quad (14)$$

with coalition value $v(S) = \min(1, \sum_{i \in S} \overline{\Delta P_i})$. With $|\mathcal{N}| = 4$, exact computation over all $2^4 = 16$ coalitions is tractable; the four Shapley axioms (efficiency, symmetry, null player, additivity) guarantee a fair, principled attribution capturing interaction effects between categories. Attack-graph kill-chain analysis [27] is left to future work, as the current single-attack-dominance regime makes multi-step chains uninformative.

4.5 Evaluate-to-Train Closed Loop

The export pipeline ties directly to the evaluation layer: a minimum behavioural-score threshold gates which conversations enter the training set, DPO pairs are auto-constructed from high- and low-scoring responses to the same stimulus, and every exported example carries full provenance (persona config, per-evaluator scores, evaluator model, timestamp). Three redaction profiles—None, Basic, Strict—govern information sharing. Because each conversation is grounded in a specific OCEAN trait vector, the exported dataset has structural diversity in directness, emotional valence, and engagement style that naive prompt perturbation cannot achieve.

4.6 Personality-Stratified Model Comparison

A critical use case is evaluating whether fine-tuning improves performance *across the full personality spectrum*. The `ModelComparisonRunner` runs a baseline and one or more fine-tuned checkpoints through identical simulation packs, producing comparison reports stratified by OCEAN quadrant. This enables *personality-aware early stopping* (best pass^k on the hardest quadrant rather than lowest aggregate loss), *regression detection* (does epoch N degrade cooperative-user performance while improving adversarial?), and *fault-category tracking* (e.g., `WRONG_PARAMETERS`: 18% $\rightarrow$ 12% $\rightarrow$ 7% across epochs). The `ExperimentRunner` adds A/B testing with paired significance tests and Cohen’s d effect sizes.

5 Experiments

5.1 Experiment 1: Persona-Panel Evaluation of an Analytics Agent

To demonstrate AgentWorld’s diagnostic capabilities on a real agent, we evaluated a production *conversational analytics agent*—an LLM-based system that provides natural-language access to business KPIs across geographies, segments, and products. The agent was integrated into AgentWorld with zero code changes.

Setup. Ten analyst personas were configured, each combining an OCEAN trait vector with a distinct role and query domain (e.g., campaign analysis, regional market deep-dives, executive synthesis, data quality auditing, competitive intelligence). Each persona ran 3 multi-turn conversation exchanges against the analytics agent, producing 60 messages (30 from the agent, 30 from personas) and 240 evaluator judgments (4 evaluators $\times$ 60 messages). An LLM judge scored coherence and relevance; heuristic evaluators scored length and keyword safety. All **30/30 agent calls succeeded** with no fallbacks. Average response length showed a **7.2 $\times$ persona-driven range**—from 244 chars for the most terse persona to 1762 chars for the most verbose—direct evidence that persona configuration shapes conversation *shape*, not just content.

Scope note: because each persona here varies role and query domain jointly with its trait vector, this experiment is a persona-panel/domain-coverage demonstration; it is not designed to isolate the causal effect of personality alone. The controlled personality-vs-task analysis—identical task, OCEAN variant as the only manipulated factor—is provided by Experiment 2 (§5.2).

Aggregate results. At the aggregate level, the agent appeared to perform well (Table 4). A single-number evaluation would report

coherence of 0.79 and relevance of 0.75 as an acceptable agent. The per-persona breakdown tells a different story.

Table 4: Experiment 1: Evaluation of an analytics agent across 10 OCEAN-parameterised personas (60 messages, 240 evaluator judgments). (a) Aggregate scores across all messages. (b) Relevance by persona panel (agent responses only), ranked best to worst—aggregate relevance of 0.75 conceals a range from 0.88 to 0.61.

