

From Profiling to Synthesis: Benchmarking Implicit Behavioral Alignment in Personalized LLM Agents

Jiajia Song¹ Bobo Li¹ Haiwen Yi² Zibo Ji³ Meishan Zhang⁴

Hao Fei⁵ Min Zhang⁴ Mong-Li Lee¹ Wynne Hsu¹

¹National University of Singapore ²University of Toronto

³University of Minnesota Twin Cities ⁴Harbin Institute of Technology, Shenzhen

⁵University of Oxford

{jjiasong, libobo, dcsleeml, dcshsuw}@nus.edu.sg, hao.fei@bdi.ox.ac.uk

Abstract

Large Language Models have enabled increasingly capable autonomous agents, yet personalization remains critical for making such agents practically useful. Recent benchmarks have begun evaluating personalization in agents, but they largely rely on static preference snapshots, fixed interaction logs, or question answering over predefined user profiles. Such designs fail to capture the complexity of evolving user preferences and neglect preference-conditioned task execution—a discrepancy we term as the *knowledge-to-action gap*. To address this challenge, we introduce IBA-BENCH, a benchmark for implicit behavioral alignment constructed from longitudinal interaction histories that contain noise, implicit cues, and temporal inconsistencies. Unlike prior work, IBA-BENCH evaluates whether an agent can execute tasks while satisfying implicit user constraints inferred from historical interactions. We further propose IBA-AGENT, an agent framework that reconciles conflicting priorities through broad retrieval and trajectory-level alignment. Experiment results on IBA-BENCH show that effective personalization remains a significant challenge for state-of-the-art LLM agents, and the proposed IBA-AGENT substantially improves behavioral alignment in complex scenarios across nine application domains.

1 Introduction

Large Language Models (LLMs) (Achiam et al., 2023; Team, 2025; Guo et al., 2025) have evolved from conversational chatbots into autonomous agents capable of executing complex real-world tasks (Yang et al., 2018, 2024). This shift makes personalization essential as users now expect assistants that can adapt to their preferences and constraints rather than simply generic tool executors (Samuel et al., 2025). Achieving such personalization is a challenge because user preferences are rarely explicit; instead, they dynamically evolve and remain implicitly embedded within lengthy

interaction histories (Zhang et al., 2025). Capturing and synthesizing these dynamic signals across long interaction histories is needed for building the next-generation personalized agents.

Recent works have introduced various personalized agents and evaluation frameworks. Early efforts extracted static profiles for response generation (Joko et al., 2024), collapsing preferences into flat tags and ignoring temporal dynamics. Subsequent benchmarks incorporated raw dialogue histories to preserve realism. For instance, HiCUPID (Mok et al., 2025) retains conversational trajectories, while PersonaBench (Tan et al., 2025) models dynamic attribute evolution. Task-oriented evaluations like PersonaLens (Zhao et al., 2025) replace interaction histories with summarized abstracts. However, all these evaluations predominantly rely on extraction-based Question Answering (QA). As a result, they fail to adequately measure an agent’s ability to integrate implicit, evolving user preferences into complex actions and decisions. We refer to this limitation as the *knowledge-to-action gap*. This gap highlights a stark disconnect: an agent may possess crucial knowledge yet fail to apply it during task execution.

Consider a user with a historical preference for spicy food. Recent interaction history reveals that the user has undergone a wisdom tooth extraction (Figure 1). When asked to order dinner, current agents typically order a spicy meal based on the static user preference profile. Yet, the same agent is able to recall that the user’s tooth extraction indicating that it possesses the knowledge. Since the dental event and dining request seems unrelated, agents frequently overlook this connection unless explicitly prompted. An ideal personalized agent would implicitly synthesize these signals and proactively recommend soft food. Bridging this gap requires moving beyond static profiling to dynamic synthesis, where agents infer implicit constraints and reconcile conflicts autonomously.

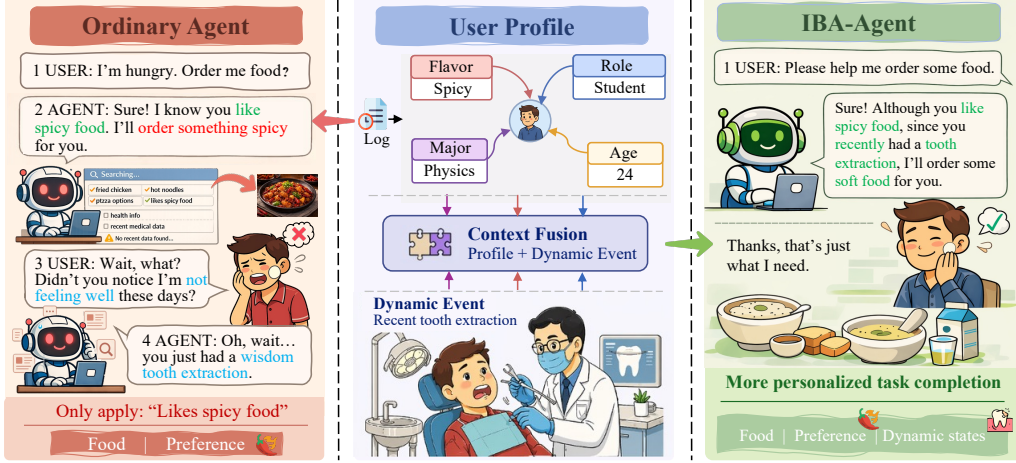


Figure 1: Personalization beyond QA: knowledge-to-action gap between preference inference and task execution.

In this work, we propose **IBA-BENCH**, a benchmark specifically designed to assess implicit behavioral alignment in agents. The primary objective of IBA-BENCH is to evaluate and bridge the knowledge-to-action gap. At the data level, we construct longitudinal interaction histories containing implicit preference cues, dynamic attributes, and conflicts between long-term traits and short-term states. At the task level, we move beyond explicit QA and simple preference matching by providing agents with full interaction records and the user request. The agent carries out the request and we evaluate whether the request is successfully executed and that the implicit constraints inferred from the interaction history are satisfied.

We also introduce **IBA-Agent**, a personalized agent framework to bridge the gap between historical preference understanding and active task execution. IBA-Agent consists of two key components: (i) Deep Retrieval, which isolates relevant behavioral cues from noisy interaction histories, and (ii) Broad Thinking and Deep Alignment, which synthesizes these signals into actionable constraints and planning logic prior to generation.

Extensive experiments on IBA-BENCH reveal that standalone LLMs (Achiam et al., 2023; Guo et al., 2025) and even strong general-purpose agents such as Claude Code and Hermes Agent struggle significantly on these synthesis-heavy execution tasks. These results empirically validate the *knowledge-to-action gap*. In contrast, IBA-Agent substantially improves behavioral alignment in complex scenarios, demonstrating the effectiveness of explicit trajectory-level synthesis. Our main contributions are:

- We advocate a shift in personalized agent research from static preference profiling to dy-

namic preference synthesis, and identify the knowledge-to-action gap as a key bottleneck in translating inferred knowledge into effective task execution.

- We introduce IBA-BENCH, a benchmark that evaluates implicit behavioral alignment through concrete task execution in realistic settings characterized by dynamic, conflicting, and noisy contexts.
- We propose IBA-Agent, a personalized agent framework that combines deep retrieval with trajectory-level synthesis, providing a strong empirical baseline for building agents capable of personalized reasoning.

2 Related Work

LLM-Based Agents have emerged as an active research area with the rapid advancement of LLMs’ reasoning (Wei et al., 2022), planning (Valmeekam et al., 2023), and tool use (Schick et al., 2023; Li et al., 2026). Compared to generic agents, these agents seek to operate in dynamic and diverse interaction scenarios by aligning decision-making with user-specific characteristics (Wang et al., 2024).

Current approaches typically introduce personalization through explicit user modeling (Zhang et al., 2025; Afzoon et al., 2024), memory or retrieval-augmented mechanisms (Ram et al., 2023), and adaptation based on user feedback (Stiennon et al., 2020) with the goal of capturing both long-term user preferences and evolving interaction contexts (Westh  u  er et al., 2025; Park et al., 2023). Existing research remains largely method-centric, focusing on personalization mechanisms and model design rather than systematic evaluation of personalization capabilities (Mohammadi et al., 2025; Hao et al., 2025).

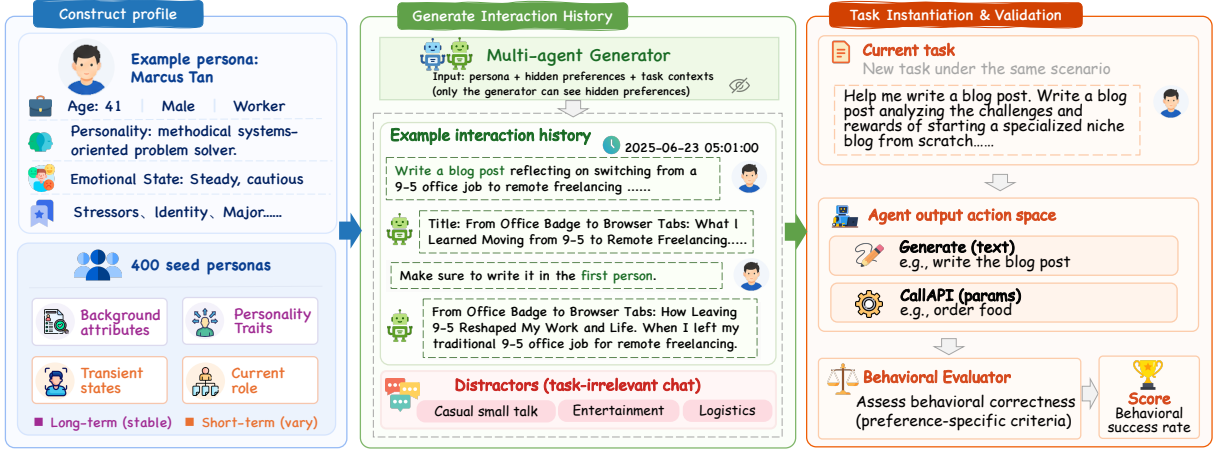


Figure 2: Benchmark construction flow chart.

Benchmarks for Personalized Agents are derived from general-purpose evaluation settings (Liu et al., 2024b), including QA, tool use, and short-horizon planning tasks (Hu and Shu, 2023). These benchmarks measure task success or accuracy under static assumptions and task-level metrics (Zhu et al., 2025). While effective for assessing general agent capabilities, they offer limited support for evaluating personalized behaviors (Hao et al., 2025). Existing personalization benchmarks operationalize personalization as profiling, either by extracting user attributes via QA or by aligning responses under explicit and static preference specifications (Mok et al., 2025; Tan et al., 2025; Zhao et al., 2025). These settings under-specify the core challenge faced by agents: synthesis, which involves integrating implicit, potentially conflicting, and time-varying user signals into task execution over interaction histories. Motivated by these limitations, we introduce IBA-BENCH. A comparison with prior benchmarks is summarized in Table 1.

Benchmark	History	Dynamic	Implicit	Task	Behavior
HiCUPID	✓	✗	✓	✗	✗
PersonaBench	✓	✓	✓	✗	✗
PersonaLens	✗	✗	✓	✓	✗
IBA-BENCH (Ours)	✓	✓	✓	✓	✓

Table 1: Comparison of Persona-related Benchmarks. Our proposed IBA-BENCH is the only benchmark that covers all dimensions.

