PersonaForge: Realistic Multi-Turn User Simulation for Agentic LLMs

Anonymous ACL submission

Abstract

Large language models are increasingly used as agentic workflow executors, yet existing training data and benchmarks largely assume informationally complete, single-turn queries. Our analysis of 16K real-world sessions shows that 75.9% of interactions are multi-turn, revealing a substantial gap between how users interact with agents and how such systems are trained and evaluated. We introduce **PersonaForge**, a user simulation framework for synthesizing realistic multi-turn user–agent interactions. PersonaForge combines a four-dimensional persona space, SOUL-driven behavioral control calibrated to real-user statistics, and Reverse Deep Construction grounded in authentic seed queries. Using PersonaForge, we construct a 6.3K-record training dataset and **PersonaForge-Bench**, a manually annotated 138-task benchmark spanning over 20 professional domains with four-dimensional scoring. Experiments on Qwen3.5-27B show that PersonaForge training improves the composite score by +4.1%, with gains across all four dimensions and the largest improvements in Task Completion (+6.0%) and Response Quality (+6.8%). Further analyses show that PersonaForge-trained agents use fewer turns and tool calls, suggesting improved interaction efficiency, while ablations confirm the contribution of SOUL components and adaptive simulation. Together, PersonaForge and PersonaForge-Bench establish a foundation for training and evaluating agents under realistic multi-turn user interaction.

1 Introduction

Large language models (Ouyang et al., 2022; OpenAI, 2023) have evolved from single-turn question answerers into agentic workflow executors (Yao et al., 2023; Schick et al., 2023; Wu et al., 2023): systems that solve real-world tasks through tool use, multi-step reasoning, and sustained dialogue. In these settings, success depends on managing iterative, tool-intensive interactions as users steer the system over time. Yet agentic training and evaluation still often assume that the user's first message is *informationally complete*. Existing tool-use data synthesis methods (Liu et al., 2024a,b) typically generate offline trajectories over fixed API schemas, where a fully specified query leads to a planned tool-call chain. This misses real workflows with incomplete and evolving user intent.

Real usage looks different. Our analysis of 16K real-world sessions from public datasets (SWE-chat (Baumann et al., 2026) and Computer-Use (Shaikh et al., 2025)) and internally collected sessions shows that **75.9% of interactions are multi-turn** (Section 2); among multi-turn sessions, the median number of user turns is 10 and the average number of tool calls is 99. Users add requirements, change direction, inspect intermediate outputs, and correct errors; in 38.7% of multi-turn sessions, they explicitly request corrections. This gap between complete-query assumptions and incomplete, user-steered interaction is a central bottleneck for building capable agentic systems.

Closing this gap requires a *realistic user simulator* that reproduces how humans interact with agentic systems: providing incomplete requests, progressively disclosing information, and challenging incorrect outputs. Building such a simulator poses three linked challenges: (i) **diverse yet coherent users**, whose occupations, personalities, technical proficiency, and knowledge backgrounds shape both task goals and communication styles; (ii) **controllable turn-level behavior**, so the simulator produces terse, incremental, and sometimes corrective messages rather than verbose, overly cooperative LLM-style dialogue; and (iii) **grounded and scalable conversations**, which originate from authentic user needs, proceed through live interaction with an agent, and avoid low-quality data.

We address these challenges with **PersonaForge**, a data synthesis framework for realistic user simu-

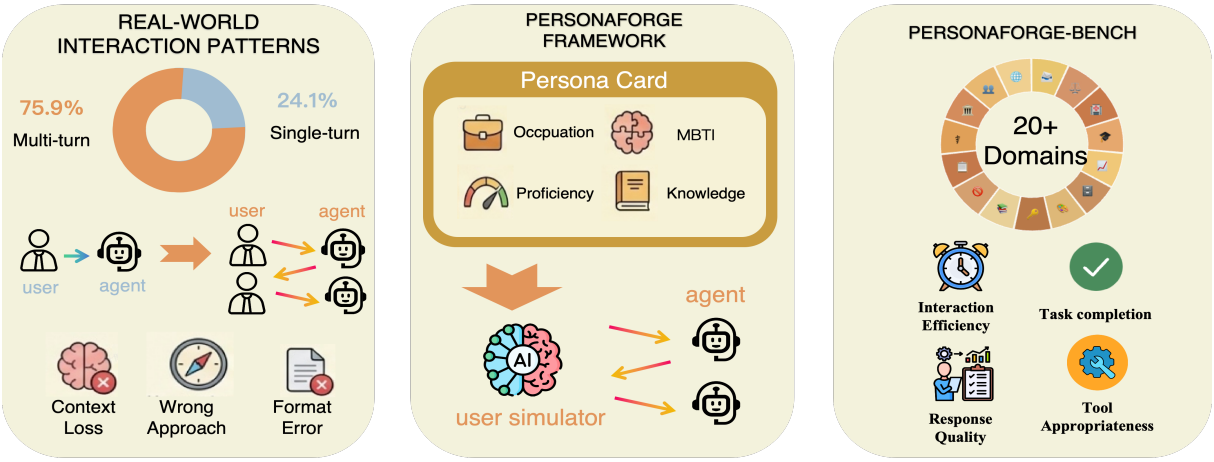

Figure 1: Overview of PersonaForge: real-session analysis, data synthesis, and benchmark evaluation.

lation. PersonaForge uses LLMs to act as realistic users interacting with a deployed agentic system, as illustrated in Figure 1. Its design maps directly to the three challenges: (1) a **4-dimensional persona space** spanning occupation, personality, technical proficiency, and knowledge background for diverse yet coherent users; (2) **SOUL-driven behavioral control**, which constrains messages to be terse, incremental, and corrective when needed; and (3) **Reverse Deep Construction**, which infers persona profiles from real seed queries to ground each live interaction in authentic user needs. Applied at scale, PersonaForge produces 6.3K realistic conversations that pass quality filtering for training. To evaluate models under realistic user interaction, we further introduce **PersonaForge-Bench**, a human-verified benchmark derived from real user requests. It contains 138 tasks that reflect real user scenarios across 20+ professional domains and are designed around progressive information disclosure, embedded contradictions, and domain-specific rubrics, requiring models to elicit missing constraints, maintain state as requirements evolve, and respond appropriately to user steering. Each task requires sustained multi-turn interaction and is scored over 3 independent trials along four dimensions: interaction efficiency, tool appropriateness, task completion, and response quality, providing a controlled testbed for measuring agent behavior beyond complete-prompt execution.

Experiments on open-source models Qwen3.5-27B (Qwen Team, 2025) and MiMo-V2-Flash (Xiaomi LLM-Core Team, 2026) show that PersonaForge training improves the composite score across all four dimensions: +4.1% on Qwen3.5-27B and +15.7% on MiMo-V2-Flash, with the largest gains on task completion and response quality. These results show training on realistic multi-turn interaction data helps models learn more effective dialogue management and tool-use strategies, strengthening their ability to interact with real users. Together, PersonaForge and PersonaForge-Bench provide a data synthesis and evaluation foundation for studying user simulation in agentic systems, moving the field beyond complete-query benchmarks toward realistic user-agent interaction.

2 Multi-Turn Interactions in the Wild

Our analysis combines public benchmarks of agentic interaction with internal deployment logs, balancing external authority with domain coverage. The public sources are **SWE-chat** (Baumann et al., 2026) and the **computer-use traces** of Shaikh et al. (2025). We complement these with conversational sessions and task-execution trajectories from an internally deployed agentic assistant, with the internal logs sampled under opt-in consent and anonymized at collection time. In aggregate, the analysis spans **16K agentic sessions**: the two public datasets each contribute 25% and anchor reproducible baselines, while internal logs contribute 50% and extend coverage to broader professional domains and deployed tool-mediated workflows.

2.1 Query and Workflow Patterns

Real agentic use is heterogeneous and situated. Figure 2 reports task volume and multi-turn share across seven categories, identified by keyword matching on the first user message (Appendix A). The distribution is broad: data analysis and system/DevOps operations account for the largest volume, while coding, file operations, IM/bot inter-

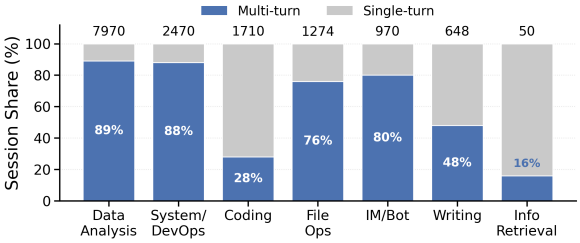

Figure 2: Task composition by category.

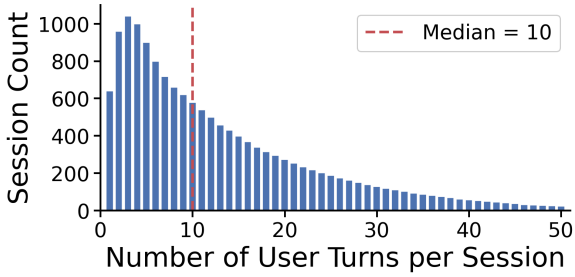

Figure 3: User turns per session.

Metric	Single	Multi	Ratio
Sessions	3.9K	12.1K	—
Avg. user turns	1.0	25.8	25.8×
Avg. tool calls	9.6	99.3	10.3×
Avg. length (chars)	45K	212K	4.7×
User correction	2.0%	38.7%	19.4×
Post-tool corr. share	—	36.0%	—

Table 1: Single-turn vs. multi-turn session statistics.

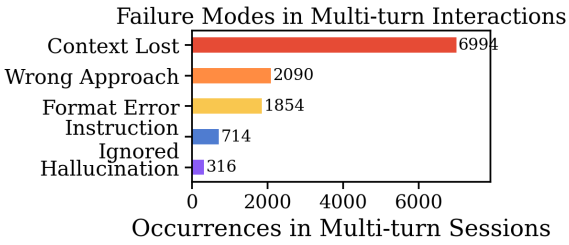

Figure 4: Failure modes in user correction messages.

actions, writing, and information retrieval all appear as recurring use cases. These categories presuppose different user backgrounds and communication norms. A DevOps request often assumes infrastructure context and command-line fluency; a writing request may rely on style preferences and implicit audience constraints; an information-retrieval request may be issued by a user with little technical context. This heterogeneity suggests that a realistic simulator cannot sample tasks independently from users. The user profile must be coherent with the task, the domain vocabulary, and the level of expertise expressed in the conversation.