(a) Aggregate scores				
Evaluator	Mean	Min	Max	n
Coherence	0.793	0.45	0.95	60
Relevance	0.749	0.30	0.95	60
Length check	0.990	0.70	1.00	60
Keyword filter	1.000	1.00	1.00	60

(b) Relevance by persona panel			
Rank	Persona Role	Rel.	Coh.
1	Campaign Analyst	0.88	0.78
2	Competitive Intelligence	0.83	0.56
3	Regional Market Specialist	0.79	0.81
4	Data Quality Auditor	0.78	0.85
5	Performance Analyst	0.77	0.78
6	Forecasting Analyst	0.76	0.70
7	Engagement Analyst	0.73	0.89
8	Product Manager	0.68	0.86
9	Growth Analyst	0.67	0.84
10	Executive Synthesis	0.61	0.88

Per-persona breakdown. Breaking relevance out by persona panel reveals a clear domain-gap pattern (Table 4b)—aggregate relevance of 0.75 spans 0.88 (Campaign Analyst) to 0.61 (Executive Synthesis). Because role and domain co-vary with personality here, this spread reflects combined role/domain/personality difficulty rather than personality in isolation; Experiment 2 isolates the personality factor on a fixed task.

Failure patterns discovered. Structured evaluation revealed three systematic failures invisible to aggregate scoring. (1) *Response truncation* (5/30 responses, coherence 0.55–0.62): the agent’s output was cut off mid-sentence due to a silent token-limit on long analytical responses. (2) *Contextual drift in multi-turn* (3 instances, worst case 0.30 relevance): the agent answered the literal question but failed to connect its response to the ongoing thread—the judge noted it “introduces a new subtopic without connecting it to the previous discussion points.” (3) *Cross-domain data leakage* (1 instance, 0.60 relevance): when a persona focused on one geographic market asked about demographics, the agent cited data from a different region.

5.2 Experiment 2: Personality-Stratified Evaluation of a Customer-Support Agent

While Experiment 1 demonstrated persona-panel evaluation on an analytics agent, Experiment 2 scales the evaluation to a full personality $\times$ task matrix—19 simulations across 5 customer-support

tasks and 4 OCEAN persona variants—and then stress-tests the same tasks under adversarial perturbation via the Risk Analyser.

Setup. We evaluated a production customer-support agent (gpt-5.2) with access to payment, account, and transaction tools against four OCEAN persona variants—*Impatient* ($t_E=0.9$, $t_N=0.8$), *Analytical* ($t_C=0.9$), *Aggressive* ($t_A=0.1$, $t_N=0.8$), and *Anxious* ($t_N=0.9$, $t_E=0.2$)—simulated by gpt-5-mini. Five tasks of increasing complexity were selected: fetch latest transactions (read-only), check refund eligibility (status-check), cancel pending payments (state-mutating), complete payment for pending bills (state-mutating), and withdraw funds to a bank account (state-mutating with verification). LLM judges scored every message on five criteria: persona adherence, relevance, coherence, consistency, and task completeness. Table 5 reports all 19 results. Judge scores come from one simulation per cell (Paper Sim Pack, May 2025); pass^k columns come from a separate reliability study ($n=16$ reruns per cell) recording goal-state completion only.

Results. Under LLM-judge criteria, 17 of 19 single-run simulations passed and 2 failed (89.5%). The aggregate scores—coherence 0.891, consistency 0.876, relevance 0.815, persona adherence 0.812, completeness 0.508—reveal that the agent communicates well but often fails to complete tasks fully in the judge’s assessment.

Finding 1: Personality predicts judge-assessed failure. Both judge failures occurred on Fetch Latest Transactions with the Impatient (Overall 0.685) and Aggressive (0.678) personas—driven by low completeness (0.200) despite coherent dialogue. The Analytical persona passed at 0.755 on the same task. A single-persona evaluation using only the Analytical profile would have reported 100% judge pass. Judge failure here reflects conversation quality on one run, not necessarily goal-state unreliability (e.g., Fetch/Impatient: judge FAIL but $\text{pass}^1=1.00$ over 16 goal-state reruns).

Finding 2: Large persona-driven quality gaps. Within-task LLM-judge score ranges reached 0.145 points (Complete Payment: Impatient 0.906 vs. Anxious 0.761) and 0.122 points (Cancel Payments: Impatient 0.836 vs. Aggressive 0.714). These gaps are masked by the 0.779 aggregate and identify which persona–task pairs need targeted improvement.