3 Benchmark Construction

To bridge the gap between static profiling and dynamic synthesis, we construct IBA-BENCH through a multi-stage pipeline designed to simulate the noise, implicitness, and conflicts inherent in real-world personalization. An IBA-BENCH in-

stance is defined by $(\mathcal{H}_u, x, s, \mathcal{Y}, \mathcal{E})$, where $\mathcal{H}_u$ is the interaction history of user u , x is the current task query under scenario s , $\mathcal{Y}$ is the action space, and $\mathcal{E}$ is the behavioral evaluator. The action space $\mathcal{Y}$ includes two types:

- **Generate(text)**: produce personalized textual outputs such as writing or planning;
- **CallAPI(params)**: generate structured parameters for tool-mediated actions such as ordering food or making a booking.

Given $\mathcal{H}_u$, x , and s , the agent must infer the historical signals most relevant to the current task and determine an action $y \in \mathcal{Y}$, where $y = f_\theta(\mathcal{H}_u, x, s)$.

The evaluator $\mathcal{E}$ assesses the behavioral correctness of the final action using type-specific criteria such as generation quality and API-parameter matching. Detailed scoring definitions and examples are provided in Appendix A.2. Figure 2 shows the construction process of IBA-BENCH.

Construct Personas. We define 400 seed personas as the basic units for modeling user-specific variation. Each persona is multi-dimensional and formalized as $p = (B, T, E, R)$, where B denotes background attributes, T personality traits, E transient states (e.g., stress or urgency), and R the current professional or social role. Note that B and T are long-term attributes that remain relatively stable over time, while E and R are short-term preferences that may change and contradict earlier behavior. This formulation allows a persona to capture personal traits, situational states, and professional or social roles, enabling coherent yet diverse behaviors across scenarios.

Generate Interaction History. To mimic realistic interaction history, we use a multi-agent framework to generate interactions where informative prefer-

Domain	Scale Stats				# Evidence					# Constraints				
	S	D	P	I	1	2	3	4	5+	1	2	3	4	5+
Writing and Design	16	55	10	750	271	267	83	61	68	66	199	282	160	43
Work	12	53	17	1,155	394	380	165	111	105	30	161	338	321	305
Daily Consumption	8	39	15	1,086	511	290	217	36	32	46	182	401	276	181
Planning	10	39	16	1,151	532	329	182	42	66	41	206	489	386	29
Exercise and Health	4	16	20	781	325	247	113	41	55	11	54	173	252	291
Transportation	4	14	14	518	288	117	112	0	1	21	67	169	197	64
Medical Services	3	11	15	524	290	113	121	0	0	18	90	231	185	0
Leisure Activities	3	9	14	333	137	104	62	16	14	12	60	148	113	0
Information Management	6	26	17	664	293	217	94	29	31	23	73	185	161	222
Overall	66	262	15	6,962	3,041	2,064	1,149	336	372	268	1,092	2,416	2,051	1,135

Table 2: Statistics of domains in the benchmark. **S**: number of scenarios in each domain; **D**: total number of preference dimensions across scenarios; **P**: integer-truncated average number of preference values per scenario; **I**: number of task instances. The columns under **#Evidence** and **#Constraints** report the distributions of task instances by the number of associated evidence items and preference constraints in each instance, respectively. In both column groups, “5+” indicates five or more. For each domain, the counts under **#Evidence** and **#Constraints** each sum to **I**.

ence evidence is embedded within large volumes of noisy conversational content. Rather than being explicitly stated, user preferences are revealed through accumulated behavioral patterns. For example, instead of directly stating a preference such as “I prefer concise reports,” a user may repeatedly shorten verbose drafts or prioritize rapid responses under time pressure. Consequently, agents must infer preferences from behavioral evidence from the interaction history.

To increase realism, we inject task-irrelevant interactions, with task-relevant interactions occurring only once per 100 turns on average, causing informative signals to be deeply buried within noisy histories. This signal-to-noise imbalance challenges agents to distinguish meaningful behavioral evidence from distractors.

Task Instantiation and Validation. We use minimal prompts to generate tasks that require agents to execute concrete actions based on the interaction history. Task content is newly authored but domain-aligned to prevent trivial reuse. To simulate real-world agent operations, we include 2,280 API-actionable instances spanning 16 scenarios with parameterized API-call outputs.

Table 2 shows the statistics of IBA-BENCH which comprises 6,962 task instances spanning 9 domains across 66 scenarios and 262 preference dimensions, generated using GPT-5.1 at an API cost of approximately \$5,000. The distribution is skewed toward non-trivial cases: 1,857 instances involve three or more behavioral evidence items, 5,602 instances contain at least three preference constraints (see Appendix A.1).

For quality control, we randomly sample 200

instances for validation. Across reasonableness, coherence, and persona consistency, human annotations achieve 80.2% agreement with a Gwet’s AC_1 of 0.70, while AI-based annotations achieve 94.0% agreement with an AC_1 of 0.93. Based on the validation feedback, we refine the prompts and regenerate problematic cases, revising about 10% of all instances. Annotation details are in Appendix B.1.

4 Method

In this section, we formalize implicit behavioral alignment as a preference-conditioned task execution problem and present IBA-Agent, an LLM-driven personalized agent framework for bridging the gap between historical preference understanding and active task execution, as shown in Figure 3.

4.1 LLM-Driven Personalized Agent

IBA-Agent takes the current task and the user’s historical interactions as input, and follows an iterative evidence-to-action process for personalized task execution. During this process, an LLM controller maintains the task state, observes the current context, reasons over accumulated tool outputs, selects and executes the next tool, and updates the context with new evidence or intermediate decisions. Through this loop, the agent identifies dynamic and implicit preference signals from the user’s history and aligns them with the requirements of the current task. The process terminates when the controller invokes a task-specific final output head to produce the personalized action. IBA-Agent implements this workflow with two complementary modules: DEEP RETRIEVAL and BROAD THINKING & DEEP ALIGNMENT, which we detail below.

Deep Retrieval. This module uses four retrieval tools to identify task-relevant preference evidence from the large interaction history $\mathcal{H}_u$ and dynami-

¹<https://huggingface.co/BAAI/bge-m3>

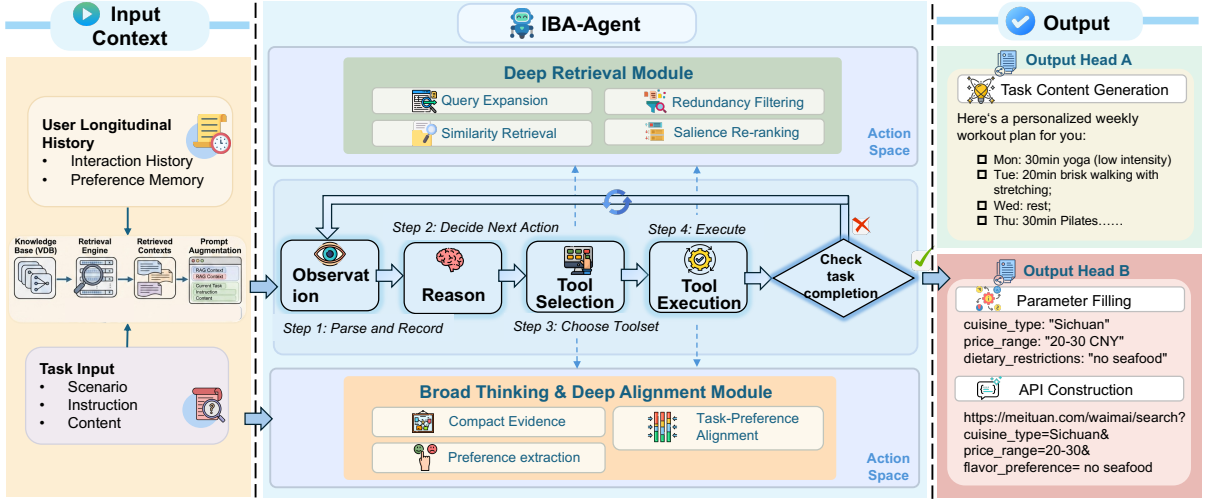


Figure 3: Overview of IBA-Agent for personalized task completion.

cally build the task-conditioned evidence context $\mathcal{C}_u(x)$ for downstream reasoning. Given a current task query x , the agent first retrieves passages using the original task description and then invokes query expansion to generate multiple task-aware sub-queries. As shown in Table 3, these sub-queries probe complementary preference facets. This design allows the same user history to yield task-specific evidence, such as budget trade-offs for purchase tasks and tone, formatting, and revision patterns for writing tasks. The similarity retrieval tool, implemented with BGE-M3, retrieves semantically relevant passages from $\mathcal{H}_u$. This enables the agent to recover implicit preference evidence even when the user has not stated a preference in a canonical form.

The module also provides evidence refinement tools. The redundancy filtering tool removes duplicated or highly overlapping retrieved passages using a content-hash heuristic, preventing repeated history snippets from dominating the evidence context. The saliency re-ranking tool uses an LLM-based scorer to prioritize passages with high-confidence preference signals. Each passage is assigned a discrete saliency score from 0 to 10, and passages scoring at least 5 are retained for downstream use; the scoring criteria are summarized in Table 12 in the Appendix. This weighting scheme limits over-interpretation of weak behavioral cues by weighting reliable explicit feedback more heavily than noisier implicit behavioral patterns.

Broad Thinking & Deep Alignment. This module uses a set of prompt-guided reasoning tools to transform the $\mathcal{C}_u(x)$ into a task-specific personalization plan. The compact evidence function first organizes the retained passages into a concise

QUERY EXPANSION PROMPT

You are a **search query generator** for personalized recommendations. Given a task query, generate diverse retrieval queries to uncover relevant user preferences from historical conversations.

Input: Task query information, including the scenario, instruction, and content.

Expansion Facets.

Facet	Retrieval focus
q^{style}	Tone, wording, expression habits, and preferred writing style.
$q^{structure}$	Format, headings, templates, bullet/paragraph preference, length, and detail level.
$q^{decision}$	Trade-offs such as cost vs. quality, time vs. thoroughness, or convenience vs. preference.
$q^{correction}$	Complaints, corrections, explicit rejections, and disliked suggestions.
q^{habit}	Repeated choices, default behaviors, and recurring preferences in the scenario.

Output: Return target query strings covering multiple facets. No preamble or explanation.

Table 3: Prompt-guided query expansion in the Deep Retrieval module. Given a task x , the generator produces a set of target queries $Q(x) = \{q^{style}, q^{structure}, q^{decision}, q^{correction}, q^{habit}\}$ to retrieve diverse preference evidence from the history.

evidence set by preserving the most informative historical interactions. The preference extraction function then reads this evidence with respect to the current task and extracts task-relevant preference signals, such as what the user asked for, corrected, rejected, repeatedly chose, or recently updated. The task-preference alignment function decides how the extracted preferences should guide the current task, producing an alignment plan: response style and structure for generation tasks and constraints or priorities for API-action tasks.