Intent is disclosed over time. Aggregating across categories, Figure 3 shows that **75.9%** of sessions involve two or more user turns, with a median of 10 user turns. Real users therefore rarely provide a complete task specification upfront. Instead, they add constraints, inspect intermediate results, clarify ambiguous requirements, and revise the request as the assistant acts. This contrasts sharply with the training corpus underlying our deployed assistant, which is *100% single-turn*, and points to a need for simulated users that can reveal intent incrementally rather than only issue complete initial queries.

2.2 Execution-Coupled Feedback

User turns are coupled to tool execution. Table 1 shows that multi-turn sessions contain many more tool calls (99.3 vs. 9.6 on average), more intermediate content (212K vs. 45K characters), and more user turns than single-turn sessions. We interpret these quantities not as a generic difficulty score, but as evidence that real conversations are grounded in execution. User messages respond to files, command outputs, retrieved information, partial calculations, and previous assistant decisions. This makes offline scripting insufficient: a realistic trajectory should unfold through interaction with a running agentic system, so that user follow-ups are conditioned on what the assistant actually does.

User feedback is corrective rather than merely conversational. Users explicitly signal errors in 38.7% of multi-turn sessions under our correction lexicon (19.4× single-turn), and 36.0% of corrections occur after tool execution (Appendix A). Figure 4 shows that these corrections target *context loss* (7.0K), *wrong approach* (2.1K), *format errors* (1.9K), *instruction ignored* (714), and *hallucination* (316). These are not generic follow-up questions; they are closed-loop responses to the assistant's behavior. A simulator must therefore control turn-level behavior so that it can produce terse, incremental, and corrective messages when the interaction state calls for them, instead of defaulting to verbose and overly cooperative dialogue.

2.3 Implications for User Simulation

The analyses above portray real agentic interaction as a user-steered workflow rather than a completed query followed by execution. Three properties are especially important for realistic simulation. First, users and tasks are coupled: the same surface task can be expressed differently depending on occu-

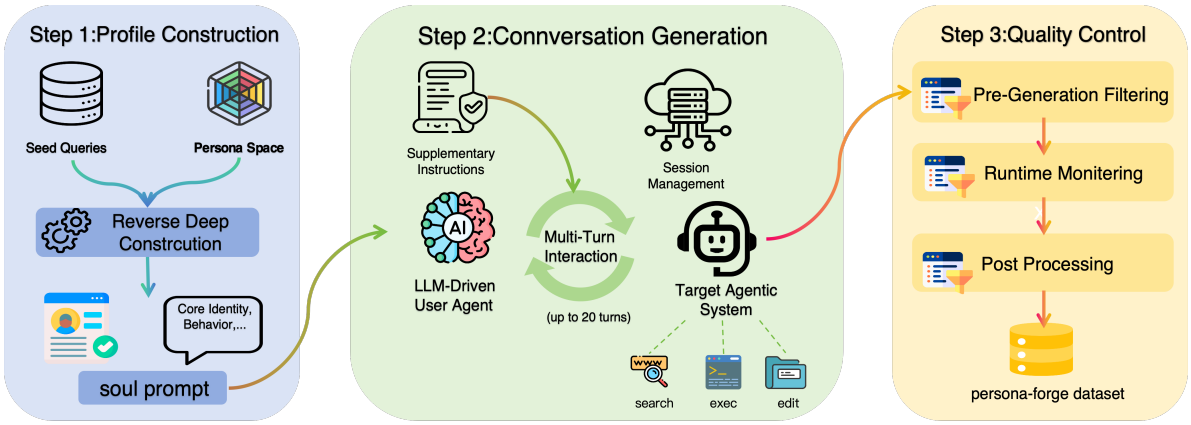

Figure 5: Overview of the PersonaForge framework.

pation, technical proficiency, personality, and domain knowledge. Second, intent unfolds over turns: users reveal constraints, react to intermediate outputs, and correct the assistant when needed. Third, the conversation is grounded in the assistant's actual behavior, especially tool calls and partial results. These observations motivate us to design a data-generation framework that controls these axes explicitly: who the user is, how the user communicates at each turn, and what authentic task context grounds the interaction.

3 PersonaForge

PersonaForge turns a real seed query into a trainable multi-turn user–agent trajectory. A realistic simulator cannot simply sample a persona or ask a fully specified prompt: it must keep the user compatible with the task, decide what hidden intent to reveal at each turn, and ground follow-ups in the assistant's actual execution. PersonaForge therefore decomposes synthesis into three stages that correspond to these requirements. **Profile construction** (§3.1) defines who the user is and why the task is natural for them. **Conversation generation** (§3.2) controls how the user reveals intent, reacts, and corrects the assistant over turns. **Quality control** (§3.3) filters live interaction traces so that only coherent, user-like trajectories enter the retained corpus (Figure 5).

3.1 Profile Construction

Profile construction addresses user–task coherence by defining the hidden state from which the simulated user will act. Rather than sampling tasks and users independently, PersonaForge links them through a real seed query: the query anchors an authentic task need, while the persona space determines the user's background, expertise, and communication style. We instantiate this state from a 4-dimensional persona space. **Occupation** (70 types) spans tech workers, professionals, students, entrepreneurs, and more, each annotated with tech affinity and compatible knowledge areas. **Personality type** (16 MBTI types) determines communication style; for example, INTJ users are terse and efficiency-focused, while ENFP users are curious and prone to topic shifts. **Technical proficiency** (3 levels: expert, intermediate, novice) controls whether the user speaks in jargon or plain language. **Knowledge background** (25 domains) encodes domain expertise with structured fields for expert skills, familiar topics, and blind spots. To keep these dimensions internally consistent, we enforce compatibility rules based on annotated affinities before generation; for example, a "non-technical" user is not paired with a high-tech-affinity occupation (see Figure 6 in the Appendix).

Reverse Deep Construction. Given the persona space, we construct profiles via **Reverse Deep Construction**: starting from real seed queries drawn from WildChat (Zhao et al., 2024) and LMSYS-Chat-1M (Zheng et al., 2024), an LLM works backward to infer a plausible persona profile, then generates a task scenario and connected memories naturally grounded in that persona. This design anchors each conversation in an authentic user need, preserving real-world task distributions rather than relying on synthetically generated queries. We contrast reverse-constructed profiles with forward-sampled ones on matched cases in Appendix B.3 to illustrate the resulting differences in anchor specificity and connected-memory depth.

SOUL prompt structure. The SOUL prompt is the executable representation of this user state.

It specifies a simulated user's identity, communication style, and hidden context, adapting the structured-prompt principle used in deployed assistant systems such as OpenClaw[1] to the *user* side of the interaction. The prompt has four components: (1) a core identity paragraph grounding the persona in a specific life situation; (2) personality–behavior mappings expressed as abstract parameters rather than utterance exemplars, which we found collapse diversity; (3) communication style parameters (message length, formality, emoji usage); and (4) connected memory templates that pre-establish session-specific facts, preventing the simulator from fabricating contradictory details when answering clarification questions. Connected memory is identified in our ablation as the single most impactful individual component.

3.2 Conversation Generation

Conversation generation addresses controllable turn-level behavior by converting the hidden user state into observable user messages. The key design is a deliberate *information asymmetry* (see Figure 7 in the Appendix): the simulator can access the SOUL prompt, connected memories, and task scenario, while the target system sees only accumulated user messages and its own tool outputs. This asymmetry lets the simulator reveal intent gradually instead of providing a complete task specification at the first turn. Each turn, the simulator conditions on the hidden state and on what the target system has actually done, producing short, naturalistic messages that add requirements, introduce realistic inconsistencies, or correct the assistant when the interaction calls for it.

Behavioral rules. Beyond the persona-specific content, every SOUL prompt includes fixed behavioral rules that make the simulator behave like a user rather than another assistant: *message brevity* (short follow-ups rather than full task restatements), *one question per turn* to avoid unnatural multi-question patterns, *active verification* of numerical results and code outputs to replicate real error correction behavior, *no AI self-awareness* to maintain persona consistency, and *natural termination* via a signal when the task is complete.

3.3 Quality Control

Quality control addresses scalable data construction by preventing common simulator failures from entering the training corpus. Because PersonaForge generates data through LLM-driven live interaction, a small number of role or format failures can otherwise contaminate many long trajectories. The filters therefore operate at three stages. *Pre-generation*: persona combinations are checked for coherence and seed queries are filtered for minimum complexity. *Runtime*: we detect *role reversal*, where the simulator shifts from issuing directives to completing the task itself, using a character-length heuristic that flags user messages exceeding 1,500 characters (calibrated on real-user data where 99.2% of follow-ups fall below this cutoff; flagged messages are additionally checked for code-paste patterns to avoid false positives). *Post-processing*: conversations undergo role-reversal truncation, degenerate pattern filtering, format normalization, and tool-configuration randomization.

Retained corpus. Table 2 summarizes the trajectories retained after filtering and formatting for training. The final data contains roughly 6.3K session-level records and 430K messages, with 96% being multi-turn (mean 9.3 user turns). User messages remain short, with a median length of about 100 characters. Detailed persona dimension distributions and quality control pipeline statistics are provided in Appendix B.4.

4 PersonaForge-Bench

PersonaForge-Bench targets realistic user-agent interaction. Section 1 identifies three requirements for realistic user simulation: diverse yet coherent users, controllable turn-level behavior, and conversations grounded in authentic needs and live agent behavior. Section 2 shows why these requirements matter in practice: real agentic use is heterogeneous across domains, user intent is disclosed over time, and feedback is often corrective and coupled to tool execution. A benchmark built from complete single-turn prompts would hide precisely these behaviors. We therefore construct 138 tasks spanning 20+ professional domains, with scenarios in both Chinese and English (composition details in Appendix C), on top of CLAW-EVAL (Ye and collaborators, 2026). Each task is derived from a realistic user scenario and requires sustained multi-turn interaction rather than one-shot task execution.