Finding 3: Completeness is the bottleneck, driven by step-budget exhaustion. Completeness averaged only 0.508 while all other criteria exceeded 0.81. The Steps column in Table 5 reveals the mechanism: 14 of 19 simulations hit their step budget ($\text{ms} = \text{max_steps}$), while only 5 terminated via natural conversation end ($\text{cs} = \text{customer_signal}$). The contrast is sharpest within a single task—Complete Payment finished in just 3 steps for the Impatient persona (0.950 completeness) but consumed all 15 steps for the Anxious persona (0.420 completeness), whose repeated clarification requests exhausted the budget before all goal conditions were met. Token usage mirrored this pattern, ranging from ~23K (Withdraw, Aggressive, 2 steps) to ~187K (Withdraw, Anxious, 15 steps). Low completeness on a single judge-scored run does not always imply poor goal-state reliability on repeat (see pass^k columns).

Finding 4: pass^k exposes reliability gaps that single judge runs hide. Rerunning every cell $n=16$ times on goal-state success (last four columns of Table 5), Complete Payment and Withdraw to Bank stay perfectly reliable ($\text{pass}^k=1.00$), but other tasks erode as k grows—Check Refund Eligibility drops from $\text{pass}^1=0.88$ –0.94

to $\text{pass}^8=0.23$ –0.50 despite judge PASS on the original run. High judge scores on a single run thus mask cells where the agent cannot sustain repeated task completion—a demo-vs-production gap the Judge column alone cannot reveal.

Forward pointer to Experiment 3. Experiment 3 (§5.3) stress-tests the same task family under adversarial perturbation to reveal trajectory-level brittleness that behavioural scoring cannot see.

5.3 Experiment 3: Adversarial Stress-Test via the Risk Analyser

Setup. We apply the Risk Analyser (§3.4, §4.4) to five customer-support tasks—three drawn from Experiment 2 (Check Refund Eligibility, Cancel Pending Payments, Fetch Latest Transactions) plus two additional probes (File an INR Dispute, Add a New Card) extending coverage beyond Experiment 2’s set. The agent and topology match Experiment 2 (gpt-5.2, mesh). Per task we run $N=8$ baseline and $N=8$ adversarial Monte-Carlo rollouts at every required intermediate state, with $|P|=8$ task-aware perturbations (2 per type). This yields several hundred Monte-Carlo rollouts per task ($N=8$ at each required intermediate state, plus $N=8$ under each of the $|P|=8$ perturbations) and on the order of a thousand across the five tasks; the per-checkpoint sample ($N=8$) is deliberately small, which we revisit as an uncertainty caveat below. Scoring uses defaults: $\alpha=0.5$, $w_1=0.7$, $w_2=0.3$.

Aggregate results. Table 6 contrasts each task’s behavioural pass-rate against four Risk Analyser metrics: $V_{\min}$, top ΔP and its attack type, task-level Belief, and per-attack-category Shapley share. Table 7 walks through a single D–S fusion at the first checkpoint of Check Refund Eligibility, where three active evidence sources (the causal source collapses into sensitivity under the default single-dim perturbation) fuse to $\text{Bel}=0.727$, $\text{Pl}=0.889$, conflict $k=0.062$. All Risk Analyser estimates derive from $N=8$ rollouts and carry correspondingly wide uncertainty: the headline $V_{\min}=0.375$ has a 95% Wilson interval of $[0.14, 0.69]$, so reported values should be read as point estimates pending higher- N confirmation. Four findings:

(1) *Pass-rates mask trajectory brittleness.* Check Refund Eligibility achieved 100% behavioural pass yet $V_{\min}=0.375$ at its first required intermediate state (only 3/8 unperturbed rollouts succeed). State-mutation drives the $\Delta P=0.458$ peak; D–S fuses to $\text{Bel}=0.727$, confirming genuine risk (Table 7).

(2) *Tool/infra layers dominate risk.* Shapley attributes 46% to system-disruption and 38% to action-interception, versus only 12% state and 4% message. The most damaging perturbation—action-interception on File an INR Dispute ($\Delta P=0.800$)—targets the layer that behavioural scoring cannot observe.

(3) *Two complementary views.* Behavioural evaluation (Experiment 2) identifies which personas and tasks fail; the Risk Analyser (Experiment 3) identifies which trajectory steps and system layers are brittle. Together they provide a two-dimensional reliability map.

(4) *Fusion is not averaging.* On the Check Refund checkpoint, flat-averaging the same evidence sources yields a risk mass of 0.385, whereas Dempster–Shafer fusion concentrates the concordant evidence to $\text{Bel}=0.727$ ($k=0.062$)—nearly double—isolating the marginal contribution of the D–S layer over naive aggregation.