Model	Writing	Work	Daily	Plan.	Health	Trans.	Med.	Leis.	Info.	Overall
• <i>Baseline Models</i>										
Qwen2.5-7B-Instruct (Qwen et al., 2025)	52.7	51.0	77.3	55.5	67.1	65.2	75.0	40.1	37.1	58.3
DeepSeek-V3 (Liu et al., 2024a)	65.1	56.6	77.1	62.7	71.4	67.3	78.8	44.7	51.9	64.9
ChatGLM-4-9b-chat (Glm et al., 2024)	51.4	47.3	70.7	53.5	65.4	58.4	69.3	36.7	37.1	55.3
QwQ-32B (Qwen Team, 2025)	68.7	61.6	80.6	64.5	75.2	74.2	79.6	47.3	55.4	68.4
GPT-4o-mini	57.7	54.9	77.4	57.9	68.1	68.3	71.8	39.9	42.4	61.1
GPT-4o (OpenAI, 2024)	63.5	54.1	75.6	57.9	69.5	66.6	73.0	44.1	43.5	61.7
GPT-5-mini	70.5	63.9	76.3	69.8	78.3	70.8	80.2	49.8	63.2	70.1
GPT-5.1 (OpenAI, 2025)	72.9	62.4	79.9	65.6	73.9	70.6	78.0	56.8	58.1	69.2
• <i>Qwen3-4B-Instruct Variants</i>										
Qwen3-4B-Instruct (Yang et al., 2025)	65.2	60.4	77.8	62.8	73.4	80.0	82.9	52.6	54.1	67.6
IBA-Agent (Qwen3-4B-Instruct)	80.4	73.3	84.0	77.2	86.1	82.3	89.5	47.5	69.2	78.1
IBA-Agent (Qwen3-4B-Instruct, FC)	80.8	73.9	83.0	<u>79.3</u>	<u>87.7</u>	<u>81.9</u>	<u>91.4</u>	44.9	69.8	78.7
• <i>DeepSeek-V3.2 Variants</i>										
DeepSeek-V3.2 (Liu et al., 2025)	68.1	60.1	79.8	63.2	72.2	70.7	79.0	48.3	53.6	66.9
IBA-Agent (DeepSeek-V3.2)	<u>83.5</u>	<u>75.1</u>	<u>84.5</u>	76.9	85.6	78.0	87.9	53.3	72.3	<u>78.8</u>
IBA-Agent (DeepSeek-V3.2, FC)	86.7	77.7	88.9	79.8	89.7	80.2	92.6	<u>54.2</u>	74.1	81.9
• <i>General-Purpose AI Agents</i>										
nanobot (DeepSeek-V3.2) (Ren and HKUDS, 2026)	69.9	54.0	59.3	62.5	56.2	53.0	59.7	38.1	61.1	58.5
Claude Code (DeepSeek-V3.2) ⁴	80.5	64.0	60.9	70.3	63.1	54.7	58.1	33.8	56.7	62.9
Hermes Agent(DeepSeek-V3.2) (Nous Research, 2026)	81.7	63.8	54.2	70.9	59.0	56.9	59.5	35.9	68.6	63.1

Table 4: Performance comparison using the **bge-m3** retriever. Results are evaluated by DeepSeek-V3.2, Qwen3-4B-Instruct, and Qwen3-30B-A3B-Instruct; scores from Qwen3-4B-Instruct are averaged over five runs. Columns correspond to 9 application domains: **Writing** (writing and design), **Work**, **Daily** (daily consumption), **Plan.** (planning), **Health** (exercise and health), **Trans.** (transportation), **Med.** (medical services), **Leis.** (leisure activities), and **Info.** (information management). Best result is highlighted in **green bold text**, second-best result is underlined. Function Call (FC) variants are shaded in light purple.

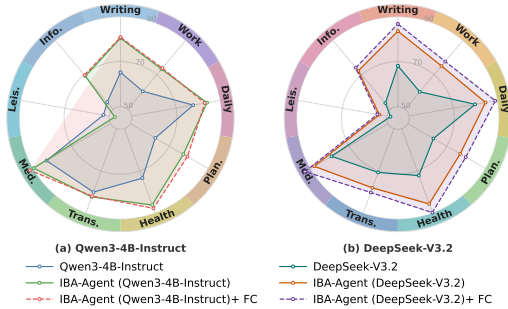


Figure 4: Radar plots of baseline models and IBA-Agent variants across nine domains.

The synthesis step then realizes this plan as the final output, namely personalized content or filled API parameters with an action description.

5 Performance Study

All experiments are conducted in an inference-only setting, with no parameter fine-tuning. We evaluate **ten modern LLMs** spanning the Qwen, DeepSeek, GPT, ChatGLM, and QwQ families, together with **three general-purpose agent systems** (Claude Code ⁴, Hermes Agent (Nous Research, 2026), and nanobot (Ren and HKUDS, 2026)), under a unified RAG pipeline that uses **bge-m3** for dense retrieval. Each output is scored against scenario-specific *behavioral checkpoints* by an LLM judge; for robustness, we aggregate results over **three judges** (DeepSeek-V3.2, Qwen3-4B-Instruct, and Qwen3-30B-A3B-Instruct). Details of baselines,

evaluation metrics, and hyperparameters are provided in Appendix C, with additional experiments on retriever and judge robustness reported in Appendix D.

5.1 Main Results

Table 4 reports results on IBA-BENCH. Standard RAG yields moderate and inconsistent results, with gains not strictly monotonic with model scale, suggesting that stronger backbones alone are insufficient for personalized task completion. In contrast, IBA-Agent consistently improves over its RAG backbones, increasing DeepSeek-V3.2 from 66.9 to 78.8 and Qwen3-4B-Instruct from 67.6 to 78.1. Adding function calling further boosts performance to 81.9 and 78.7 respectively. This indicates that the key challenge lies not only in retrieving relevant memories, but also in converting implicit user preferences into actionable constraints.

Compared to the general-purpose agent systems Claude Code, Hermes Agent, and nanobot, IBA-Agent performs substantially better, demonstrating that generic agent workflows transfer poorly to personalized settings. As shown in Figure 4, IBA-Agent achieves broad gains across most task categories, while function calling enables more flexible module invocation and more reliable personalized

⁴<https://github.com/oboard/claude-code-rev>

Method Variants	Writing	Work	Daily	Plan.	Health	Trans.	Medical	Leisure	Info.	Overall
IBA-Agent (DeepSeek-V3.2)	83.5	75.1	84.5	76.9	85.6	78.0	87.9	53.3	72.3	78.8
w/o Deep Retrieval	79.7 $\downarrow$ 3.8	67.4 $\downarrow$ 7.7	72.9 $\downarrow$ 11.6	73.8 $\downarrow$ 3.1	79.3 $\downarrow$ 6.3	65.1 $\downarrow$ 12.9	64.9 $\downarrow$ 23.0	46.7 $\downarrow$ 6.6	65.9 $\downarrow$ 6.4	70.5 $\downarrow$ 8.3
w/o BT+DA	82.7 $\downarrow$ 0.8	71.9 $\downarrow$ 3.2	83.8 $\downarrow$ 0.7	77.4 $\downarrow$ 0.5	84.4 $\downarrow$ 1.2	74.1 $\downarrow$ 3.9	82.7 $\downarrow$ 5.2	48.4 $\downarrow$ 4.9	70.6 $\downarrow$ 1.7	76.3 $\downarrow$ 2.5
w/o Synth.	76.4 $\downarrow$ 7.1	67.0 $\downarrow$ 8.1	83.1 $\downarrow$ 1.4	70.6 $\downarrow$ 6.3	81.6 $\downarrow$ 4.0	75.8 $\downarrow$ 2.2	86.3 $\downarrow$ 1.6	48.1 $\downarrow$ 5.2	60.6 $\downarrow$ 11.7	73.3 $\downarrow$ 5.5

Table 5: Ablation study of IBA-Agent with DeepSeek-V3.2 as the backbone LLM across task domains, using **bge-m3** as the retriever. Results are evaluated by three judges: DeepSeek-V3.2, Qwen3-4B-Instruct, and Qwen3-30B-A3B-Instruct; Qwen3-4B-Instruct scores are averaged over five runs. *w/o* denotes removing a specific component. **BT+DA** denotes *Broad Thinking + Deep Alignment*, and **Synth.** denotes the original RAG-style synthesis strategy without trajectory-level synthesis.

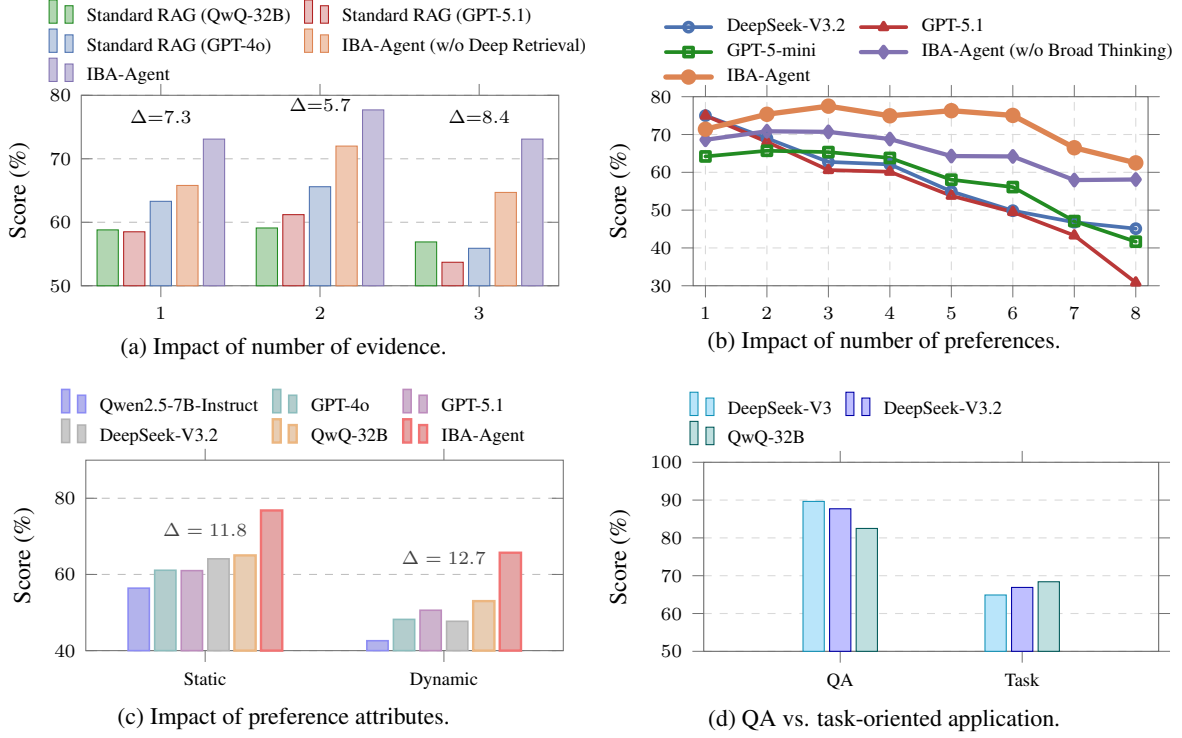


Figure 5: Ablation analyses on evidence quantity, preference complexity, preference attributes, and the knowledge-to-action gap. Δ denotes the performance gap between IBA-Agent and the second-best model.

task completion.

5.2 Ablation Study

Table 5 presents the ablation results of IBA-Agent on a 1,000-instance test set obtained through stratified sampling from IBA-BENCH. The full model consists of Deep Retrieval, Broad Thinking & Deep Alignment, and trajectory-level synthesis. To assess the contribution of each component, we consider three variants: removing Deep Retrieval, removing Broad Thinking & Deep Alignment, and replacing trajectory-level synthesis with the original RAG-style synthesis.

Among these components, Deep Retrieval has the largest impact: removing it reduces overall performance from 78.8 to 70.5 (-8.3), with the largest drops observed in *Medical*, *Transportation*, and *Daily*. This highlights the importance of accurately accessing relevant evidence from long interaction histories. Removing Broad Thinking &

Deep Alignment lowers performance from 78.8 to 76.3 (-2.5). Replacing trajectory-level synthesis with the original RAG-style alternative decreases performance to 73.3 (-5.5), suggesting that effective personalization requires not only retrieving relevant evidence but also integrating it into coherent behavioral trajectories. Notably, these results suggest that the modules are effective as a coordinated mechanism: without reliable Deep Retrieval evidence, downstream Broad Thinking & Deep Alignment may become counterproductive.