The benchmark faithfully evaluates user–LLM interaction trajectories. Its tasks are built around three principles that correspond to the observations above: *progressive information disclosure*

[1] https://openclaw.ai

Statistic	Value	Persona Dimension	Unique / Norm. H
Sessions	6.3K	Occupation	70 / 0.79
Messages (user / asst / tool)	59K / 103K / 268K	MBTI	16 / 0.77
Mean user turns	9.3	Tech Level	3 / 0.93
User msg length (median)	100 chars	Knowledge BG	25 / 0.91

Table 2: Retained training corpus statistics. *Norm. H* is normalized Shannon entropy.

tests whether the assistant can ask for and integrate missing constraints; *embedded contradictions* test whether it can maintain state and detect conflicts as new information arrives; and *domain-specific rubrics* ensure that success is judged against the concrete requirements of the professional scenario rather than generic helpfulness. We score each task along four equally weighted dimensions: Interaction Efficiency, Tool Appropriateness, Task Completion, and Response Quality (detailed definitions in Appendix C). These dimensions match the failure surfaces observed in Section 2: efficiency and tool appropriateness measure management of tool-intensive workflows, task completion measures recovery of hidden requirements and contradictions, and response quality measures whether the final answer is accurate and structured. Scoring is *task-specific*: each task includes a rubric with required clarification items, reference calculations with tolerance windows, and evaluation chains, ensuring the LLM judge (DeepSeek-V4-Pro (DeepSeek-AI, 2026)) evaluates against domain-appropriate criteria. The composite score averages all four dimensions across 3 independent trials per task, and human validation ($\kappa_{jh} = 0.79$, $r = 0.84$; Appendix D) confirms near-human reliability.

5 Experiments

This section evaluates whether PersonaForge data improves agentic interaction behavior and whether the gains hold beyond the PersonaForge-Bench setting. We first compare PersonaForge-trained models with their base counterparts and strong external systems, then analyze robustness, data-source effects, SOUL ablations, and transfer to an independent agent benchmark.

5.1 Setup

Model and training. We fine-tune **Qwen3.5-27B** (Qwen Team, 2025) and **MiMo-V2-Flash** (Xiaomi LLM-Core Team, 2026), covering both a 27B-scale model and a larger-scale MoE model, with full-parameter supervised fine-tuning (SFT) on the 6.3K conversations retained from the PersonaForge pipeline. Both models are trained on NVIDIA A100 GPUs for 5 epochs with a learning rate of 5×10^{-5} and a 128K maximum context length; training takes 2,550 GPU-hours total.

Evaluation protocol. We evaluate on the full 138 tasks of PersonaForge-Bench. For each task, we run 3 independent trials; all trials are scored and the final score is the average across trials and tasks. The user simulator is Claude Opus 4.6 (Anthropic, 2026); the LLM judge is DeepSeek-V4-Pro (DeepSeek-AI, 2026) with the task-specific scoring protocol described in Section 4. We additionally evaluate transfer on held-out tasks from CLAW-EVAL (Ye and collaborators, 2026). We report Overall scores and per-dimension breakdowns. A full per-domain score table is provided in Appendix E. We verify strict separation between training and evaluation data in Appendix F.

Baselines. We compare PersonaForge-trained models against their zero-shot base counterparts: Qwen3.5-27B (Qwen Team, 2025) and MiMo-V2-Flash (Xiaomi LLM-Core Team, 2026). We additionally include Qwen3.5-27B trained on real-user replay data produced from our internally collected user-agent interaction logs, using the same data scale as the PersonaForge SFT corpus, isolating the effect of synthetic versus authentic interaction trajectories. As external references, we include leading closed-source models (Claude Opus 4.6 (Anthropic, 2026), Gemini 3.1 Pro (Google DeepMind, 2026), GPT-5.5 (OpenAI, 2026)) and competitive proprietary models (MiMo-V2-Pro (Xiaomi AI Lab, 2026), Kimi-K2.5 (Kimi Team, 2026), MiniMax-M2.7 (MiniMax, 2026), GLM-5.1 (GLM Team, 2026)); all evaluated under an identical, rigorously controlled evaluation protocol.

5.2 Main Results

Table 3 reports the overall performance. On Qwen3.5-27B, the 27B-scale base model reaches a composite score of 76.2% with strong Tool Appropriateness (88.1%). PersonaForge yields a +4.1% composite gain, concentrated in Task Completion (+6.0%) and Response Quality (+6.8%), with

Model	Eff.	Tool	Compl.	Resp.	Overall
Claude Opus 4.6	83.0%	95.3%	86.5%	87.8%	88.2%
Gemini 3.1 Pro	83.8%	88.0%	85.0%	84.5%	85.9%
GPT-5.5	81.0%	93.5%	86.2%	87.0%	86.8%
MiMo-V2-Pro	81.7%	91.7%	84.6%	84.7%	85.7%
Kimi-K2.5	80.0%	87.8%	81.4%	85.0%	83.6%
MiniMax-M2.7	76.4%	88.3%	79.7%	78.0%	80.6%
GLM-5.1	73.6%	85.5%	70.3%	72.5%	75.5%
Qwen3.5-27B (Base)	74.0%	88.1%	70.6%	72.1%	76.2%
Qwen3.5-27B + PersonaForge	76.5%	89.1%	76.5%	78.9%	80.3%
MiMo-V2-Flash (Base)	53.2%	73.5%	53.5%	61.3%	60.4%
MiMo-V2-Flash + PersonaForge	69.8%	83.7%	75.5%	75.4%	76.1%

Table 3: Overall PersonaForge-Bench performance.

all four dimensions improving. This shows that PersonaForge provides a consistent interaction-training signal even when the base model already has capable tool-use behavior.

The gain is substantially larger on MiMo-V2-Flash, the larger-scale MoE model. PersonaForge improves the composite score from 60.4% to 76.1%, a $+15.7\%$ absolute gain (26.0% relative), with the largest improvement on Task Completion ($+22.0\%$, +41.1%). Tool Appropriateness also rises by $+10.1\%$, indicating that the additional capacity is not only reflected in final-answer quality but also in more effective use of the interaction and tool budget. Together, these results suggest a clear scaling pattern: PersonaForge improves both evaluated models, but the effect is more pronounced in the larger-parameter, large-scale MoE setting.

5.3 Ablation Studies

SOUL component ablation. Table 4 shows that PersonaForge's behavioral controls are important for downstream performance. On Qwen3.5-27B, removing connected memory reduces the composite score from 80.3% to 78.9%, with the largest drop on Task Completion (-3.5%). Removing flaws and traits further reduces performance to 78.2%, indicating that both latent context and persona-specific behavioral variation contribute to the training signal.

User-simulation ablation. We also compare PersonaForge against a real-user replay baseline built from our internally collected user-agent interaction logs. Replay preserves authentic human utterances but fixes the user script, so it cannot adapt to the assistant's actual responses. PersonaForge outperforms replay by $+4.7\%$ composite, with the largest gain on Task Completion ($+8.5\%$), showing that adaptive simulation provides value beyond exposing the model to human-written multi-turn transcripts.

We further verify that these gains are robust to scoring protocol and judge choice, and are not artifacts of response length or formatting bias; full analyses are provided in Appendix D.

5.4 Analysis

SOUL components contribute synergistically. Behavioral ablations across 40 diverse personas ($N = 240$ conversations) show that SOUL components jointly shape realistic user behavior (full results in Appendix G). Connected memory anchors progressive disclosure by giving the simulator latent context to reveal over time, while flaws and traits constrain how that disclosure unfolds. This explains why removing either component weakens downstream Task Completion: the model receives less signal about how to recover hidden intent through interaction.

Scale effects and interaction efficiency. Gains are positive across all eight evaluated domains (see Appendix E, which also describes the keyword-matching rule used for domain assignment). For MiMo-V2-Flash, the largest domain gains are Policy & Engineering (+29.6%) and Education & Healthcare (+23.7%); Qwen3.5-27B shows smaller but still broad gains, led by Policy & Engineering (+6.0%) and Technology & Data (+5.3%). Figure 11 quantifies the interaction-efficiency shift for the larger-scale MoE model: MiMo-V2-Flash after PersonaForge SFT uses 20.7% fewer turns, 9.2% fewer tool calls, and **54.2% fewer web_fetch calls**. Tasks reaching the 25-turn limit drop from 73 to 30 (-58.9%), while Task Completion increases by $+22.0\%$. Qwen3.5-27B follows the same direction: turns drop from 17.8 to 14.3 (-19.7%), tool calls from 18.5 to 13.8 (-25.5%), and 25-turn-

Ablation	Condition	Eff.	Tool	Compl.	Resp.	Overall
Training-data effect on Qwen3.5-27B						
Qwen3.5-27B	Base	74.0%	88.1%	70.6%	72.1%	76.2%
Qwen3.5-27B	+ PersonaForge	76.5%	89.1%	76.5%	78.9%	80.3%
Qwen3.5-27B	Δ	+2.6%	+1.0%	+6.0%	+6.8%	+4.1%
SOUL component ablation on Qwen3.5-27B						
Qwen3.5-27B	Full PersonaForge	76.5%	89.1%	76.5%	78.9%	80.3%
Qwen3.5-27B	−Connected Memory	75.5%	89.5%	73.0%	77.5%	78.9%
Qwen3.5-27B	−Flaws & Traits	74.8%	88.0%	70.5%	79.5%	78.2%
User-simulation ablation on Qwen3.5-27B						
Qwen3.5-27B	Real-User Replay	72.5%	88.5%	68.0%	73.0%	75.6%
Qwen3.5-27B	PersonaForge	76.5%	89.1%	76.5%	78.9%	80.3%
Qwen3.5-27B	Δ	+4.0%	+0.6%	+8.5%	+5.9%	+4.7%

Table 4: PersonaForge-Bench ablations on Qwen3.5-27B. MiMo-V2-Flash is reported in Appendix 11.

limit cases from 133 to 53 (−60.2%), while Task Completion increases by +6.0%. This shows that PersonaForge supervision improves both completion quality and interaction selectivity, with the effect most pronounced in the larger-parameter, large-scale MoE setting.

Gains generalize to out-of-distribution tasks. On held-out tasks from CLAW-EVAL (no overlap with PersonaForge-Bench), MiMo-V2-Flash improves from 59.0% to 69.1% (+10.1%), with Pass^3 rising from 10.5% to 23.7% and Pass@3 from 52.6% to 76.3%. Qwen3.5-27B shows more modest gains (68.9%→70.3%, +1.4%), mirroring its smaller PersonaForge-Bench improvement. These out-of-distribution results confirm that PersonaForge instills general interaction competencies rather than overfitting to benchmark rubrics.

6 Related Work

Agentic LLMs. Recent LLM systems increasingly act as agents that reason, call tools, and operate in external environments. ReAct (Yao et al., 2023) couples reasoning traces with actions, Toolformer (Schick et al., 2023) studies self-supervised tool use, and AutoGen (Wu et al., 2023) builds multi-agent workflows through conversation. Evaluation has likewise shifted from static QA to executable agent tasks: AgentBench (Liu et al., 2023) evaluates agents across interactive environments; SWE-bench (Jimenez et al., 2024) tests real GitHub issue resolution; WebArena (Zhou et al., 2024) and OSWorld (Xie et al., 2024) evaluate web and desktop operation; BrowseComp (Wei et al., 2025) focuses on persistent web browsing; and τ-bench (Yao et al., 2024) studies tool-agent-user interaction in constrained service domains. These benchmarks establish the importance of long-horizon tool use, but most evaluate agents against fixed tasks rather than training and evaluating them under realistic, progressively steering users.