Table 5: Experiment 2: 19 (task×persona) cells on a customer-support agent (mesh topology). Left block: one simulation per cell (agent gpt-5.2, customer gpt-5-mini); LLM judges score five criteria per message; Judge = PASS if aggregated overall score ≥ 0.70 . Right block: $n=16$ independent reruns per cell (both roles gpt-5.2); $\text{pass}^k = \binom{c}{k} / \binom{n}{k}$ on goal-state completion (judge scores not re-aggregated over reruns). Steps = turns/budget; cs/ms = customer_signal / max_steps. $\dagger$ Judge and pass^k diverge.

Task	Persona	Steps	Single run (n = 1; judge)					Compl	Judge	Reliability (n = 16; goal state)			
			Overall	P.Adh	Rel	Coh	Con			pass ¹	pass ²	pass ⁴	pass ⁸
Complete Payment for Pending Bills	Impatient	3/15 cs	0.906	0.742	0.967	0.938	0.933	0.950	PASS	1.00	1.00	1.00	1.00
	Analytical	4/15 cs	0.875	0.857	0.723	0.909	0.939	0.950	PASS	1.00	1.00	1.00	1.00
	Aggressive	3/15 cs	0.846	0.680	0.697	0.933	0.970	0.950	PASS	1.00	1.00	1.00	1.00
	Anxious	15/15 ms	0.761	0.842	0.780	0.868	0.892	0.420	PASS	1.00	1.00	1.00	1.00
Withdraw to Bank	Aggressive	2/15 cs	0.793	0.698	0.743	0.955	0.950	0.620	PASS	1.00	1.00	1.00	1.00
	Impatient	15/15 ms	0.858	0.830	0.876	0.936	0.928	0.720	PASS	1.00	1.00	1.00	1.00
	Analytical	15/15 ms	0.824	0.869	0.793	0.863	0.875	0.720	PASS	1.00	1.00	1.00	1.00
	Anxious	15/15 ms	0.777	0.834	0.855	0.937	0.840	0.420	PASS	1.00	1.00	1.00	1.00
Cancel Pending Payments	Impatient	10/10 ms	0.836	0.824	0.884	0.912	0.940	0.620	PASS	1.00	1.00	1.00	1.00
	Anxious	10/10 ms	0.749	0.866	0.813	0.929	0.936	0.200	PASS	1.00	1.00	1.00	1.00
	Analytical	10/10 ms	0.742	0.894	0.867	0.923	0.827	0.200	PASS	0.94	0.88	0.75	0.50
	Aggressive	10/10 ms	0.714	0.814	0.738	0.927	0.813	0.280	PASS	0.94	0.88	0.75	0.50
Check Refund Eligibility	Aggressive	10/10 ms	0.776	0.768	0.861	0.887	0.742	0.620	PASS [†]	0.88	0.76	0.55	0.23
	Anxious	10/10 ms	0.769	0.845	0.856	0.908	0.816	0.420	PASS [†]	0.88	0.76	0.55	0.23
	Analytical	10/10 ms	0.758	0.904	0.804	0.851	0.792	0.440	PASS [†]	0.94	0.88	0.75	0.50
Fetch Latest Transactions	Analytical	10/10 ms	0.755	0.861	0.782	0.789	0.822	0.520	PASS	0.94	0.88	0.75	0.50
	Anxious	10/10 ms	0.724	0.876	0.829	0.839	0.875	0.200	PASS	1.00	1.00	1.00	1.00
	Impatient	10/10 ms	0.685	0.767	0.764	0.816	0.879	0.200	FAIL [†]	1.00	1.00	1.00	1.00
	Aggressive	10/10 cs	0.678	0.653	0.861	0.805	0.872	0.200	FAIL [†]	0.88	0.76	0.55	0.23
Aggregate (19 sims)			0.779	0.812	0.815	0.891	0.876	0.508	17/19 judge pass per-cell; not pooled				

Notes: Judge scores from Paper Sim Pack (May 2025; one run per cell). pass^k from a separate reliability rerun; Check Refund / Fetch Latest used semantically equivalent evaluation DB task configs.