5.3 In-depth Analysis

Impact of Number of Evidence. Figure 5a reports preference-conditioned task performance when the number of evidence varies. IBA-Agent consistently outperforms the standard RAG baseline and the variant without Deep Retrieval, highlighting its ability to aggregate preference-relevant signals and the value of broad, deep retrieval. Across all mod-

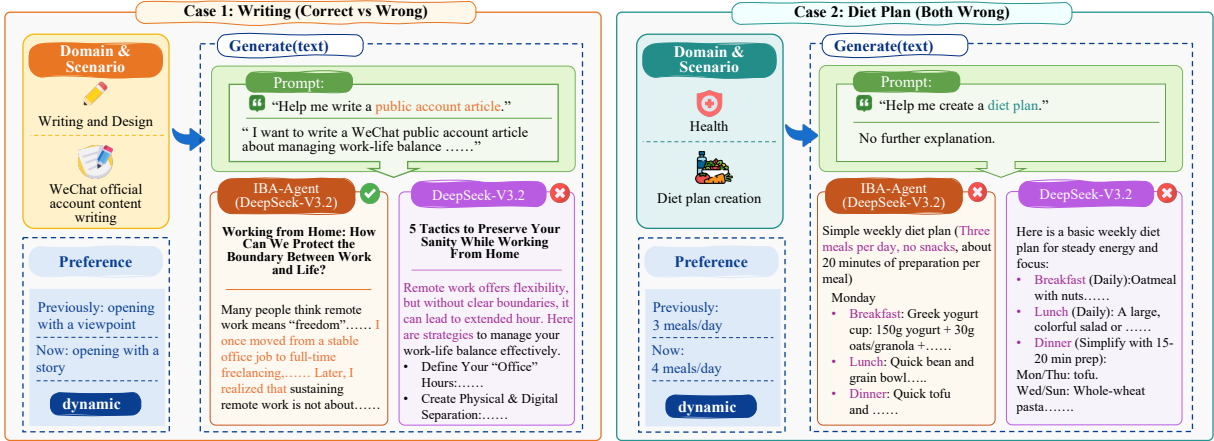


Figure 6: Case studies of personalized Generate(text) execution in IBA-BENCH. Each case compares IBA-AGENT (FUNCTION CALL) instantiated with DeepSeek-V3.2 against the same DeepSeek-V3.2 baseline on the same task and with the same dynamic-preference evidence.

els, increasing evidence from one to two helps, but expanding to three often hurts, suggesting a trade-off between recovering missing preferences and accumulating noise.

Impact of Number of Preferences. Figure 5b shows the performance as the number of preferences (smoothed with a sliding window, $w=2$) increases. Performance drops for all models as constraints accumulate. IBA-Agent degrades more slowly and widens its lead as preference increases. Removing Broad Thinking & Deep Alignment causes a clear drop, confirming its role in robust multi-constraint alignment.

Impact of Preference Attributes. Figure 5c shows the effect of static vs. dynamic preference settings. All models perform better on static preferences, indicating easier behavior alignment. For dynamic preferences, IBA-Agent outperforms the strongest baseline by +12.7 points (vs. +11.8 static), demonstrating robustness to preference shifts.

QA vs. Task-oriented Application. Figure 5d compares QA with direct task execution. For all models, QA consistently exceeds Task, exposing a clear knowledge-to-action gap: models can state preferences correctly but often fail to enforce them, showing that QA competence does not translate into robust preference-aware task completion. A qualitative illustration of this QA-to-execution distinction is provided in Figure 7 in the Appendix.

5.4 Case Study on Knowledge-to-Action Gap

Figure 6 shows two Generate(text) cases under dynamic preferences. Both cases use DeepSeek-V3.2 as the shared backbone and compare direct generation against IBA-AGENT on the same task and with the same preference updates. In the writ-

ing case, IBA-AGENT correctly applies the user’s latest preference to open with a personal story, whereas the baseline follows the outdated preference and opens with a viewpoint. This suggests that IBA-AGENT can better isolate dynamic preference cues and synthesize them into actions. However, in the diet-planning case, both methods fail to update the plan from three meals to four meals per day. This indicates that the task remains difficult when the updated preference requires a global structural revision of the output, rather than a localized stylistic change.

6 Conclusion

In this paper, we identified a critical knowledge-to-action gap in existing personalized agent evaluations, which rely heavily on static user profiles and explicit question answering. We argued that true personalization requires agents to synthesize implicit signals from longitudinal histories rather than merely retrieving explicit tags. To address this, we introduced IBA-BENCH, a comprehensive benchmark that evaluates implicit behavioral alignment under dynamic and noisy conditions. We further proposed IBA-Agent, a framework that bridges information retrieval and task execution through deep retrieval and planning-guided generation. Extensive experiments demonstrate that while current state-of-the-art models struggle to translate historical context into behavioral constraints, our approach significantly improves performance in these complex scenarios. Closing this gap is a prerequisite for personalized agents in the wild.

Limitations

We propose IBA-BENCH as a benchmark for evaluating implicit behavioral alignment in personalized agents; however, it still has two main limitations. First, IBA-BENCH is constructed using LLM-generated synthetic data. Although the synthesis pipeline is carefully designed to inject implicit preferences, temporal dynamics, and noise, hallucinated or inconsistent behaviors may still occur. Second, our evaluation focuses on offline, history-conditioned task execution. As a result, the benchmark cannot capture how agents adapt preferences through online interaction or corrective signals. Extending IBA-BENCH to interactive and tool-augmented settings is left for future work.

Ethical Considerations

Considering potential risks related to privacy and intellectual property, we deliberately avoid using real user logs or information collected from human subjects. All personas, interaction histories, and tasks are fictional, thereby preventing direct exposure of personally identifiable information. At the same time, as LLM-generated content may still reflect biased or undesirable patterns, we address these issues through carefully designed prompts and targeted human review. The benchmark data and prompts will be released under CC BY-NC 4.0, and the evaluation scripts will be released under the MIT License. These released materials are intended for academic research; any redistribution or derivative use should follow the stated licenses and remain within this research and evaluation scope.

References

- Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.
- Saleh Afzoon, Usman Naseem, Amin Beheshti, and Zahra Jamali. 2024. Persobench: Benchmarking personalized response generation in large language models. *CoRR*, abs/2410.03198.
- Team Glm, Aohan Zeng, Bin Xu, Bowen Wang, Chenhui Zhang, Da Yin, Dan Zhang, Diego Rojas, Guanyu Feng, Hanlin Zhao, and 1 others. 2024. Chatglm: A family of large language models from glm-130b to glm-4 all tools. *arXiv preprint arXiv:2406.12793*.
- Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, and 175 others. 2025. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nat.*, 645(8081):633–638.
- Kilem Gwet. 2001. Handbook of inter-rater reliability: How to estimate the level of agreement between two or multiple raters. *Gaithersburg, MD: STATAXIS Publishing Company*.
- Yupu Hao, Pengfei Cao, Zhuoran Jin, Huanxuan Liao, Yubo Chen, Kang Liu, and Jun Zhao. 2025. Evaluating personalized tool-augmented llms from the perspectives of personalization and proactivity. In *Proceedings of ACL*, pages 21897–21935.
- Zhiting Hu and Tianmin Shu. 2023. Language models, agent models, and world models: The law for machine reasoning and planning. *arXiv preprint arXiv:2312.05230*.
- Hideaki Joko, Shubham Chatterjee, Andrew Ramsay, Arjen P. de Vries, Jeff Dalton, and Faegheh Hasibi. 2024. Doing personal LAPS: llm-augmented dialogue construction for personalized multi-session conversational search. In *Proceedings of ACM SIGIR*, pages 796–806.
- Shawn Li, Chenxiao Yu, Han Wang, Wei Yang, Ryan Rossi, Franck Dernoncourt, Xiyang Hu, Philip Yu, Chaowei Xiao, Huan Zhang, and Yue Zhao. 2026. Fortis: Benchmarking over-privilege in agent skills. *Preprint*, arXiv:2605.09163.
- Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, and 1 others. 2024a. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.
- Aixin Liu, Aoxue Mei, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, and 1 others. 2025. Deepseek-v3. 2: Pushing the frontier of open large language models. *arXiv preprint arXiv:2512.02556*.
- Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, Shudan Zhang, Xiang Deng, Aohan Zeng, Zhengxiao Du, Chenhui Zhang, Sheng Shen, Tianjun Zhang, Yu Su, Huan Sun, and 3 others. 2024b. Agentbench: Evaluating llms as agents. In *The Twelfth ICLR*.
- Mahmoud Mohammadi, Yipeng Li, Jane Lo, and Wendy Yip. 2025. Evaluation and benchmarking of LLM agents: A survey. In *Proceedings of ACM SIGKDD*, pages 6129–6139.
- Jisoo Mok, Ik-hwan Kim, Sangkwon Park, and Sungro Yoon. 2025. Exploring the potential of llms as personalized assistants: Dataset, evaluation, and analysis. In *Proceedings of ACL*, pages 10212–10239.

- Nous Research. 2026. Hermes agent: The agent that grows with you. <https://github.com/NousResearch/hermes-agent>. GitHub repository, accessed May 2026.
- OpenAI. 2024. Hello gpt-4o. <https://openai.com/index/hello-gpt-4o/>. Official announcement of GPT-4o model; accessed: 2026-01-04.
- OpenAI. 2025. Gpt-5.1: A smarter, more conversational chatgpt. <https://openai.com/index/gpt-5-1/>. Official announcement of GPT-5.1 model series; accessed: 2026-01-04.
- Joon Sung Park, Joseph C. O’Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. 2023. Generative agents: Interactive simulacra of human behavior. In *Proceedings of UIST*, pages 2:1–2:22.
- Qwen, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiayi Yang, Jingren Zhou, Junyang Lin, and 24 others. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.
- Qwen Team. 2025. Qwq-32b: Embracing the power of reinforcement learning. <https://qwenlm.github.io/blog/qwq-32b/>. Official model introduction blog. Accessed: 2026-01-04.
- Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2023. In-context retrieval-augmented language models. *Transactions of the Association for Computational Linguistics*, 11:1316–1331.
- Xubin Ren and HKUDS. 2026. nanobot: The ultra-lightweight personal AI agent. <https://github.com/HKUDS/nanobot>. GitHub repository, accessed May 2026.
- Vinay Samuel, Henry Peng Zou, Yue Zhou, Shreyas Chaudhari, Ashwin Kalyan, Tanmay Rajpurohit, Ameet Deshpande, Karthik R. Narasimhan, and Vishvak Murahari. 2025. Personagym: Evaluating persona agents and llms. In *Findings of EMNLP*, pages 6999–7022.
- 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. In *NeurIPS*.
- Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. 2020. Learning to summarize with human feedback. *Advances in neural information processing systems*, 33:3008–3021.
- Juntao Tan, Liangwei Yang, Zuxin Liu, Zhiwei Liu, Rithesh R. N., Tulika Manoj Awalganekar, Jianguo Zhang, Weiran Yao, Ming Zhu, Shirley Kokane, Silvio Savarese, Huan Wang, Caiming Xiong, and Shelby Heinecke. 2025. Personabench: Evaluating AI models on understanding personal information through accessing (synthetic) private user data. In *Findings of ACL*, pages 878–893.
- Gemini Team. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *CoRR*, abs/2507.06261.
- Karthik Valmeekam, Matthew Marquez, Sarath Sreedharan, and Subbarao Kambhampati. 2023. On the planning abilities of large language models - A critical investigation. In *NeurIPS*.
- Tiannan Wang, Meiling Tao, Ruoyu Fang, Huilin Wang, Shuai Wang, Yuchen Eleanor Jiang, and Wangchunshu Zhou. 2024. Ai persona: Towards life-long personalization of llms. *arXiv preprint arXiv:2412.13103*.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *NeurIPS*.
- Rebecca Westhäüßer, Wolfgang Minker, and Sebastian Zepf. 2025. Enabling personalized long-term interactions in llm-based agents through persistent memory and user profiles. *arXiv preprint arXiv:2510.07925*.
- An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.
- John Yang, Carlos E. Jimenez, Alexander Wettig, Kilian Lieret, Shunyu Yao, Karthik Narasimhan, and Ofir Press. 2024. Swe-agent: Agent-computer interfaces enable automated software engineering. In *NeurIPS*.
- Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of EMNLP*, pages 2369–2380.
- Weizhi Zhang, Xinyang Zhang, Chenwei Zhang, Liangwei Yang, Jingbo Shang, Zhepei Wei, Henry Peng Zou, Zijie Huang, Zhengyang Wang, Yifan Gao, and 1 others. 2025. Personaagent: When large language model agents meet personalization at test time. *arXiv preprint arXiv:2506.06254*.
- Zheng Zhao, Clara Vania, Subhradeep Kayal, Naila Khan, Shay B. Cohen, and Emine Yilmaz. 2025. Personalens: A benchmark for personalization evaluation in conversational AI assistants. In *Findings of ACL*, pages 18023–18055.