Persona modeling and user simulation. Persona-based generation has become a common mechanism for diversifying synthetic behavior. PersonaHub (Chan et al., 2024) curates large-scale persona inventories, while Li et al. (2025) show that LLM-generated personas can introduce systematic biases. Generative Agents (Park et al., 2023) and CAMEL (Li et al., 2023) combine personas, memory, and role play for sandbox simulation, and domain-specific simulators model recommendation feedback (Cai et al., 2025) or personality-driven dynamics (Kroczek et al., 2025; de Paula et al., 2025). PersonaForge differs by using persona modeling not only to vary user identity, but also to control communication style, hidden context, progressive disclosure, and corrective behavior during live interaction with a deployed agentic system.

7 Conclusion

We presented PersonaForge, an effective framework for synthesizing realistic multi-turn user interactions, and PersonaForge-Bench, a benchmark for evaluating agentic systems under progressive user steering. Training on PersonaForge data consistently improves model performance across dense and MoE architectures and across model scales, and the improvements transfer to held-out agentic tasks beyond PersonaForge-Bench. These experiments suggest that realistic user simulation is a practical path for studying and improving agentic interaction, offering valuable insight into how models learn to manage incomplete intent, correction, and tool-intensive workflows.

Limitations

Persona design choices. We adopt MBTI types as the personality dimension not for psychometric validity but for practical controllability: the 16 discrete types provide an interpretable, well-documented space for steering communication style (e.g., INTJ→terse, ENFP→exploratory), and LLMs respond reliably to MBTI-based prompting. We do not claim that MBTI captures real personality structure; it serves as a behavioral diversity mechanism. Alternative trait models (e.g., Big Five) could substitute with similar diversity effects.

Ethical considerations. All interaction logs analyzed in Section 2 were collected with explicit opt-in consent from participants, who were informed that anonymized usage patterns would be used for research. No individual sessions are reproduced in the paper. For medical and legal benchmark tasks, all scenarios are fictional composites that do not correspond to real patients or cases, and model outputs are not intended as professional advice.

References

Anthropic. 2026. Claude opus 4.6 system card.

Joachim Baumann, Vishakh Padmakumar, Xiang Li, John Yang, Diyi Yang, and Sanmi Koyejo. 2026. Swe-chat: Coding agent interactions from real users in the wild. *arXiv preprint arXiv:2604.20779*.

Shihao Cai, Jizhi Zhang, Keqin Bao, Chongming Gao, Qifan Wang, Fuli Feng, and Xiangnan He. 2025. Agentic feedback loop modeling improves recommendation and user simulation. In *Proceedings of the 48th International ACM SIGIR Conference*.

Xin Chan, Xiaoyang Wang, Dian Yu, Haitao Mi, and Dong Yu. 2024. Scaling synthetic data creation with 1,000,000,000 personas. *arXiv preprint arXiv:2406.20094*.

Angel Felipe Magnossão de Paula, J. Shane Culpepper, Alistair Moffat, Sachin Pathiyan Cherumanal, Falk Scholer, and Johanne Trippas. 2025. The effects of demographic instructions on LLM personas. In *Proceedings of the 48th International ACM SIGIR Conference*.

DeepSeek-AI. 2026. DeepSeek-V4-Pro: Hugging face model repository.

GLM Team. 2026. Glm-5: From vibe coding to agentic engineering. *arXiv preprint arXiv:2602.15763*.

Google DeepMind. 2026. Gemini 3.1 pro model card.

Carlos E. Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. 2024. SWE-bench: Can language models resolve real-world GitHub issues? In *International Conference on Learning Representations*.

Kimi Team. 2026. Kimi k2.5: Visual agentic intelligence. *arXiv preprint arXiv:2602.02276*.

Leon O.H. Kroczek, Alexander May, Selina Hettenkofer, Andreas Ruider, Bernd Ludwig, and Andreas Mühlberger. 2025. The influence of persona and conversational task on social interactions with a LLM-controlled embodied conversational agent. *Computers in Human Behavior*, 172:108759.

Ang Li, Haozhe Chen, Hongseok Namkoong, and Tianyi Peng. 2025. LLM generated persona is a promise with a catch. *arXiv preprint arXiv:2503.16527*.

Guohao Li, Hasan Abed Al Kader Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. 2023. Camel: Communicative agents for "mind" exploration of large scale language model society. *Advances in Neural Information Processing Systems*, 36.

Weiwen Liu, Xu Zeng, Xinlong Cheng, Shuai Wang, Zishan Li, Yong Liang, Peng Zhao, Bo Zhang, and Hai-Tao Li. 2024a. Toolace: Winning the points of llm function calling. *arXiv preprint arXiv:2409.00920*.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, and 1 others. 2023. AgentBench: Evaluating LLMs as agents. *arXiv preprint arXiv:2308.03688*.

Zuxin Liu, Thai Hoang, Jianguo Zhang, Ming Zhu, Tian Lan, Shirley Kokane, Juntao Tan, Weiran Yao, Zhiwei Liu, Yihao Feng, and 1 others. 2024b. Apigen: Automated pipeline for generating verifiable and diverse function-calling datasets. *arXiv preprint arXiv:2406.18518*.

MiniMax. 2026. MiniMax M2.7: Early echoes of self-evolution.

OpenAI. 2023. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*.

OpenAI. 2026. GPT-5.5 System Card.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Joon Sung Park, Joseph C O'Brien, Carrie J Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. 2023. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*.

Qwen Team. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language models can teach themselves to use tools. *Advances in Neural Information Processing Systems*, 36.

Omar Shaikh, Shardul Sapkota, Shan Rizvi, Eric Horvitz, Joon Sung Park, Diyi Yang, and Michael S. Bernstein. 2025. Creating general user models from computer use. In *Proceedings of the 38th Annual ACM Symposium on User Interface Software and Technology (UIST)*.

Jason Wei, Nguyen Karina, Hyung Won Chen, Ethan Perez, Samuel R. Bowman, Jared Kaplan, Buck Shlegeris, and Liam Fedus. 2025. BrowseComp: A simple yet challenging benchmark for browsing agents. *arXiv preprint arXiv:2504.12516*.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, and 1 others. 2023. AutoGen: Enabling next-gen LLM applications via multi-agent conversation. *arXiv preprint arXiv:2308.08155*.

Xiaomi AI Lab. 2026. MiMo-V2-Pro: Hugging face model repository.

Xiaomi LLM-Core Team. 2026. Mimo-v2-flash technical report. *arXiv preprint arXiv:2601.02780*.

Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Tainuo Hua, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, and 1 others. 2024. OSWorld: Benchmarking multimodal agents for open-ended tasks in real computer environments. *arXiv preprint arXiv:2404.07972*.

Shunyu Yao, Noah Shinn, Parth Razavi, and Karthik Narasimhan. 2024. τ-bench: A benchmark for tool-agent-user interaction in real-world domains. *arXiv preprint arXiv:2406.12045*.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. ReAct: Synergizing reasoning and acting in language models. *International Conference on Learning Representations*.

Bowen Ye and collaborators. 2026. Claw-eval: End-to-end transparent benchmark for real-world agentic systems. *Preprint*, arXiv:2604.06132.

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

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P Xing, and 1 others. 2024. Lmsys-chat-1m: A large-scale real-world llm conversation dataset. *International Conference on Learning Representations*.

Shuyan Zhou, Frank F. Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Yonatan Bisk, Daniel Fried, Uri Alon, and 1 others. 2024. WebArena: A realistic web environment for building autonomous agents. In *International Conference on Learning Representations*.

A Real-Session Analysis Details

We use deterministic regular-expression matching to derive the descriptive statistics in Section 2. Before matching, user messages are normalized by lowercasing English text, converting full-width punctuation to half-width punctuation, and stripping URLs and code blocks. Task categories in Figure 2 are assigned from the first user message using priority-ordered regex families. Representative English keywords include `python`, `pandas`, `sql`, `excel`, `csv`, `plot`, `chart`, and `analy[sz]e` for data analysis; `docker`, `kubernetes`, `linux`, `server`, `deploy`, `nginx`, `ssh`, `cron`, and `devops` for system/DevOps; `code`, `bug`, `function`, `class`, `api`, `compile`, `test`, and `repo` for coding; `file`, `pdf`, `docx`, `rename`, `merge`, and `convert` for file operations; `wechat`, `slack`, `discord`, `bot`, `message`, and `notification` for IM/bot interactions; `write`, `rewrite`, `draft`, `email`, `summary`, and `polish` for writing; and `search`, `find`, `lookup`, `what is`, and `who is` for information retrieval. For Chinese sessions, we use translated counterparts of the same task and correction lexicons. If multiple categories match, we use the first category in this priority order; sessions with no category match are retained in aggregate statistics but excluded from the category plot.

We identify explicit user correction signals by matching follow-up user messages against correction-oriented expressions, including `wrong`, `incorrect`, `not right`, `mistake`, `error`, `failed`, `doesn't work`, `try again`, `fix`, `redo`, `you missed`, and `not what I meant`, plus translated Chinese counterparts maintained in the same lexicon. A session is counted as containing a user correction if any post-initial user message matches this set. We mark a correction as post-tool when the immediately preceding assistant step contains at least one tool call or tool return.

For Figure 4, correction messages are further classified by keyword families. *Context loss*

matches references to forgotten or previously stated information, such as `forgot`, `earlier`, `previous`, `as I said`, `context`, and `again`. *Wrong approach* matches method-level objections such as `wrong approach`, `not this way`, `should use`, and `instead`. *Format errors* matches output-shape requests such as `format`, `json`, `table`, `markdown`, and `schema`. *Instruction ignored* matches `ignore`, `didn't follow`, `as requested`, and `I asked`. *Hallucination* matches fabrication cues such as `hallucinat`, `made up`, `not real`, `fabricat`, and `source?`. Chinese messages are matched with translated equivalents of these five keyword families. When multiple failure families match the same correction, we count all matched families for the failure-mode histogram; the session-level correction rate remains binary.

B Framework Design Supplements

B.1 Interaction Structure

Figure 7 details the information flow between the PersonaForge user simulator and the target agentic system. The simulator's hidden context (SOUL prompt, connected memories, and task scenario) is never visible to the target system, which sees only the accumulated user messages and its own tool outputs.