Table 6: Risk Analyser reveals trajectory-level brittleness invisible to pass-rate testing. $V_{\min}$ = lowest baseline success probability across checkpoints (<1.0 = pre-existing brittleness); top ΔP = largest failure-probability gap (attack type in parentheses); task Bel = D-S fused belief (lower bound on risk); Shapley % = risk attribution to State / Message / Action / Infra attacks. Config: $N=8/8$, $|P|=8$.

Task	pass %	$V_{\min}$	top ΔP (type)	task Bel	Shapley % (S/M/A/I)
Fetch Latest Transactions	50	1.00	.333 (sys.)	.900	0/0/53/47
Check Refund Eligibility	100	.375	.458 (state)	.925	28/12/22/38
Cancel Pending Payments	100	1.00	.500 (act.)	1.00	0/0/50/50
File an INR Dispute	100	1.00	.800 (act.)	1.00	3/5/31/61
Add a New Card	100	1.00	.333 (sys.)	1.00	32/4/32/32
Mean Shapley:					12/4/38/46

Table 7: D-S worked example at checkpoint $t=1$ (the first required intermediate state) of Check Refund Eligibility. Baseline $V(s_t)=0.375$ ($N=8$, $3/8$ succeed)—pre-existing brittleness invisible to behavioural scoring. Dempster’s rule fuses three active sources to Bel=0.727, Pl=0.889, Uncertainty=0.162, conflict $k=0.062$.

Evidence Source	$m(\text{risk})$ (risky)	$m(\neg\text{risk})$ (safe)	$m(\Theta)$ (uncertain)
MC rollouts ($N=8$, conf=0.646)	0.404	0.242	0.354
Sensitivity ($\bar{g}=0.375$, one-sided)	0.375	0	0.625
Causal	collapsed into sensitivity ($\bar{c}=\bar{g}$)		
Max ΔP ($=0.375$, one-sided)	0.375	0	0.625
Fused result	0.727	0.111	0.162
Bel=0.727, Pl=0.889, Uncertainty=0.162, conflict $k=0.062$			

6 Discussion and Limitations

Limitations. The persona-to-behaviour mapping relies on prompt engineering without validated calibration against human studies: we do not yet verify that a simulated persona with a given OCEAN vector elicits the same agent behaviour a real user of that profile would, so the personas should be read as controlled, reproducible stress profiles rather than validated human proxies. Establishing that link requires a human-subjects study (matching simulated personas to real users on Big Five instruments and comparing elicited agent behaviour), which we leave to future work. Relatedly, our conclusions depend on LLM judges, which introduce their own biases; a cross-judge agreement analysis is needed to bound this dependence, and DPO auto-construction uses score differentials as

a proxy for human preference. On the adversarial side, sensitivity $g(s_t)$ and causal $c(s_t)$ are numerically equivalent under the default single-dimension perturbation (multi-dimensional perturbations separate them); Dempster’s rule degrades under high conflict ($k > 0.7$), which we surface but do not yet resolve; and attack-graph kill-chain analysis is deferred because current single-attack dominance makes multi-step chains uninformative.

Closing. AgentWorld unifies personality-driven simulation, reliability measurement, training-data generation, and adversarial risk analysis in one closed-loop pipeline—exposing IR failure modes that uniform testing misses and making personality-aware reliability evaluation practical for agentic IR.