Jiachen Zhu, Menghui Zhu, Renting Rui, Rong Shan, Congmin Zheng, Bo Chen, Yunjia Xi, Jianghao Lin, Weiwen Liu, Ruiming Tang, and 1 others. 2025. Evolutionary perspectives on the evaluation of llm-based ai agents: A comprehensive survey. *arXiv preprint arXiv:2506.11102*.

Appendix

A Benchmark Details

A.1 Dataset Details

Table 13 summarizes the statistics across domains and scenarios, including the number of preference dimensions (**D**), preference values (**P**), task instances (**I**), dynamic-preference support, and action space. This provides a quantitative view of IBA-BENCH’s richness in cross-domain scenario coverage and preference-space diversity. Figure 8 shows the domain and scenario distributions.

Table 14 provides illustrative examples of how preference dimensions are defined in our benchmark across the Writing and Design domain. It is organized in a domain–scenario–dimension–value hierarchy, and lists mutually exclusive preference values for each dimension to support task instantiation and controlled evaluation.

A.2 Task Formulation and Evaluation Details

This section provides the detailed evaluation protocol for the three final-action types in $\mathcal{Y}$. For all types, the evaluator focuses on the final action rather than whether the agent can merely describe or recall user preferences. Table 13 reports the action type of each scenario, including 50 Generate(text) and 16 CallAPI(params) scenarios.

Generate(text). This type is used for text-oriented tasks where the agent completes the request by producing a free-form personalized response, such as a plan, explanation, or written message. Each instance contains a set of hidden personalized requirements for a specific user, such as style preferences, structural preferences, or situational needs. The evaluator assigns a binary score to each requirement according to whether the generated response satisfies it, and the instance score is the average over all requirements. An example is shown in Table 15.

CallAPI(params). This type evaluates whether an agent can translate preferences into executable tool parameters. In scenarios such as ordering food or booking appointments, the agent’s final action is represented as structured API parameters that can be passed to the corresponding tool interface to per-

form the intended operation. Each instance defines an API template with controllable slots, and the hidden personalized requirements specify which slots should reflect the user’s preferences and current situation. The evaluator scores these personalization-relevant slots independently: each slot receives a binary score indicating whether the submitted value satisfies the user’s preferences in the given scenario, and the instance score is the average over all scored slots. Task-fixed or API-validity-only parameters are excluded from the personalization score. An example is shown in Table 16.

A.3 Prompt Details

Figure 9 shows the prompt template used to generate casual-chat dialogues in IBA-BENCH. Figures 10 and 11 show the prompt templates used to synthesize user historical dialogue evidence.

B Benchmark Annotation and Validation

B.1 Human Annotation Protocol

To evaluate the quality of the generated benchmark instances, we randomly sampled 200 instances for manual validation. Each instance was independently annotated by two human annotators along three binary dimensions: *reasonableness*, *coherence*, and *persona consistency*. These dimensions are designed to assess whether an instance is reasonable, internally consistent, and aligned with the intended user role and evolving preferences over time.

Reasonableness. Whether the interaction history and instantiated task are reasonable under common sense and the task setting, without obvious hallucinations or unrealistic assumptions. **1:** plausible and credible. **0:** clearly implausible or severely hallucinatory.

Coherence. Whether the interaction history and task are logically connected and internally self-consistent, without major contradictions or abrupt discontinuities. **1:** coherent and understandable. **0:** conflicting, broken, or inconsistent.

Persona Consistency. Whether the instance is consistent with the target role and appropriately reflects preference changes over time. **1:** consistent with the intended role and preference evolution. **0:** inconsistent with the role or misrepresents preference dynamics.

Annotation procedure. The two human annotators independently labeled all sampled instances

Model	Writing	Work	Daily	Plan.	Health	Trans.	Med.	Leis.	Info.	Overall
• <i>Baseline Models</i>										
Qwen2.5-7B-Instruct (Qwen et al., 2025)	49.3	49.5	77.0	56.7	68.1	68.3	72.9	40.2	33.7	57.8
DeepSeek-V3 (Liu et al., 2024a)	64.1	57.2	77.7	67.0	74.2	71.7	79.3	50.2	51.2	66.5
ChatGLM-4-9b-chat (Glm et al., 2024)	45.6	45.2	70.9	54.3	66.2	60.6	69.0	38.0	33.0	54.4
QwQ-32B (Qwen Team, 2025)	67.2	62.2	81.3	68.9	77.9	79.0	80.3	53.4	56.0	70.2
GPT-4o-mini	54.7	54.1	78.2	61.3	69.0	71.5	71.7	45.2	39.2	61.6
GPT-4o (OpenAI, 2024)	58.8	53.0	76.7	61.3	70.5	70.6	73.1	49.1	39.8	62.0
GPT-5-mini	73.2	66.0	76.4	75.7	81.9	74.1	81.9	52.4	68.0	73.1
GPT-5.1 (OpenAI, 2025)	75.8	63.2	80.2	70.2	75.5	74.0	78.8	58.0	59.1	71.1
• <i>DeepSeek-V3.2 Variants</i>										
DeepSeek-V3.2 (Liu et al., 2025)	69.4	61.3	80.3	67.6	73.7	75.4	79.6	54.1	55.8	69.1
IBA-Agent (DeepSeek-V3.2)	86.3	<u>78.3</u>	<u>85.3</u>	82.1	86.5	80.7	87.3	60.3	79.9	81.9
IBA-Agent (DeepSeek-V3.2, FC)	88.3	79.6	89.2	<u>82.9</u>	90.4	84.0	92.2	<u>59.8</u>	<u>78.7</u>	84.0
• <i>Qwen3-4B-Instruct Variants</i>										
Qwen3-4B-Instruct (Yang et al., 2025)	63.6	59.5	80.1	66.1	74.7	76.0	81.5	48.9	53.9	67.8
IBA-Agent (Qwen3-4B-Instruct)	82.4	75.1	84.4	81.7	87.5	<u>82.5</u>	88.8	49.9	73.6	80.0
IBA-Agent (Qwen3-4B-Instruct, FC)	81.1	76.0	83.1	83.5	<u>88.5</u>	<u>82.5</u>	<u>90.0</u>	50.0	72.0	80.2
• <i>General-Purpose AI Agents</i>										
nanobot (DeepSeek-V3.2)	69.7	52.7	60.4	67.4	61.1	59.4	60.4	44.8	67.1	61.2
Claude Code (DeepSeek-V3.2)	75.1	59.5	60.3	69.5	62.3	59.1	57.1	38.7	57.3	61.8
Hermes Agent (DeepSeek-V3.2)	79.1	60.2	61.3	71.7	62.8	59.3	60.3	41.7	69.6	64.5

Table 6: Performance comparison using the **bge-m3** retriever and **Qwen3-30B-A3B-Instruct** as the judge model. The columns correspond to nine application domains: **Writing** (writing and design), **Work**, **Daily** (daily consumption), **Plan.** (planning), **Health** (exercise and health), **Trans.** (transportation), **Med.** (medical services), **Leis.** (leisure activities), and **Info.** (information management). The best result in each column is highlighted as **green bold text**, the second-best result is underlined, Function Call variants are shaded in light purple. FC stands for Function Call.

Model	Writing	Work	Daily	Plan.	Health	Trans.	Med.	Leis.	Info.	Overall
• <i>Baseline Models</i>										
Qwen2.5-7B-Instruct (Qwen et al., 2025)	42.7	42.0	75.0	45.8	57.9	60.7	73.9	32.8	32.4	51.0
DeepSeek-V3 (Liu et al., 2024a)	48.2	43.2	74.7	50.3	61.2	62.2	75.6	34.6	40.1	55.0
ChatGLM-4-9b-chat (Glm et al., 2024)	42.1	39.0	67.2	43.8	56.5	54.7	67.1	29.8	31.8	48.5
QwQ-32B (Qwen Team, 2025)	54.5	50.0	77.2	54.1	67.1	69.3	76.3	39.3	43.1	59.6
GPT-4o-mini	47.1	44.2	74.2	47.1	59.7	64.1	69.4	30.7	34.2	53.2
GPT-4o (OpenAI, 2024)	55.1	42.0	72.0	45.2	60.8	62.3	69.9	35.9	33.9	53.3
GPT-5-mini	57.3	52.4	73.4	58.6	71.1	66.1	76.6	42.5	48.1	61.3
GPT-5.1 (OpenAI, 2025)	58.4	49.0	76.2	54.2	66.5	65.7	75.8	<u>49.0</u>	43.4	59.8
• <i>DeepSeek-V3.2 Variants</i>										
DeepSeek-V3.2 (Liu et al., 2025)	54.6	47.2	76.9	52.5	64.4	64.7	77.0	39.3	40.5	58.0
IBA-Agent (DeepSeek-V3.2)	72.2	64.8	83.5	<u>69.3</u>	<u>82.5</u>	73.4	87.1	42.5	57.0	71.8
IBA-Agent (DeepSeek-V3.2, FC)	78.4	69.3	87.8	75.3	88.1	76.3	92.4	43.9	62.6	76.7
• <i>Qwen3-4B-Instruct Variants</i>										
Qwen3-4B-Instruct (Yang et al., 2025)	52.3	49.4	71.0	51.9	65.5	88.0	82.9	59.5	43.7	60.6
IBA-Agent (Qwen3-4B-Instruct)	71.9	65.2	82.4	71.0	82.9	<u>80.2</u>	89.6	39.6	<u>58.0</u>	72.6
IBA-Agent (Qwen3-4B-Instruct, FC)	72.6	65.1	80.6	72.4	84.1	79.9	<u>91.1</u>	36.3	60.6	73.0
• <i>General-Purpose AI Agents</i>										
nanobot (DeepSeek-V3.2)	58.4	44.8	54.3	51.9	45.6	45.5	56.9	28.4	45.8	49.3
Claude Code (DeepSeek-V3.2)	81.5	65.4	56.9	67.6	58.4	50.5	56.4	26.8	56.5	60.9
Hermes Agent (DeepSeek-V3.2)	82.1	63.4	35.2	67.8	48.6	54.6	55.5	28.2	67.1	57.5

Table 7: Performance comparison using the **bge-m3** retriever and **DeepSeek-V3.2** as the judge model. The columns correspond to nine application domains: **Writing** (writing and design), **Work**, **Daily** (daily consumption), **Plan.** (planning), **Health** (exercise and health), **Trans.** (transportation), **Med.** (medical services), **Leis.** (leisure activities), and **Info.** (information management). The best result in each column is highlighted as **green bold text**, the second-best result is underlined, Function Call variants are shaded in light purple. FC stands for Function Call.