B.2 Persona Coherence

Figure 6 contrasts two simulated conversations: one with incoherent persona dimensions (a non-technical user paired with expert-level jargon) that produces contradictory behavior, and one with coherent dimensions that yields consistent behavior throughout. This validates the coherence constraints described in Section 3.1.

B.3 Profile Construction

Figure 8 illustrates the Reverse Deep Construction pipeline described in Section 3.1: starting from real seed queries, an LLM infers persona profiles and generates task scenarios, converging on the same SOUL prompt structure. We also explored a complementary *forward* (persona-driven) path that samples a persona combination first and then generates a seed query grounded in that persona. While this provides direct control over user-type coverage, we found that Reverse Deep Construction alone produces sufficient diversity when seeded from large-scale real-query corpora, and all PersonaForge synthetic training data used in our experiments is generated via the reverse path.

Forward vs. Reverse: a case comparison. To illustrate why reverse-constructed profiles produce richer simulator state than forward-sampled ones, we present two matched cases on the same domain. Table 5 pairs a forward-sampled and a reverse-constructed profile in the finance domain; Table 6 does the same for the engineering domain. Each profile is summarized by its persona descriptor and three representative connected memory entries, the dominant component of the SOUL prompt and the single most impactful element identified in our ablation (Appendix G). Memories were originally generated in Chinese; we translate them to English here for readability and preserve domain-specific terms.

Observations. Three differences are consistent across both pairs and explain why the reverse path supplies a stronger training signal. (1) *Anchor specificity*: reverse-constructed memories carry real-world proper nouns and quantitative references (*Irish Revenue, CGT, Ireland SPV, constrained decoding, streaming*) that an LLM is unlikely to fabricate without grounding, whereas forward-sampled memories rely on generic concepts (PE/PB ratios, Redux, money-market funds). (2) *Connected-memory depth*: reverse memories converge on a single ongoing scenario viewed from complementary angles—an Ireland SPV acquisition seen as tax planning / cross-jurisdiction comparison / Revenue filing procedure—giving the simulator latent context to reveal across turns; forward memories are plausible but disjoint incidents that leave no hidden context to disclose progressively. (3) *User–task coherence*: reverse profiles tie the persona's professional identity to the same authentic task that produced the seed query (a finance manager facing an Ireland CGT question, an engineer with a real agent-pipeline output-format problem); forward sampling occasionally pairs a tangential occupation with the knowledge background (a real-estate agent issuing equity-valuation queries), which the simulator must paper over at generation time. These observations are consistent with the SOUL ablation in Appendix G, where connected memory ranks as the single most impactful component for downstream task completion.

B.4 Persona Dimension Distributions

Figure 9 shows the detailed distribution of each persona dimension in the synthesized dataset.

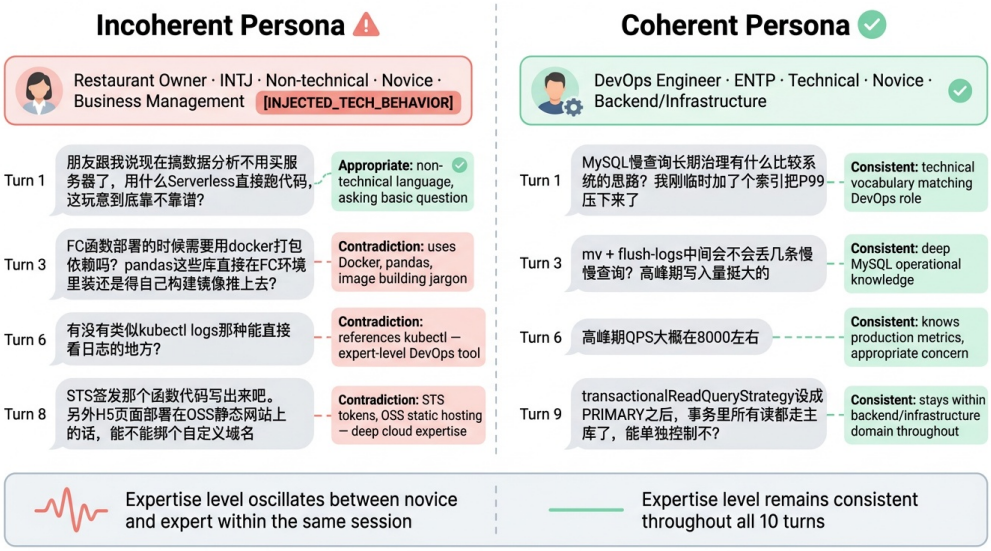

Figure 6: Effect of persona coherence on simulated user behavior. **Left**: inconsistent persona dimensions produce contradictory behavior within the same session (e.g., the user alternates between novice-level questions and expert jargon). **Right**: coherent persona dimensions yield consistent behavior throughout the trajectory.

B.5 Data Quality Analysis

Table 7 reports the effect of each quality control stage on the final dataset. Pre-generation filtering removes 12% of persona combinations due to coherence violations and 8% of seed queries for insufficient complexity. Runtime role-reversal detection flags 7% of sessions. Post-processing removes 4% of sessions due to degenerate patterns (infinite loops, repetitive exchanges). The listed rates are stage-level diagnostics rather than a strict multiplicative cascade; additional post-processing bookkeeping is not itemized in the table.

Stage	Removed	Ret.
Persona coherence	12%	88%
Query complexity	8%	92%
Role reversal	7%	93%
Degenerate patterns	4%	96%

Table 7: Data quality control pipeline statistics.

We note that runtime role reversal is the most subtle filter: the 1,500-character threshold was calibrated on real-user data, where 99.2% of follow-up messages fall below this cutoff. Raising the threshold to 2,000 characters drops flagged sessions to 3% but admits borderline role-reversed turns; lowering to 1,000 characters flags 12% but may discard legitimate long messages (e.g., users pasting code snippets).

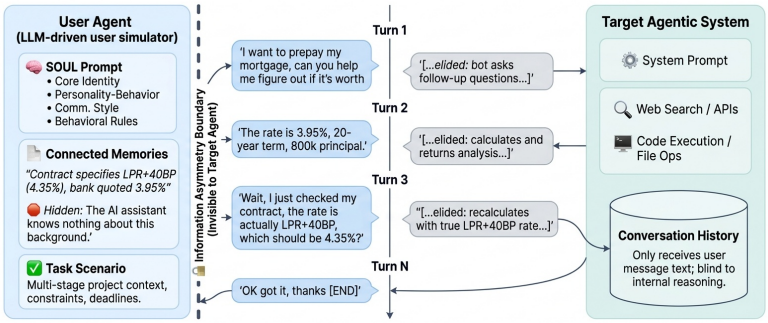

Figure 7: Information asymmetry between the PersonaForge (left) and the Target Agentic System (right). The user simulator's hidden context (SOUL prompt, connected memories, and task scenario) is never visible to the target system, which sees only user messages and its own tool outputs.

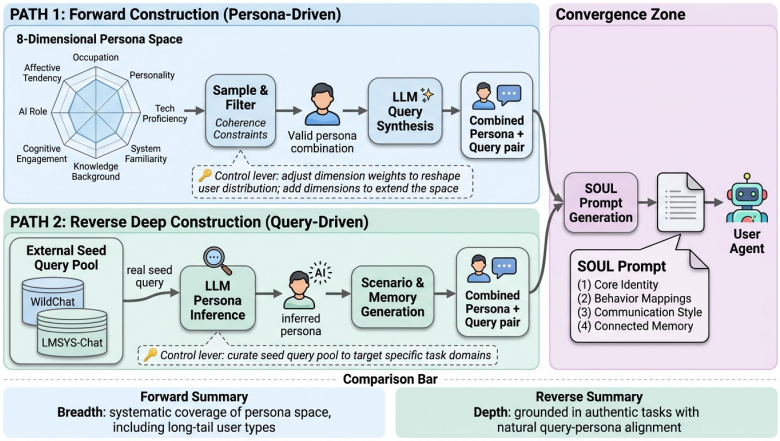

Figure 8: Reverse Deep Construction pipeline. Starting from real seed queries, an LLM infers persona profiles and generates task scenarios, producing a fully specified SOUL prompt structure.

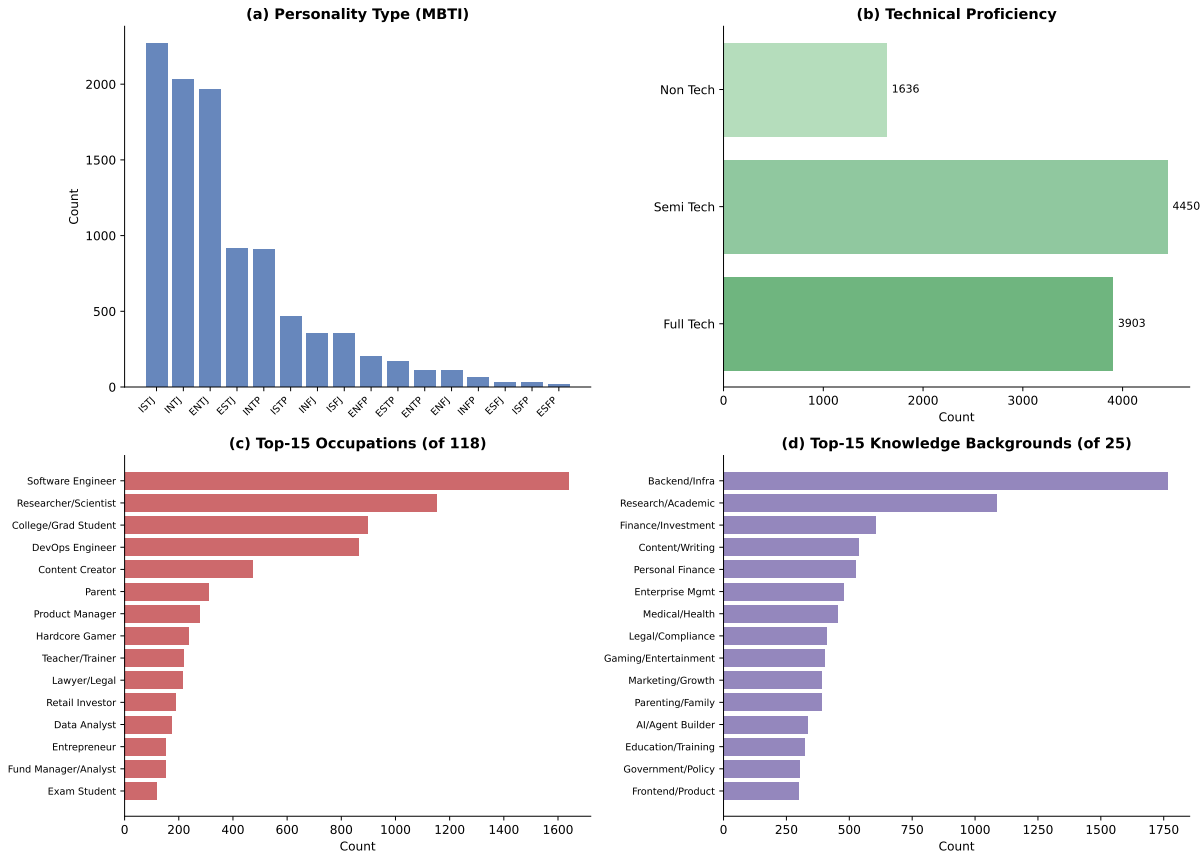

Figure 9: Distribution of persona dimensions. (a) MBTI types (16 types). (b) Technical proficiency (3 levels). (c) Top-15 occupations (of 70). (d) Top-15 knowledge backgrounds (of 25).