References

- [1] J. S. Park, J. C. O'Brien, C. J. Cai, M. R. Morris, P. Liang, and M. S. Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proc. UIST*, 2023.
- [2] G. Li, H. Hammoud, H. Itani, D. Khizbullin, and B. Ghanem. CAMEL: Communicative agents for “mind” exploration of large language model society. In *Proc. NeurIPS*, 2023.
- [3] S. Yao, N. Shinn, P. Razavi, and K. Narasimhan. τ -bench: A benchmark for tool-agent-user interaction in real-world domains. *arXiv:2406.12045*, 2024.
- [4] V. Barres, H. Dong, S. Ray, X. Si, and K. Narasimhan. τ^2 -bench: Evaluating conversational agents in a dual-control environment. *arXiv:2506.07982*, 2025.
- [5] H. Trivedi, T. Khot, M. Hartmann, R. Manku, et al. AppWorld: A controllable world of apps and people for benchmarking interactive coding agents. In *Proc. ACL*, 2024.
- [6] P. T. Costa and R. R. McCrae. *Revised NEO Personality Inventory (NEO-PI-R) and NEO Five-Factor Inventory (NEO-FFI) Professional Manual*. Psychological Assessment Resources, 1992.
- [7] Q. Wu, G. Bansal, J. Zhang, et al. AutoGen: Enabling next-gen LLM applications via multi-agent conversation. *arXiv:2308.08155*, 2023.
- [8] Microsoft Research. TinyTroupe: LLM-powered multiagent persona simulation. <https://github.com/microsoft/TinyTroupe>, 2024.
- [9] S. Zhou, F. F. Xu, H. Zhu, X. Zhou, et al. WebArena: A realistic web environment for building autonomous agents. In *Proc. ICLR*, 2024.
- [10] T. Xie, D. Zhang, J. Chen, et al. OSWorld: Benchmarking multimodal agents for open-ended tasks in real computer environments. In *Proc. NeurIPS*, 2024.
- [11] X. Liu, H. Yu, H. Zhang, Y. Xu, et al. AgentBench: Evaluating LLMs as agents. In *Proc. ICLR*, 2024.
- [12] S. G. Patil, H. Mao, F. Yan, et al. BFCL: Berkeley function calling leaderboard. In *Proc. ICML*, 2025.
- [13] G. Jiang, M. Xu, S. Zhu, et al. Evaluating and inducing personality in pre-trained language models. In *Proc. NeurIPS*, 2024.
- [14] X. Zhou, H. Zhu, et al. SOTOPIA: Interactive evaluation for social intelligence in language agents. In *Proc. ICLR*, 2024.
- [15] X. Zhou, Z. Su, et al. SOTOPIA-S4: A user-friendly system for flexible, customizable social simulation. In *Proc. NAACL Demos*, 2025.
- [16] R. Shea, Y. Lu, L. Qiu, and Z. Yu. SAGE: A top-down bottom-up knowledge-grounded user simulator for multi-turn agent evaluation. *arXiv:2510.11997*, 2025.
- [17] P. Chong, H. Abichandani, J. Shen, et al. TED: Talk, evaluate, diagnose: User-aware agent evaluation with automated error analysis. *arXiv:2603.15483*, 2026.
- [18] L. R. Goldberg. An alternative “description of personality”: The Big-Five factor structure. *Journal of Personality and Social Psychology*, 59(6):1216–1229, 1990.
- [19] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafraan, K. Narasimhan, and Y. Cao. ReAct: Synergizing reasoning and acting in language models. In *Proc. ICLR*, 2023.
- [20] E. Debenedetti, J. Zhang, M. Balunović, L. Beurer-Kellner, M. Fischer, and F. Tramèr. AgentDojo: A dynamic environment to evaluate prompt injection attacks and defenses for LLM agents. In *Proc. NeurIPS Datasets and Benchmarks*, 2024.
- [21] Q. Zhan, Z. Liang, Z. Ying, and D. Kang. InjecAgent: Benchmarking indirect prompt injections in tool-integrated large language model agents. In *Findings of ACL*, 2024.
- [22] Z. Chen, Z. Xiang, C. Xiao, D. Song, and B. Li. AgentPoison: Red-teaming LLM agents via poisoning memory or knowledge bases. In *Proc. NeurIPS*, 2024.
- [23] Y. Ruan, H. Dong, A. Wang, et al. Identifying the risks of LM agents with an LM-emulated sandbox. In *Proc. ICLR*, 2024.
- [24] G. Shafer. *A Mathematical Theory of Evidence*. Princeton University Press, 1976.
- [25] R. R. Yager. On the Dempster–Shafer framework and new combination rules. *Information Sciences*, 41(2):93–137, 1987.
- [26] L. S. Shapley. A value for n -person games. In *Contributions to the Theory of Games*, vol. II, pages 307–317. Princeton University Press, 1953.
- [27] O. Sheyner, J. Haines, S. Jha, R. Lippmann, and J. M. Wing. Automated generation and analysis of attack graphs. In *Proc. IEEE S&P*, 2002.
- [28] A. Madaan and A. Yazdanbakhsh. Text and patterns: For effective chain of thought, it takes two to tango. *arXiv:2209.07686*, 2022.