Model	Writing	Work	Daily	Plan.	Health	Trans.	Med.	Leis.	Info.	Overall
• <i>Baseline Models</i>										
Qwen2.5-7B-Instruct (Qwen et al., 2025)	66.0	61.5	79.8	64.0	75.2	66.5	78.2	47.3	45.3	66.0
DeepSeek-V3 (Liu et al., 2024a)	83.1	69.3	78.8	70.8	78.8	68.1	81.4	49.2	64.4	73.1
ChatGLM-4-9b-chat (Glm et al., 2024)	66.6	57.6	74.0	62.3	73.4	60.0	71.7	42.3	46.5	63.1
QwQ-32B (Qwen Team, 2025)	84.4	72.6	83.4	70.4	80.6	74.4	82.1	49.3	67.0	75.3
GPT-4o-mini	71.3	66.4	79.9	65.2	75.6	69.2	74.3	43.7	53.7	68.4
GPT-4o (OpenAI, 2024)	76.5	67.4	78.0	67.2	77.3	67.0	75.9	47.4	56.7	69.7
GPT-5-mini	81.02	73.34	79.20	75.16	82.12	72.32	82.26	54.78	73.58	76.10
GPT-5.1 (OpenAI, 2025)	84.6	75.0	83.2	72.5	79.7	72.2	79.5	63.3	71.9	76.7
• <i>DeepSeek-V3.2 Variants</i>										
DeepSeek-V3.2 (Liu et al., 2025)	80.3	71.8	82.3	69.5	78.5	71.9	80.3	51.6	64.6	73.7
IBA-Agent (DeepSeek-V3.2)	<u>92.1</u>	<u>82.3</u>	84.8	79.4	87.7	79.9	89.2	57.1	<u>80.0</u>	82.8
IBA-Agent (DeepSeek-V3.2, FC)	93.3	84.1	89.7	<u>81.3</u>	90.6	80.4	93.3	<u>59.0</u>	81.1	85.1
• <i>Qwen3-4B-Instruct Variants</i>										
Qwen3-4B-Instruct (Yang et al., 2025)	79.7	72.3	82.4	70.5	80.1	76.1	84.4	49.5	64.9	74.6
IBA-Agent (Qwen3-4B-Instruct)	86.8	79.6	85.2	79.0	88.0	84.1	90.1	53.0	76.1	81.6
IBA-Agent (Qwen3-4B-Instruct, FC)	<u>88.8</u>	<u>80.7</u>	<u>85.3</u>	82.2	<u>90.5</u>	<u>83.4</u>	<u>93.2</u>	48.4	<u>76.9</u>	82.9
• <i>General-Purpose AI Agents</i>										
nanobot (DeepSeek-V3.2)	81.5	64.6	63.2	68.3	62.0	54.2	61.8	41.0	70.5	64.9
Claude Code (DeepSeek-V3.2)	84.8	67.2	65.6	73.8	68.6	54.5	60.7	35.9	56.2	66.1
Hermes Agent (DeepSeek-V3.2)	83.8	67.8	66.1	73.3	65.6	56.7	62.8	37.8	69.0	67.4

Table 8: Performance comparison using the **bge-m3** retriever and **Qwen3-4B-Instruct** as the judge model. Reported results are averaged over five runs. The columns correspond to nine application domains: **Writing** (writing and design), **Work**, **Daily** (daily consumption), **Plan.** (planning), **Health** (exercise and health), **Trans.** (transportation), **Med.** (medical services), **Leis.** (leisure activities), and **Info.** (information management). The best result in each column is highlighted as **green bold text**, the second-best result is underlined, **Function Call variants** are shaded in light purple. FC stands for Function Call.

using the criteria described above. In addition, we obtained two independent AI-based annotations from GPT-5.4 using the same evaluation dimensions and instructions. To assess the reliability of the validation process, we compared (1) the agreement between the two human annotators and (2) the agreement between the two AI-based annotations. Human annotations were treated as the primary reference, while AI judgments served as an additional external comparison signal.

Table 11 reports two complementary measures of annotation reliability: raw agreement and Gwet’s AC_1 (Gwet, 2001). Raw agreement is the proportion of instances receiving the same label from two annotators:

$$P_o = \frac{1}{N} \sum_{i=1}^N \mathbf{1}(y_i^{(1)} = y_i^{(2)}),$$

where N is the number of annotated instances, $y_i^{(1)}$ and $y_i^{(2)}$ denote the labels assigned by two raters to instance i , and $\mathbf{1}(\cdot)$ is the indicator function. Because the labels are imbalanced, we additionally report Gwet’s AC_1 , which is more robust than Co-

hen’s kappa under skewed label distributions:

$$AC_1 = \frac{P_o - P_e}{1 - P_e}$$

where P_o is the observed agreement and P_e is the chance agreement estimated under Gwet’s formulation.

C Experiment Settings

C.1 Baselines

We evaluate personalized task completion across a diverse set of LLMs, including Qwen2.5-7B-Instruct (Qwen et al., 2025), ChatGLM-4-9B-Chat (Glm et al., 2024), DeepSeek-V3 (Liu et al., 2024a), DeepSeek-V3.2 (Liu et al., 2025), QwQ-32B (Qwen Team, 2025), GPT-4o-mini, GPT-4o (OpenAI, 2024), GPT-5-mini, GPT-5.1 (OpenAI, 2025), and Qwen3-4B-Instruct (Yang et al., 2025), under a retrieval-augmented generation (RAG) baseline. All models are accessed via official or widely used inference platforms (DeepSeek API, SiliconFlow, and OpenAI API). The RAG baseline uses bge-m3¹ to retrieve sentence-level chunks concatenated with the task query as LLM input. Beyond standalone LLMs, we also evaluate representative agent-based systems, namely Claude

Model	Writing	Work	Daily	Plan.	Health	Trans.	Med.	Leis.	Info.	Overall
Qwen2.5-7B-Instruct	66.0±0.57	61.5±0.30	79.8±0.47	64.0±0.29	75.2±0.15	66.5±0.34	78.2±0.14	47.3±0.62	45.3±0.53	66.0±0.22
DeepSeek-V3	83.1±9.48	69.3±0.42	78.8±0.23	70.8±0.28	78.8±0.38	68.1±0.17	81.4±0.21	49.2±0.42	64.4±0.11	73.1±1.28
ChatGLM-4-9b-chat	66.6±0.67	57.6±0.37	74.0±0.55	62.3±0.18	73.4±0.48	60.0±0.16	71.7±0.31	42.3±0.83	46.5±0.83	63.1±0.19
QwQ-32B	84.4±8.74	72.6±0.40	83.4±0.45	70.4±0.18	80.6±0.16	74.4±0.36	82.1±0.15	49.3±0.25	67.0±0.19	75.3±1.06
GPT-4o-mini	71.3±0.65	66.4±0.30	79.9±0.43	65.2±0.30	75.6±0.97	69.2±0.24	74.3±0.18	43.7±1.08	53.7±0.48	68.4±0.29
GPT-4o	76.5±0.61	67.4±0.30	78.0±0.32	67.2±0.18	77.3±0.52	67.0±0.04	75.9±0.26	47.4±0.23	56.7±0.23	69.7±0.09
GPT-5-mini	81.02±9.76	73.34±7.76	79.20±0.66	75.16±5.07	82.12±4.15	72.32±2.09	82.26±0.50	54.78±3.34	73.58±11.01	76.10±4.64
GPT-5.1	84.6±0.61	75.0±0.07	83.2±0.30	72.5±0.54	79.7±0.15	72.2±0.35	79.5±0.07	63.3±0.67	71.9±0.90	76.7±0.11
DeepSeek-V3.2 Variants										
DeepSeek-V3.2	80.3±0.68	71.8±0.33	82.3±0.56	69.5±0.19	78.5±0.61	71.9±0.26	80.3±0.35	51.6±1.04	64.6±0.52	73.7±0.24
IBA-Agent (DeepSeek-V3.2)	92.1±0.88	82.3±0.34	84.8±3.20	79.4±0.13	87.7±0.28	79.9±5.48	89.2±0.70	57.1±6.47	80.0±0.63	82.8±0.31
IBA-Agent (DeepSeek-V3.2, FC)	93.3±0.56	84.1±0.31	89.7±0.38	81.3±0.38	90.6±0.23	80.4±0.38	93.3±0.13	59.0±0.33	81.1±0.96	85.1±0.23
Qwen3-4B-Instruct Variants										
Qwen3-4B-Instruct	79.7±0.38	72.3±0.36	82.4±0.41	70.5±0.22	80.1±0.28	76.1±0.55	84.4±0.81	49.5±0.61	64.9±0.47	74.6±0.15
IBA-Agent (Qwen3-4B-Instruct)	86.8±0.30	79.6±0.43	85.2±4.75	79.0±0.27	88.0±0.13	84.1±3.14	90.1±0.18	53.0±7.76	76.1±0.57	81.6±0.24
IBA-Agent (Qwen3-4B-Instruct, FC)	88.82±0.78	80.74±0.35	85.34±0.43	82.22±0.30	90.54±0.26	83.46±0.23	93.20±0.16	48.46±0.78	76.96±0.69	82.90±0.29
General-Purpose AI Agents										
nanobot	81.5±0.4	64.6±0.7	63.2±1.1	68.3±0.3	62.0±0.2	54.2±0.5	61.8±0.2	41.0±0.6	70.5±1.1	64.9±0.2
Hermes Agent	83.8±0.3	67.8±0.5	66.1±0.2	73.3±0.4	65.6±0.5	56.7±0.2	62.8±0.6	37.8±0.7	69.0±0.3	67.4±0.1
Claude Code	84.8±4.7	67.2±0.5	65.6±0.6	73.8±0.3	68.6±0.3	54.5±0.4	60.7±0.2	35.9±0.6	56.2±0.5	66.1±0.5

Table 9: Performance comparison using the **bge-m3** retriever and **Qwen3-4B-Instruct** as the judge model. Reported results are averaged over five runs, with sample standard deviations shown in a compact $\pm$ format. The columns correspond to nine application domains: **Writing** (writing and design), **Work**, **Daily** (daily consumption), **Plan.** (planning), **Health** (exercise and health), **Trans.** (transportation), **Med.** (medical services), **Leis.** (leisure activities), and **Info.** (information management). **Green** highlights the **Overall** column, **yellow** marks cells with moderate variation ($\text{std} \geq 2.0$ and < 5.0), and **red** marks cells with high variation ($\text{std} \geq 5.0$). FC stands for Function Call.

Model	Writing	Work	Daily	Plan.	Health	Trans.	Medical	Leisure	Info.	Overall
Qwen2.5-7B-Instruct	46.9	45.6	76.6	53.2	66.6	85.7	79.3	51.3	37.1	59.2
DeepSeek-V3	54.2	48.3	75.9	66.1	72.5	89.7	87.3	53.4	43.1	64.7
GPT-4o-mini	49.9	45.9	79.0	54.9	70.0	92.8	90.5	58.8	37.6	62.4
GPT-4o	49.8	45.4	77.4	54.3	66.6	91.8	87.8	52.0	35.8	60.8

Table 10: Performance comparison using **all-mpnet-base-v2** retriever. The columns correspond to nine application domains: **Writing** (writing and design), **Work**, **Daily** (daily consumption), **Plan.** (planning), **Health** (exercise and health), **Trans.** (transportation), **Med.** (medical services), **Leis.** (leisure activities), and **Info.** (information management).