	Forward (persona-driven)	**Reverse Deep Construction**
Persona	Realtor / Insurance broker · ISTP · Non-tech · Novice · Finance & Investment	Finance manager / Accountant · ISTJ · Semi-tech · Casual user · Finance & Investment
Memory 1	A colleague gave me an Excel template to compute holdings yield, but the result doesn't match what Tonghuashun shows and I can't read those formulas. I have one stock waiting on this to decide whether to add.	The company has a potential equity-acquisition target in *Ireland*; the boss wants a draft cross-border tax-structure proposal by Friday. I haven't dug into *Ireland CGT* or holding-company tax efficiency before, so I need to lay out a framework first and fill in details later.
Memory 2	I'm trying to take a position in a pharma stock; let me have the AI pull PE, PB, and recent net-profit growth so I can decide.	Last week's review meeting flagged that overseas expansion may involve Ireland; I want a side-by-side of corporate income tax, capital gains tax, and dividend withholding across *Ireland, Singapore, and Hong Kong*.
Memory 3	A client just decided on a flat and asked where to park 800K of down-payment money for two months—safe, redeemable any time.	Our East-China project is now committed to an *Ireland SPV*. I need to clarify the registration and filing procedure with *Irish Revenue*: registration requirements, which taxes the online system handles, and *CGT filing timing*.

Table 5: Profile comparison in the finance domain. The reverse-constructed profile's connected memories converge on a single authentic business scenario (an Ireland SPV acquisition) seen from three complementary angles, while the forward-sampled profile's memories are plausible but disjoint incidents.

	Forward (persona-driven)	**Reverse Deep Construction**
Persona	Product Manager · ISFP · Semi-tech · Heavy user · Frontend / Product	Software Engineer · INTJ · Tech-expert · Heavy user · AI Agent Builder
Memory 1	Tomorrow I sync with the designer on the user-settings page interaction. I have a rough left-tab / right-content layout in mind, but I haven't worked out per-module fields and states.	Just hit a tool-calling JSON instability problem in the agent pipeline—occasional extra fields and nested misalignment crash downstream parsing. I need to add a hard prompt-level constraint and want a structured-output design quickly.
Memory 2	Frontend wants to switch state management from *Redux to Zustand* for "lighter weight." I know roughly what Redux does but only know Zustand by name.	I've been patching structured output with regex post-processing—too brittle. I should systematically look at *constrained decoding* and design a general layer that fits the existing architecture.
Memory 3	A new feature launches soon and needs a first-time onboarding paragraph; my draft reads too dry, so I want different tones to choose from.	Next week's architecture review will discuss output-format constraints for agents; I need a horizontal comparison of structured-output tools—*constraint granularity, compatibility with mainstream inference frameworks, streaming support*—no concept primers.

Table 6: Profile comparison in the engineering domain. The reverse-constructed profile is grounded in a specific technical pain point (LLM agent output reliability) with concrete domain vocabulary; the forward-sampled profile's memories are plausible but unrelated peripheral tasks of a product manager.

C Rubric Design and Examples

Each task in PersonaForge-Bench is accompanied by a detailed rubric specifying **MUST_ASK** clarification items, reference calculations with tolerance windows (e.g., ±100 yuan, ±0.5%), embedded contradictions or traps, and step-by-step evaluation chains.

C.1 Benchmark Composition

PersonaForge-Bench contains 138 tasks across Chinese and English scenarios: 110 tasks are in Chinese and 28 tasks are in English. The language label is specified in each task file and is used only to define the user-facing scenario and rubric language; the same multi-turn evaluation protocol and four scoring dimensions are applied to both subsets.

C.2 Benchmark Design Principles

Every task requires sustained multi-turn interaction and is designed around three principles. *Progressive information disclosure*: the user persona reveals details only when the assistant asks the right questions. *Embedded contradictions*: deliberate inconsistencies test whether the assistant can detect and reconcile conflicting information. *Domain-specific rubrics*: each task specifies required clarification items, reference calculations with tolerance windows, and step-by-step decision chains.

C.3 Scoring Dimensions

Each task is scored along four dimensions, each weighted equally at 25%, motivated by the real-user interaction patterns identified in Section 2. **Interaction Efficiency** measures how many conversation turns and tool calls are required to reach a satisfactory answer; fewer turns with comparable quality score higher. **Tool Appropriateness** evaluates whether the assistant selects the correct tools for each sub-task (programmatic computation via Bash, web search for current information, web_fetch for verification, file operations for document manipulation), whether search queries are well-crafted, and whether tool calls succeed on the first attempt. **Task Completion** assesses whether the assistant delivers a final, actionable answer covering all user requirements, discovers hidden needs through follow-up questions, and detects embedded contradictions in the user's information. **Response Quality** evaluates structural clarity (tables, headings, bullet points), factual and numerical accuracy, and directness, specifically whether the response gets to the point without unnecessary preamble.

C.4 Task-Specific Scoring Protocol

Scoring is *task-specific*: each of the 138 tasks includes a hand-crafted `judge_rubric` defining domain-appropriate evaluation criteria and a `reference_solution` specifying expected clarification items, correct calculations with tolerance windows, and step-by-step decision chains that a competent assistant should follow. When scoring a conversation, the LLM judge receives (1) the task's original user prompt, (2) its reference solution as ground truth, (3) its domain-specific evaluation criteria, and (4) the four generic dimension rubrics. This design ensures that the judge evaluates each conversation against what a correct answer *for that specific task* should look like, rather than applying a one-size-fits-all rubric that cannot account for domain-specific requirements such as detecting a 4.35% vs. 3.95% interest rate contradiction or verifying drug interaction risks for a renally impaired patient.

C.5 Design Rationale

The four scoring dimensions are directly motivated by the real-world patterns in Section 2. **Interaction Efficiency** penalizes inefficient tool use in multi-turn sessions, where models average 99.3 tool calls and 36.0% of correction signals occur after tool execution. **Tool Appropriateness** targets the wrong-approach errors (2.1K occurrences) and instruction-ignored failures (714) identified in our failure analysis, rewarding correct first-attempt tool selection. **Task Completion** reflects the inherent incompleteness of real queries; users progressively disclose requirements, so the assistant must discover hidden needs through follow-up. **Response Quality** addresses the mismatch between terse user commands (63% imperative, median 82 characters) and verbose model outputs, rewarding directness and structural clarity. By weighting efficiency and tool precision at 50% combined, the benchmark aligns with the terse, correction-heavy patterns of real usage rather than the verbose style of synthetic data.

C.6 Task-Specific Rubric Examples

We provide three representative task rubrics to illustrate how the four general dimensions are instantiated with task-specific criteria and weights.

Example 1: Mortgage Prepayment Analysis (C01zh). The user seeks financial advice on pre-

paying a mortgage. The task contains an embedded contradiction: the user claims a 3.95% interest rate, but the contract specifies LPR+40BP (actual rate: 4.35%). The assistant must collect financial details, detect the discrepancy, compute two repayment strategies (shortening term vs. reducing monthly payment), and recommend an approach matched to the user's cash flow.

Example 2: Design Spec Documentation Tools (C074en). The user needs actionable tool recommendations for auto-generating design specification documentation from Figma files (interaction behaviors, state descriptions, and edge cases). The assistant must balance research thoroughness against practical conciseness: the user needs curated top picks with setup instructions, not an exhaustive academic survey.

Example 3: Container Orchestration Architecture (C11zh). A DevOps engineer plans a microservice migration to Kubernetes. The assistant must assess the user's infrastructure scale, avoid over-engineering (e.g., not every team needs a full service mesh), and deliver a phased rollout plan.

Rubric summary. Interaction Efficiency (25%) penalizes generic "here are all the CNCF landscape tools" responses in favor of right-sized recommendations. Tool Appropriateness (25%) rewards Bash-based resource estimation and targeted web_search for tool compatibility. Task Completion (25%) requires concrete migration steps with per-phase checkpoints. Response Quality (25%) expects architecture diagrams or structured concept maps where helpful, with phased plans and clear completion criteria per phase.

D Judge Reliability and Robustness Analysis

Human validation of judge scores. To validate the LLM judge's reliability, we conducted a human audit on 60 scored conversations drawn to cover all four dimensions, both models (Base and SFT), and a spread of score levels. Three domain-expert annotators independently scored each conversation using the same dimension-specific rubrics and five-point scale as the judge. The inter-annotator agreement was $\kappa_{hh} = 0.81$ (Cohen's weighted κ, quadratic weights, averaged over all annotator pairs), providing a human-performance baseline. Judge–human agreement, measured by the weighted Cohen's κ between the judge and the averaged human ratings, was $\kappa_{jh} = 0.79$, closely approaching the inter-annotator baseline. The Pearson correlation between judge and averaged human composite scores was $r = 0.84$ ($p < 0.001$). Both metrics confirm the LLM judge as a reliable proxy for human quality assessment.

Scoring protocol robustness. To verify that our findings are not an artifact of the task-specific scoring protocol, we also evaluated all models using a generic four-dimension rubric without task-specific reference solutions or domain criteria. Both protocols agree on the direction and magnitude of improvement (generic $\Delta = +13.0\%$, task-specific $\Delta = +15.7\%$), with a Pearson correlation of $r = 0.94$ between the two scoring methods.

Cross-judge validation. To further verify that our scoring results are not an artifact of the specific judge model, we re-scored all 138 Flash tasks across 3 trials (both Base and SFT) using Gemini 3.1 Pro as an independent second judge with identical prompts and rubrics. The two judges exhibit strong agreement on composite scores (Pearson $r = 0.84$, $n = 828$ scored traces). Critically, both independently estimate the PersonaForge gain at nearly identical values: $+15.7\%$ (DeepSeek-V4-Pro) vs. $+15.6\%$ (Gemini 3.1 Pro), with a per-task delta correlation of $r = 0.70$ after averaging over 3 trials. These results confirm that the observed improvement is robust to the choice of judge model.

Building on this validation, we further verify that the observed gains are not an artifact of response length or formatting bias, a concern given that SFT increases assistant output by 57.0%. We conduct three additional robustness checks.

Length-stratified analysis. Table 9 partitions all scored responses from the 3 evaluation trials into four length quartiles and computes the SFT − Base composite difference within each bin. The sample distribution across bins reflects the behavioral difference between models: Base traces are concentrated in shorter bins (144 in Q1 vs. 63 for SFT), while SFT traces predominate in longer bins as it produces more substantive responses (165 in Q4 vs. 42 for Base). Contrary to the length-bias hypothesis, the SFT advantage is *largest* in the shortest quartile ($+24.8\%$, Q1) and remains positive in every quartile. If the judge merely rewarded longer responses, the gain would be concentrated in longer bins rather than peaking in Q1.

We further verify that the Q1 gain is not an artifact of incomplete or ceiling-limited short traces. After excluding traces with zero assistant output or turn counts at the 25-turn ceiling, the clean Q1

Dimension	Key Criteria (C01zh)	Wt.
Interaction Efficiency	$\leq$ 12 turns, $\leq$ 25 tool calls. Penalizes repeated recalculations or searching for general mortgage knowledge.	0.25
Tool Appropriateness	Bash (Python) for amortization computation; web_search only to verify bank-specific penalty policies. No manual arithmetic.	0.25
Task Completion	Covers both repayment strategies with total interest saved. Provides a concrete, personalized recommendation.	0.25
Response Quality	Clean comparison table. Correctly identifies 4.35% vs. 3.95% contradiction. No preamble. Proactively flags LPR discrepancy.	0.25
Dimension	**Key Criteria (C074en)**	**Wt.**
Interaction Efficiency	$\leq$ 12 turns, $\leq$ 15 tool calls. Uses search $\rightarrow$ fetch to verify a focused set of tools rather than exhaustively enumerating every option.	0.25
Tool Appropriateness	web_search to discover candidate tools; web_fetch on official docs/GitHub to verify maintenance status and features.	0.25
Task Completion	Delivers top 3–5 picks with justification, setup instructions, and a reusable per-component spec template.	0.25
Response Quality	Scannable output: comparison table, clear headings, bullet takeaways. Factual claims are source-verified. Direct and concise.	0.25

Table 8: Task-specific rubric examples. **Top**: C01zh (mortgage prepayment analysis). **Bottom**: C074en (design spec documentation tools).

subset (45 Base, 39 SFT traces) yields an even larger SFT advantage of $+29.6\%$ (Base 48.1% vs. SFT 77.7%), confirming that the Q1 gain reflects genuine quality differences rather than comparing incomplete traces against successful ones.

Length Bin	#Base	#SFT	Base	SFT	Δ
Q1 (<7.7K chars)	144	63	37.1%	61.8%	+24.8%
Q2 (7.7–13.9K)	117	90	71.8%	80.0%	+8.2%
Q3 (13.9–22.8K)	111	96	76.6%	83.8%	+7.2%
Q4 (>22.8K chars)	42	165	65.8%	74.9%	+9.1%
All	414	414	60.4%	76.1%	+15.7%

Table 9: Length-stratified composite scores (138 tasks, 3 scored traces per model per task).

Regression analysis. We fit a linear model composite $= \beta_0 + \beta_1 \log(\text{length}) + \beta_2 \cdot \text{is_SFT}$ over all 828 scored traces (adjusted $R^2 = 0.33$). The length coefficient is $\beta_1 = +0.056$ ($p < 0.001$, SE = 0.006), indicating that each e-fold increase in response length corresponds to a 0.056-point score increase. The SFT identity effect is $\beta_2 = +0.087$ ($p = 0.001$, SE = 0.026), meaning that, after controlling for response length, SFT traces score 0.087 points higher on average. Length and SFT identity together explain 33% of the variance; the SFT effect accounts for the majority of the composite improvement (raw $\Delta = +0.157$), with a complementary but smaller contribution from increased response length. Even after adjusting for length, the SFT improvement remains statistically and practically significant.

Structure and formatting bias. Table 10 stratifies scores by two binary surface features: whether the response contains Markdown tables and whether it uses section headings. In every sub-group, the SFT advantage remains positive and similar in magnitude. Notably, the base model already employs tables (78% of traces) and headings (88%) at high rates; SFT modestly increases these proportions (93% and 99%), but the within-group gains are comparable across feature values, indicating that structural formatting does not explain the composite improvement. The "No Headings" sub-group is excluded due to insufficient sample size (SFT $n = 4$).

Feature	Base (%)	SFT (%)	Base Overall	SFT Overall
Has Table	78%	93%	67.3%	77.6%
No Table	22%	7%	36.6%	54.4%
Has Headings	88%	99%	65.8%	77.0%

Table 10: Structure-stratified composite scores.

Taken together, these analyses establish that the reported gains are not artifacts of length or formatting bias. The SFT improvement persists within every length quartile (including a clean subset that excludes zero-output and turn-ceiling traces), within every structural sub-group, and after regression adjustment for response length ($\beta_{\text{SFT}} = +0.087$, $p = 0.001$). PersonaForge training produces genuine interaction quality improvements beyond what increased verbosity alone can explain.

E Additional Experimental Results

Condition	Eff.	Tool	Compl.	Resp.	Overall
Base	53.2%	73.5%	53.5%	61.3%	60.4%
+ PersonaForge	69.8%	83.7%	75.5%	75.4%	76.1%
Δ	+16.6%	+10.1%	+22.0%	+14.0%	+15.7%

Table 11: Partial MiMo-V2-Flash ablation on PersonaForge-Bench. We report the Base vs. PersonaForge SFT comparison here because the full SOUL and user-simulation ablations were conducted on Qwen3.5-27B.

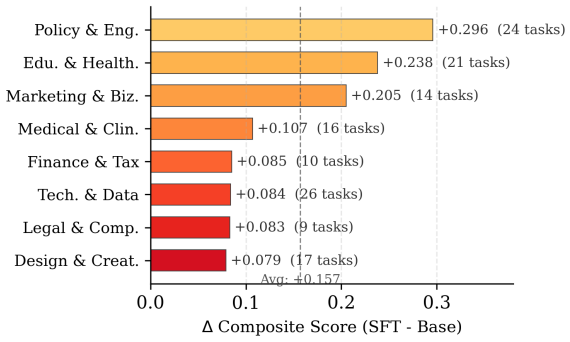

Figure 10: Per-domain Overall scores. Δ = SFT − Base. Dashed line indicates the average MiMo-V2-Flash gain (+15.7%).

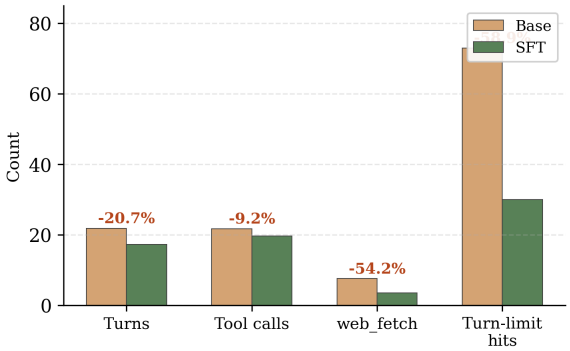

Figure 11: Interaction efficiency comparison. Base (brown) vs. SFT (green), with percentage reduction labeled.

Domain assignment. Per-domain results use deterministic keyword matching on task names. We lowercase each task name, match it against the domain keyword lists used by our evaluation script, and assign the task to the first matched reporting domain in a fixed priority order. The keyword families include: Design & Creative (`design`, `creative`, `ui_ux`, `accessibility`, `illustration`, `graphic`, `visual`, `typography`, `branding`); Technology & Data (`image_processing`, `container`, `distributed`, `api_`, `database`, `code_`, `programming`, `devops`, `sop_`, `server`, `deployment`, `kubernetes`, `docker`, `microservice`, `backend`, `frontend`, `algorithm`); Education & Healthcare (`academic`, `education`, `teaching`, `curriculum`, `research_method`, `thesis`, `literature`, `paper_`, `course_`, `student`); Marketing & Business (`marketing`, `business`, `sales`, `ecommerce`, `brand_`, `advertising`, `seo_`, `social_media`, `content_`, `real_estate`); Finance & Tax (`mortgage`, `finance`, `tax`, `investment`, `accounting`, `portfolio`, `income_tax`, `stock_`, `banking`, `insurance`, `loan`); Medical & Clinical (`clinical`, `pharmacy`, `medical`, `health`, `mental_health`, `diagnostic`, `patient`, `drug_`, `disease`, `surgery`); Legal & Compliance (`labor_law`, `legal`, `compliance`, `contract`, `regulation`, `patent`, `copyright`, `gdpr`, `audit`); and Policy & Engineering (`policy`, `government`, `public_`, `social_work`, `urban_`, `environmental`, `sustainability`, `ngo_`, plus engineering-oriented task names not captured by earlier keyword families). Scores are averaged over the three trials for each task, then averaged uniformly across tasks within each domain.

Tables 12 and 13 provide the full per-domain breakdown across all four evaluation dimensions for the base and PersonaForge SFT models. Task Completion and Response Quality show the largest absolute gains across most domains, consistent with PersonaForge's core corrective effect of converting tool-execution-heavy trajectories into actionable answers.