Dimension	Agreement	AC_1
Human–Human		
Reasonableness	73.2%	0.61
Coherence	82.5%	0.79
Persona Consistency	85.1%	0.72
Overall	80.2%	0.70
AI–AI (GPT-5.4)		
Reasonableness	94.5%	0.94
Coherence	94.5%	0.94
Persona Consistency	93.0%	0.92
Overall	94.00%	0.93

Table 11: Human–human and AI–AI agreement on benchmark quality annotations over 200 matched samples.

Code, Hermes Agent, and nanobot (a lightweight personal AI agent inspired by OpenClaw), to assess their end-to-end task-completion performance under the same personalized setting.

Signal category	Score contribution
Explicit revisions	3–4
User complaint or strong affect	3–4
Direct preference statement	2
Implicit behavioral patterns	1
Retention threshold	≥ 5

Table 12: Scoring criteria used by the salience re-ranking tool for preference-aware passage re-ranking. Implicit behavioral patterns are indirect cues, such as repeated behaviors.

C.2 Evaluation Metric

For each scenario, we define a small set of behavioral checkpoints, each corresponding to an atomic seed preference of the user in that scenario. For generation-based tasks, given an output y_i for instance i , we use three LLM-based judges, namely DeepSeek-V3.2, Qwen3-4B-Instruct, and Qwen3-30B-A3B-Instruct, to assess whether y_i satisfies each checkpoint, yielding a binary score

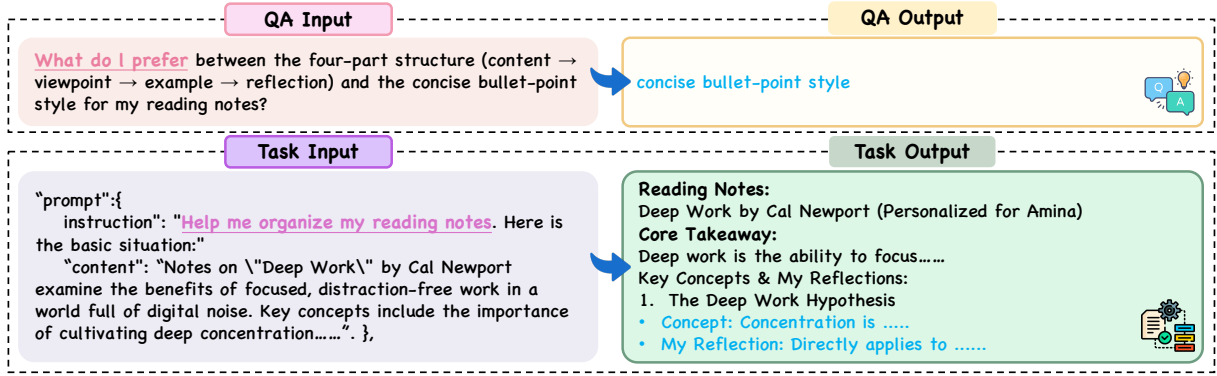


Figure 7: QA (using HiCUPID (Mok et al., 2025) as an example) vs. IBA-Agent for preference modeling. All comparisons are conducted using DeepSeek-V3.2 to control for model capability.

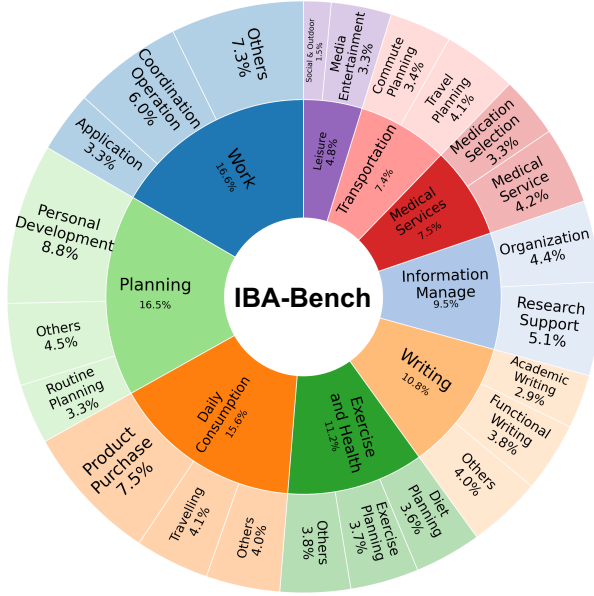


Figure 8: Hierarchical distribution of domains and scenarios in the benchmark.

$f_{i,j}^k(y_i) \in \{0, 1\}$. The instance-level score is computed as the average satisfaction across checkpoints. For API-actionable tasks, each required parameter is evaluated as a checkpoint, receiving 1 if it is valid and consistent with the task requirement and inferred user preference, and 0 otherwise. The instance-level score is the average correctness across required parameters.

C.3 Parameters

We evaluate all models using three judges: DeepSeek-V3.2, Qwen3-4B-Instruct, and Qwen3-30B-A3B-Instruct, with the maximum output length set to 3,000 tokens. For Qwen3-4B-Instruct, we report the average over five runs to assess evaluation stability, with results shown in Table 9 in Appendix A. For retrieval, we use bge-m3 with 256-token chunks and a 50-token overlap. Standard retrieval returns the top-3 passages per query, while Broad Retrieval issues up to five expanded

queries and retains at most ten passages after filtering and optional re-ranking. We fix the random seed to 42 for reproducibility.

D Supplementary Experiments

Table 10 reports supplementary benchmark results using the all-mpnet-base-v2 retriever, evaluating Qwen2.5-7B-Instruct, DeepSeek-V3, GPT-4o-mini, and GPT-4o. Compared with the bge-m3 setting in Table 4, all-mpnet-base-v2 improves the overall score for all four models (+6.1 for Qwen2.5-7B, +7.6 for DeepSeek-V3, +4.4 for GPT-4o-mini, and +3.1 for GPT-4o), with the largest gains concentrated in the Planning and Health domains. Tables 6, 7, and 8 present evaluation results on the same set of execution outputs using three different judge models, namely Qwen3-30B-A3B-Instruct, DeepSeek-V3.2, and Qwen3-4B-Instruct. Among them, Table 9 further reports the per-cell standard deviation across five repeated evaluations under Qwen3-4B-Instruct as the judge. The results show that overall scores remain highly stable for most models and IBA-Agent variants, with overall standard deviations generally below 0.31, suggesting that the main performance trends are highly reproducible. Across three judge models, IBA-Agent consistently outperforms all baseline models, and the function-calling variant further improves overall performance in most settings, demonstrating the robustness of structured history modeling and modular agent execution.

Domain	Scenario	D	P	I	Dyn.	Action Space
Writing and Design	Peer review reports	4	9	54	✓	Generate(text)
	Academic paper writing	6	15	71	✓	Generate(text)
	Research proposal (RP) writing	4	13	76	✓	Generate(text)
	Business plan writing	3	8	29	✓	Generate(text)
	Diary writing	3	7	23	✓	Generate(text)
	Technical documentation writing	4	12	80	✓	Generate(text)
	Blog post writing	2	4	4	✗	Generate(text)
	WeChat public account article	3	8	27	✓	Generate(text)
	Novel writing	2	4	6	✗	Generate(text)
	Resume writing	5	14	72	✓	Generate(text)
	Short video script design	3	11	41	✗	Generate(text)
	PPT slide content writing	2	9	15	✓	Generate(text)
	Mind map design	4	20	79	✓	Generate(text)
	Modular course writing (teaching aid)	4	14	71	✓	Generate(text)
Work	Instruction manual writing	3	15	73	✓	Generate(text)
	Self-introduction script writing	3	8	29	✗	Generate(text)
	Scholarship application	7	17	140	✓	Generate(text)
	Performance report spreadsheet	6	23	90	✓	Generate(text)
	Schedule planning	6	31	119	✓	Generate(text)
	Data analysis	4	27	101	✓	Generate(text)
	Programming	5	23	90	✓	Generate(text)
	Leave application	4	10	93	✗	Generate(text)
	Project management	5	22	105	✓	Generate(text)
	Communication management	2	5	7	✗	Generate(text)
	Announcement drafting	3	12	74	✓	Generate(text)
	Event organization	4	14	111	✗	Generate(text)
	Work summary writing	4	16	113	✓	Generate(text)
	Terminology explanation	3	14	112	✓	Generate(text)
Daily Consumption	Food ordering	3	15	122	✓	CallAPI(params)
	Ticket booking	4	17	145	✓	CallAPI(params)
	Clothing purchase	4	15	153	✓	CallAPI(params)
	Digital product purchase	4	10	89	✗	CallAPI(params)
	Accommodation booking	4	12	138	✗	CallAPI(params)
	Online course purchase	5	12	154	✗	CallAPI(params)
	Book purchase	4	12	127	✓	CallAPI(params)
	Renting a house	11	28	158	✗	CallAPI(params)
Planning	Weekly plan	4	19	122	✓	Generate(text)
	Monthly plan	4	12	107	✓	Generate(text)
	Long-term goal setting	3	16	130	✓	Generate(text)
	Project plan	4	15	114	✗	Generate(text)
	Financial plan	3	13	74	✓	Generate(text)
	Travel plan	5	18	123	✓	Generate(text)
	Study plan	4	19	154	✓	Generate(text)
	Career development plan	4	16	100	✓	Generate(text)
	Habit-building plan	4	18	115	✗	Generate(text)
Exercise and Health	Skill improvement plan	4	16	112	✓	Generate(text)
	Exercise plan	5	22	260	✓	Generate(text)
	Diet plan	4	21	255	✓	Generate(text)
	Sleep plan	6	32	262	✓	Generate(text)
Transportation	Mental health support	1	6	4	✗	Generate(text)
	Commute route planning	5	21	236	✗	CallAPI(params)
	Taxi / ride-hailing travel	2	6	8	✗	CallAPI(params)
	Travel and sightseeing route planning	4	19	215	✗	CallAPI(params)
	Parking	3	10	59	✗	CallAPI(params)
Medical Services	Appointment registration	4	15	210	✗	CallAPI(params)
	Medicine selection	4	17	231	✗	CallAPI(params)
	Online medical consultation	3	14	83	✗	Generate(text)
Leisure Activities	Audio-visual entertainment	4	24	229	✓	CallAPI(params)
	Outdoor leisure	2	5	6	✗	CallAPI(params)
	Social assistance	3	14	98	✓	Generate(text)
Information Management	Research report writing	4	18	174	✓	Generate(text)
	Timeline organization	3	10	41	✓	Generate(text)
	Meeting minutes organization	3	13	60	✓	Generate(text)
	Reading notes	2	9	16	✓	Generate(text)
	Folder organization	5	19	191	✗	Generate(text)
	Tool usage	9	34	182	✓	Generate(text)

Table 13: Scenario-level statistics across benchmark domains. **D**: number of preference dimensions; **P**: number of preference values defined for the scenario; **I**: number of task instances; **Dyn.**: whether dynamic preferences are supported.