F Training–Evaluation Data Separation

PersonaForge-Bench and the PersonaForge training corpus are constructed through independent pipelines with no shared data. The 138 evaluation tasks in PersonaForge-Bench are curated from real consultation scenarios encountered in our internal daily usage of the deployed assistant. Each seed query was selected for domain representativeness, then manually revised and augmented with task-specific rubrics, embedded contradictions, and progressive-disclosure structures to form a rigor-

	Base (MiMo-V2-Flash)					PersonaForge SFT					
Domain	**Eff**	**Tool**	**Compl**	**Resp**	**Overall**	**Eff**	**Tool**	**Compl**	**Resp**	**Overall**	**Δ**
Design & Creative	62.1	78.2	50.3	72.9	65.9	63.5	75.3	77.1	79.1	73.8	+7.9
Technology & Data	62.9	76.3	65.6	65.0	67.5	72.1	82.7	76.9	71.6	75.8	+8.4
Education & Healthcare	50.0	69.5	58.1	71.7	62.3	79.3	91.4	86.4	87.1	86.1	+23.7
Marketing & Business	48.6	67.9	47.9	52.9	54.3	70.0	83.6	70.7	74.6	74.7	+20.4
Finance & Tax	58.5	87.0	54.0	49.5	62.3	69.0	91.0	58.0	65.0	70.8	+8.5
Medical & Clinical	61.6	83.4	61.4	65.6	68.0	72.5	83.8	86.3	72.5	78.8	+10.7
Legal & Compliance	68.3	75.2	76.7	83.0	75.8	75.6	85.6	82.2	92.9	84.1	+8.2
Policy & Engineering	26.9	60.0	26.0	37.5	37.6	60.6	79.4	63.3	65.5	67.2	+29.6
Total	53.2	73.5	53.5	61.3	60.4	69.8	83.7	75.5	75.4	76.1	**+15.7**

Table 12: Per-domain performance of MiMo-V2-Flash (%). All four dimensions plus Overall. Δ=Overall SFT−Base.

	Base (Qwen3.5-27B)					PersonaForge SFT					
Domain	**Eff**	**Tool**	**Compl**	**Resp**	**Overall**	**Eff**	**Tool**	**Compl**	**Resp**	**Overall**	**Δ**
Design & Creative	78.8	89.2	74.7	79.4	80.5	78.3	87.5	80.2	81.7	81.9	+1.4
Technology & Data	75.2	89.5	73.8	71.2	77.4	78.8	91.9	83.0	77.1	82.7	+5.3
Education & Healthcare	72.3	86.5	67.3	70.2	74.1	74.4	84.4	71.9	80.0	77.7	+3.6
Marketing & Business	79.3	91.5	79.9	76.3	81.7	83.5	91.8	78.9	83.6	84.4	+2.7
Finance & Tax	73.0	87.6	74.3	69.7	76.1	73.5	91.3	75.9	76.5	79.3	+3.2
Medical & Clinical	74.9	86.1	68.3	72.8	75.5	70.1	90.5	75.3	77.8	78.4	+2.9
Legal & Compliance	78.0	97.2	77.2	77.6	82.5	86.2	91.2	85.6	84.8	86.9	+4.5
Policy & Engineering	67.4	83.5	59.7	67.2	69.4	71.0	87.0	69.2	74.4	75.4	+6.0
Total	74.0	88.1	70.6	72.1	76.2	76.5	89.1	76.5	78.9	80.3	**+4.1**

Table 13: Per-domain performance of Qwen3.5-27B (%). All four dimensions plus Overall. Δ=Overall SFT−Base.

ous evaluation task. The training corpus, by contrast, draws its seed queries from public datasets (WildChat and LMSYS-Chat-1M), following the Reverse Deep Construction pipeline described in Section 3.1.

To verify the absence of data leakage, we performed an exhaustive overlap analysis between the candidate training pool and the evaluation set. We compared all first user messages from the candidate SFT training pool (6,333 records before near-duplicate removal) against all 138 benchmark task prompts using three matching criteria: exact string match, first-150-character fingerprint match, and substring containment. All three checks returned **zero matches**, confirming that no evaluation query appears in the candidate training pool in any form.

Semantic similarity analysis. Beyond surface-level matching, we computed pairwise semantic similarity between all 138 benchmark prompts and all 6.3K candidate training seed queries using TF-IDF character n-gram vectors (bigrams through 4-grams) with cosine similarity. The mean similarity across all $138 \times 6{,}333$ pairs is 0.010, confirming that the two sets are distributionally distinct. Two benchmark queries exhibited elevated similarity (≥ 0.80) to a candidate training query: one pair at 0.90 (a near-paraphrase about social media algorithm changes) and one at 0.89 (overlapping phrasing about real estate rent growth assumptions). Both corresponding candidate training records were removed before model training. After removal, the final SFT training set contains 6,331 records (reported as 6.3K elsewhere), and the maximum similarity between any benchmark–training pair drops to 0.75, which corresponds to topically related but textually distinct queries (e.g., two questions about e-commerce advertising that share domain vocabulary but differ in specific scenarios and parameters). No benchmark query exceeds the 0.80 near-duplicate threshold against any remaining training record.

G Ablation Details

To verify that the SOUL prompt components collectively contribute to simulation fidelity, we conduct an ablation study across **40 diverse personas** spanning 12 MBTI types, 3 technical proficiency levels, 20+ knowledge domains, and 2 languages (Chinese and English). For each persona, we generate conversations under six conditions: (A) full SOUL prompt, (B) personality traits removed, (C) human flaws removed, (D) behavior parameters removed, (E) minimal (only fixed rules + background), and (F) connected memory removed. All other variables (bot model, tools, seed query) are held constant across conditions for each persona, yielding $40 \times 6 = 240$ conversations.

Evaluation protocol. We measure six quantitative behavioral metrics computed directly from user messages: **Turn Count** (conversation length), **Progressive Disclosure** (ratio of mean follow-up message length to first message length; higher indicates gradual information revelation), **Lexical Diversity** (type–token ratio; lower indicates more consistent persona vocabulary), **Message Length** (mean user message length in characters), **Topic Switch Rate** (fraction of follow-up messages introducing unreferenced topics), and **Length Variability** (coefficient of variation of message lengths). We apply the Friedman test across all six conditions to assess overall differences, followed by paired Wilcoxon signed-rank tests with Cohen's d effect sizes for each ablation condition versus the full baseline.

Overall significance. Table 14 reports the Friedman repeated-measures test, confirming that ablation conditions produce significant behavioral differences across all 40 personas on all six metrics.

Metric	χ^2	p
Progressive Disclosure	25.61	$< 0.001^{***}$
Lexical Diversity	23.21	$< 0.001^{***}$
Message Length	17.17	0.004^{**}
Turn Count	14.25	0.014^{*}
Topic Switch Rate	11.52	0.042^{*}
Length Variability	11.93	0.036^{*}

Table 14: Friedman test across 6 ablation conditions ($N = 40$ personas). Significant differences confirm that SOUL prompt components collectively affect simulated user behavior.

Pairwise effects. Table 15 reports paired Cohen's d (Full $-$ Ablation; positive indicates the full SOUL scores higher) and Wilcoxon significance for each condition.

Cond.	Turn	Prog	Lex	Len	Top	#S
B.−Pers	$+.05$	$+.13$	$-.24^{*}$	$+.12$	$-.21$	1
C.−Flaw	$+.03$	$+.12$	$-.15$	$+.23$	$-.44^{*}$	1
D.−Parm	$+.00$	$+.08$	$-.13$	$+.07$	$-.28$	0
E. Min	$+.40^{*}$	$\mathbf{+.44}^{***}$	$\mathbf{-.53}^{***}$	$+.43^{**}$	$-.34^{*}$	5
F.−Mem	$-.09$	$\mathbf{+.31}^{**}$	$-.33^{*}$	$+.44^{**}$	$-.36^{*}$	4

Table 15: Paired Cohen's d vs. Full ($N = 40$). $^{*}p < .05$, $^{**}p < .01$, $^{***}p < .001$. Pers=Personality, Flaw=Flaws, Parm=Params, Min=Minimal, Mem=Memory.

Synergistic rather than additive effects. Removing any single component among personality, human flaws, or behavior parameters produces small effects ($|d| < 0.25$ on most metrics, ≤ 1 metric reaching significance). However, the Minimal condition, which removes all three simultaneously, yields the strongest degradation ($|d| = 0.34$–0.53, 5 metrics significant). This pattern indicates that the components provide overlapping behavioral constraints: each partially compensates for the others when one is absent, but the system degrades substantially when all are removed together.

Connected memory as the critical anchor. Memory removal is the only *single-component* ablation that produces multiple significant effects (4 metrics: Progressive Disclosure $d = +0.31$, $p = 0.003$; Lexical Diversity $d = -0.33$, $p = 0.032$; Message Length $d = +0.44$, $p = 0.002$; Topic Switch Rate $d = -0.36$, $p < 0.05$). Without hidden background context, users front-load all information in the first message (mean InfoGrowth ratio: 0.58 vs. 0.80 in Full) rather than progressively disclosing requirements across turns, eliminating the gradual need-discovery pattern that characterizes real user interactions (Section 2).

Progressive disclosure as the primary discriminator. The Progressive Disclosure metric shows the strongest overall effect in the Friedman test ($\chi^2 = 25.61$, $p < 0.001$) and the largest effect sizes in pairwise comparisons. The full SOUL prompt produces a mean InfoGrowth ratio of 0.80 (follow-up messages are 80% the length of the initial message, indicating substantial new information in later turns), while the Minimal condition drops to 0.54 ($p < 0.001$) and Memory removal to 0.58 ($p = 0.003$). This confirms that progressive information revelation, a hallmark of real user behavior, requires both the connected memory (which provides latent context to disclose) and the full persona specification (which constrains *how* disclosure unfolds).

Robustness across persona types. We tested whether persona attributes (MBTI type, technical proficiency, usage frequency) moderate ablation sensitivity using Kruskal–Wallis tests on per-persona divergence scores. No significant interactions were found (all $p > 0.2$), confirming that the SOUL prompt's contribution to simulation fidelity generalizes across the full persona space rather than being driven by specific persona subtypes.

Implications. The ablation results support a *synergistic* view of SOUL prompt design: personality traits, human flaws, and behavior parameters form a redundant ensemble that collectively maintains simulation fidelity, with connected memory serving as the irreplaceable anchor that drives goal-directed, progressively-disclosing interaction. This redundancy is an engineering strength: the sys-

tem degrades gracefully under partial component failure, while the strong Minimal-vs-Full contrast ($\chi^2 = 25.61$, 5 significant metrics) confirms that the full SOUL specification is necessary for realistic simulation.

Downstream training ablation. To verify that SOUL components affect not only simulation behavior but also downstream task performance, we train Qwen3.5-27B on data generated under three SOUL configurations and evaluate on PersonaForge-Bench. Table 16 reports the results.

Condition	Eff.	Tool	Compl.	Resp.	Overall
Full SOUL (baseline)	76.5	89.1	76.5	78.9	80.3
−Connected Memory	75.5	89.5	73.0	77.5	78.9
−Flaws & Traits	74.8	88.0	70.5	79.5	78.2

Table 16: Downstream impact of SOUL components on PersonaForge-Bench (%). Since the total Qwen3.5-27B gain is $+4.1\%$, the -1.4% and -2.1% drops represent 34% and 51% of the total improvement being lost. Task Completion is most sensitive to component removal.