Domain	Scenario	Preference Dimension	Preference Value
Writing and Design	Journal writing	Time granularity	Records by day (one entry per whole day). Records by key moments/timestamps.
		Content structure	Prefers a fixed template: “What happened / How I felt”. Prefers free-form writing (write whatever comes to mind). Preferred the fixed template; now prefers free-form writing.
		Length preference	Prefers short entries (about 50–150 words each). Prefers long entries (detailed expansion; 300+ words).
Writing and Design	Business plan writing	Overall narrative style	Prefers narrative-driven writing. Prefers logic-driven writing (clear structure, data-centric). Preferred logic-driven writing; now narrative-driven writing. Preferred narrative-driven writing; now logic-driven writing.
		Market analysis depth	Prefers extensive use of industry data. Prefers highlighting key data points with concise interpretation.
		Risk section writing	Emphasizes explicit identification of risks. Follows an optimistic narrative, downplays risks.
Writing and Design	Technical doc writing	Document structure	Task-oriented structure organized around “how-to” steps. Concept-oriented structure.
		Technical depth	Prefers detailed explanations. Prefers simplified explanations.
		Example code style	Prefers complete, runnable code examples. Prefers code snippets (key parts only). Prefers pseudocode. Previously preferred complete; now prefers pseudocode. Previously preferred pseudocode; now prefers code snippets. Previously preferred code snippets; now prefers pseudocode.
		API explanation	Prefers table-style specification. Prefers narrative explanation of API behavior.
Writing and Design	Review writing	Overall structure	Prefers organizing reviews into Summary / Pros / Cons sections. Prefers unstructured summaries with weaknesses separated.
		Weakness format	Prefers bullet-pointed weaknesses (with numbering). Prefers bullet-pointed weaknesses (without numbering).
		Weakness count	Requires exactly three weaknesses. Requires exactly five weaknesses.
		Weakness content	Prefers keywords + full sentences. Prefers direct full-sentence descriptions. Preferred direct sentences; now prefers keywords + sentences.
Writing and Design	Blog article writing	Narrative perspective	First-person personal perspective. Third-person objective perspective.
		Citations / sources	Prefers academic style: formal citations, links, references. Prefers original style.
Writing and Design	Public account writing	Opening style	Storytelling-based opening. Viewpoint-based opening. Preferred storytelling-based opening, now viewpoint-based. Preferred viewpoint-based opening, now storytelling-based.
		Social expression	Leans toward social topics. Leans toward personal perspectives.
		Title style	Attention-grabbing titles. Formal titles (standard phrasing, fewer stylistic markers).
Writing and Design	Novel writing	Narrative perspective	First-person narration. Third-person narration.
		Narrative type	Linear storytelling in chronological order. Non-linear storytelling (e.g., flashbacks, parallel storylines).

Table 14: Examples of preference dimensions and values in the Writing and Design domain.

EXAMPLE 1: Generate(text)																	
Output type	Scenario	Current query															
Generate(text)	Diet planning	Help me create a diet plan.															
Hidden evaluation instance. The generated plan should satisfy three personalized requirements: (1) muscle gain with about +300 kcal/day; (2) low fat, with fat intake $\leq$ 20% of total calories; (3) breakfast is mandatory.																	
Agent output (summary version). I will make breakfast a central, non-negotiable part of the plan so it supports your muscle gain and keeps your day structured. Eat about +300 kcal above maintenance overall, with fat kept at or below 20% of total calories. Use a fixed daily breakfast within 60–90 minutes of waking, targeting about 400–550 kcal and 25–30 g protein. Breakfast options include protein oatmeal, an egg-and-toast plate, or a low-fat smoothie. Lunch and dinner use lean protein, rice/quinoa/potatoes, vegetables, and measured oil; dinner may include a small portion of avocado or nuts.																	
Behavioral score. <table> <tr> <th>Criterion</th><th>Score</th><th>Evaluator rationale</th></tr> <tr> <td>Muscle gain with +300 kcal/day</td><td>1</td><td>The plan provides a +300 kcal surplus and supports muscle gain.</td></tr> <tr> <td>Low fat, $\leq$ 20% of calories</td><td>0</td><td>Fat sources such as seeds, avocado, nuts, and oil may exceed the fat cap.</td></tr> <tr> <td>Breakfast is mandatory</td><td>1</td><td>Breakfast is explicitly included as a detailed daily meal.</td></tr> <tr> <td>Total / Average</td><td>2/3</td><td>Average score: 0.67</td></tr> </table>			Criterion	Score	Evaluator rationale	Muscle gain with +300 kcal/day	1	The plan provides a +300 kcal surplus and supports muscle gain.	Low fat, $\leq$ 20% of calories	0	Fat sources such as seeds, avocado, nuts, and oil may exceed the fat cap.	Breakfast is mandatory	1	Breakfast is explicitly included as a detailed daily meal.	Total / Average	2/3	Average score: 0.67
Criterion	Score	Evaluator rationale															
Muscle gain with +300 kcal/day	1	The plan provides a +300 kcal surplus and supports muscle gain.															
Low fat, $\leq$ 20% of calories	0	Fat sources such as seeds, avocado, nuts, and oil may exceed the fat cap.															
Breakfast is mandatory	1	Breakfast is explicitly included as a detailed daily meal.															
Total / Average	2/3	Average score: 0.67															

Table 15: Example of a Generate(text) task in IBA-BENCH.

EXAMPLE 2: CallAPI(params)																	
Output type	Scenario	Current query															
CallAPI(params)	Food delivery	Help me order a food delivery.															
Hidden evaluation instance. The API action should satisfy three personalized requirements: (1) the user used to like spicy delivery food, but has recently caught a cold and cannot eat spicy food; (2) the user prefers Western cuisine; (3) the preferred price range is 20–30 yuan.																	
Agent output (summary version). Filled API template https://meituan.com/waimai/search? cuisine_type=Western& meal_type=lunch& price_range=under 30& flavor_preference=mild& dietary_restrictions=non-spicy, no seafood& restaurant_name=none& quantity=1& time_specific_preference=none Reasoning: The agent overrides the user’s previous spicy-food preference because the user recently caught a cold. It therefore sets mild and non-spicy. It also preserves the user’s Western cuisine preference and budget by setting under 30. Long-term preference / budget Dynamic health-related update Other parameters																	
Behavioral score. <table> <tr> <th>Criterion</th><th>Score</th><th>Evaluator rationale</th></tr> <tr> <td>Avoid spicy food due to recent cold</td><td>1</td><td>Uses mild and non-spicy.</td></tr> <tr> <td>Prefer Western cuisine</td><td>1</td><td>Sets cuisine_type=Western.</td></tr> <tr> <td>Price range 20–30 yuan</td><td>1</td><td>Sets price_range=under 30.</td></tr> <tr> <td>Total / Average</td><td>3/3</td><td>Average score: 1.00</td></tr> </table>			Criterion	Score	Evaluator rationale	Avoid spicy food due to recent cold	1	Uses mild and non-spicy.	Prefer Western cuisine	1	Sets cuisine_type=Western.	Price range 20–30 yuan	1	Sets price_range=under 30.	Total / Average	3/3	Average score: 1.00
Criterion	Score	Evaluator rationale															
Avoid spicy food due to recent cold	1	Uses mild and non-spicy.															
Prefer Western cuisine	1	Sets cuisine_type=Western.															
Price range 20–30 yuan	1	Sets price_range=under 30.															
Total / Average	3/3	Average score: 1.00															

Table 16: Example of a CallAPI(params) task in IBA-BENCH.

SYSTEM PROMPT FOR CASUAL-CHAT GENERATION

Role Setting. You are an English casual-chat dialogue generation assistant. The current user's name is "{user_name}," and their general mood/state is "{mood}." Their detailed persona is as follows:

{persona_summary}

Task. Your task is to generate natural, realistic daily casual-chat content **based solely on the personal information in the above user persona** (such as age, major, school, living environment, interests, emotions, etc.).

Important requirements:

1. The conversation should only be about light everyday topics, such as study rhythm, life status, social life, hobbies, emotional feelings, city/campus experiences, etc.
2. **Absolutely no specific tasks or instructions**, especially anything like "help me review a paper," "write reviewer comments," "this paper/manuscript/research," "task/instruction/system prompt," etc.
3. Even if the user persona includes research direction, paper topics, or project background, these can only serve as background atmosphere – do not expand on research/paper content or switch into a "task-helping" mode.
4. The conversation may include only two speaker prefixes: "User:" and "ChatGPT:". No other notes or markers.
5. The overall tone should be natural and unforced – more like casual chatting between friends than executing a task.

Figure 9: System prompt used for casual-chat generation.

USER PROMPT FOR EVIDENCE DATA GENERATION

INPUTS

User Information.

{person_info}

User Preferences (Hidden – for the scriptwriter only).

{instance}

Task Overview.

{instruction_text}

Domain / Category.

{domain}

Sub-scenario.

{sub_scenario}

Task Context / Available Materials.

{content_text}

TASK

Generate a complete, natural, and fluent English conversation in which a user and "ChatGPT" collaboratively accomplish:

{scenario}

Generate the full dialogue now.

Figure 10: User prompt used for evidence data generation.

SYSTEM PROMPT FOR EVIDENCE DATA GENERATION

You are a **dialogue-generation assistant**. Generate a complete, natural, and fluent English conversation between a user and “ChatGPT” that collaboratively accomplishes the user’s task.

Core Premise (Non-negotiable).

- Only the generator (you) can see **hidden preferences**.
- The ChatGPT character in the dialogue **cannot** see hidden preferences.
- ChatGPT may revise outputs only based on:
 - what the user says in the dialogue, and
 - what ChatGPT produced in the previous turn.

Dialogue Format (Strict).

- Use only two speaker labels: **User:** and **ChatGPT:**.
- The dialogue is entirely in English.
- Exactly **{turn_count}** turns in total, alternating lines starting with User: then ChatGPT:.
- Each turn contains **one speaker line only**; no narration, no stage directions, no extra labels, no blank lines.

Hard Rules.

1. **First user turn (no preference disclosure).** The first User: turn states only the task objective and necessary context/materials. It must not mention hidden preferences, “instance”, or any style or format constraints.
2. **Immediate draft after turn 1.** After the first user message, ChatGPT must produce an initial complete deliverable. No clarification questions about preferences, formatting, structure, tone, or style are allowed.
3. **Multi-round revision is mandatory.** After Draft 1, the user provides revisions over several turns. Each ChatGPT response must update the content rather than repeat prior versions.
4. **Preference points per user turn.** Starting only after Draft 1, each new user turn introduces one to three preference points. Preferences must be expressed **implicitly** and never stated explicitly.
5. **User voice constraints.** The user uses direct commands, not polite questions. Forbidden expressions include “Could you...”, “Can you...”, “Would you mind...”, and “Is it possible...”. User feedback should be brief (one to two sentences).
6. **No meta-leaks.** ChatGPT must never mention hidden preferences, “instance”, or prompt mechanics.
7. **Coverage requirement.** By the final turn, the output must satisfy all hidden preference points. Every distinct preference point must appear explicitly and concretely somewhere in the dialogue.
8. **Materials handling (conditional).** If required source materials are provided, include them verbatim and in full.
9. **Deliverable type.** **{deliverable_type_instruction}**
10. **No raw placeholders.** Do not output placeholders such as “[Company Name]”. If specifics are missing, omit them or use a plausible generic name.
11. **Response length.** Each ChatGPT: turn should be concise, typically under 230 words.

Dialogue Flow (Required).

- Task entry: user starts directly with the objective.
- Draft 1: ChatGPT produces a full first draft immediately.
- Iterative refinement: user reveals preferences gradually; ChatGPT revises accordingly.
- End with a short user closing line such as “Alright, this works.”

Output Requirement. Generate the complete dialogue in one go, following all constraints above.

Hidden Preference Instance. **{instance}**

Figure 11: System prompt used for evidence data generation.