PIPA: PROFILE-INFERRED PERSONALIZATION ADAPTATION FOR PROACTIVE AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Proactive agents must decide whether and how to assist before users explicitly ask for help. This requires inferring *how a particular user tends to respond to assistance* from limited interaction history. We introduce **PIPA** (Profile-Inferred Personalization Adaptation), which represents each user through compact *interaction skills* that pair recurring contexts and intervention patterns with behavioral evidence of receptivity. PIPA maintains a probabilistic belief over these preferences and uses profile-conditioned simulation to choose among remaining silent, asking, suggesting, or acting. Across three LLM backbones, PIPA improves proactive timing, and on two of the three it substantially improves intention prediction. The reconstructed profiles are user-specific and predictive of held-out behavior: replacing a user's profile with another user's reduces proactive-timing AUROC from 0.891 to 0.492. Controlled ablations separately test the contributions of user-specific representation, posterior uncertainty, and uncertainty-sensitive intervention selection. In particular, we compare posterior sampling against a compute-matched posterior-mean variant and isolate the effect of the uncertainty penalty while holding the user representation and simulator fixed. Attached to five agent harnesses on π-Bench, PIPA improves all 30 harness–backbone–metric comparisons, with mean gains of $+3.55$ PROC and $+3.79$ COMP.

1 INTRODUCTION

Large language model agents are increasingly expected to anticipate user needs and provide assistance without an explicit request (Tang et al., 2026; Yang et al., 2025; Nathani et al., 2026). Unlike ordinary response personalization, proactive assistance requires the agent to decide not only *what* may help, but also *whether and how* a particular user wants it to intervene. A poor user model can therefore lead to interruption, unnecessary assistance, or excessive initiative. Effective proactivity requires learning how an individual tends to respond to assistance from limited interaction history and anticipating their behavior in future situations.

Existing personalized agents typically rely on retrieved histories, long-context records, summarized memories, or explicit user profiles (Salemi et al., 2024; Packer et al., 2023; Zhao et al., 2025). Proactive systems additionally use personas, user simulators, or explicit preference elicitation (Kim et al., 2026; Sun et al., 2025; Li et al., 2026). Naturally occurring interactions provide useful preference signals (Shi et al., 2026), but this evidence is sparse and uneven: a behavior observed once should not carry the same confidence as one repeatedly confirmed. Existing representations record what a user did or said, but not how strongly each preference is supported, and so cannot tell the agent when it knows enough to act. Appendix A discusses related work in full.

To address this problem, we introduce Profile-Inferred Personalization Adaptation (**PIPA**), which represents each user through a compact library of *interaction skills*. Whereas recent work extracts reusable skills from interaction experience to improve task execution (Liu et al., 2024; Yang et al., 2026; Wang et al., 2026; Mi et al., 2026), PIPA uses skills to represent *how a particular user prefers the agent to behave* in recurring situations. Each skill pairs a context and intervention pattern with behavior-grounded evidence of user receptivity, over which PIPA maintains a probabilistic belief. At decision time, it samples plausible user states, simulates how each would respond to candidate interventions (Park et al., 2023; Kim et al., 2026; Sun et al., 2025), and chooses among remaining silent, asking, suggesting, or acting while penalizing actions whose value is unstable across those states. Thin evidence therefore leads to lower initiative, and stronger evidence licenses more.

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

We evaluate PIPA from user modeling to downstream proactive behavior. Reconstructed profiles recover user-specific information on both π-Bench (Zhang et al., 2026) and KnowU-Bench (Chen et al., 2026) and remain predictive of held-out behavior. Across three LLM backbones, PIPA improves proactive timing and intention prediction on ProAgentBench (Tang et al., 2026), with gains of up to 23.9 macro-F1 points on 14-way intention classification. These gains depend on modeling the correct individual: replacing the user's profile with another user's reduces proactive-timing AUROC from 0.891 to 0.492, below the 0.752 of population-only modeling. User identity is also most informative precisely when the current context is ambiguous.

Our results further show that personalization is not only a question of recovering the correct user profile, but also of representing how strongly that profile is supported by evidence. PIPA's probabilistic belief becomes more concentrated as user observations accumulate and is better calibrated than probabilities inferred from the same profile through the LLM alone. We further isolate two distinct roles of uncertainty. First, posterior sampling represents multiple user states that remain plausible under limited behavioral evidence. Second, the intervention policy can explicitly penalize actions whose predicted utility varies across those plausible states. We evaluate these components separately using compute-matched ablations, so that additional simulator calls are not confounded with access to posterior uncertainty. Finally, PIPA is not tied to a particular agent architecture. Across five heterogeneous agent harnesses and three LLM backbones on π-Bench, it improves all 30 evaluated harness–backbone–metric comparisons. Taken together, these results support our broader view of personalized proactivity: an agent should not merely remember a user, but construct a behavioral model that can predict how that individual is likely to respond, quantify the evidence behind that prediction, and use both to determine how much initiative to take.

Our contributions are:

- **User modeling.** We introduce interaction skills as a compact, behavior-grounded representation of individual users. Rather than describing users only through static attributes or textual summaries, PIPA models recurring situations together with evidence about how each user responds to different forms of assistance. This provides a user model that is directly testable against future behavior and naturally expresses uncertainty when evidence is sparse.
- **Uncertainty-aware personalized initiative.** We show how a learned user model can directly shape an agent's intervention policy. PIPA combines probabilistic preference estimation with profile-conditioned user simulation to decide not only what assistance may be useful, but also whether the agent should remain silent, ask, suggest, or act. Because a wrong intervention costs interruption, unnecessary assistance, and loss of user control, PIPA propagates uncertainty from the user model into intervention selection and behaves more conservatively when evidence about the user is insufficient. This extends personalization from adapting generated content to adapting the level of agent initiative.
- **End-to-end empirical validation.** We evaluate PIPA from profile reconstruction and held-out behavior prediction to proactive timing, intention prediction, cold-start adaptation, uncertainty calibration, and transfer across heterogeneous agent harnesses. The results show that performance depends on modeling the correct individual, that calibrated uncertainty improves intervention decisions, and that the learned personalization layer transfers across multiple LLM backbones and agent architectures.

2 PIPA: Profile-Inferred Personalization Adaptation

2.1 Overview

At interaction step t, the agent observes the current context $x_{u,t}$ and user history

$$H_{u,t} = \{(x_{u,\tau}, a_{u,\tau}, y_{u,\tau})\}_{\tau<t},$$

where $a_{u,\tau}$ is the previous intervention and $y_{u,\tau}$ is observed behavioral feedback. The agent selects

$$a_{u,t} \in \mathcal{A} = \{\texttt{silent}, \texttt{ask}, \texttt{suggest}, \texttt{act}\}.$$

Because user receptivity is latent and only partially observed, PIPA represents the user through interaction skills and maintains uncertainty over the preferences encoded by them. At decision time,

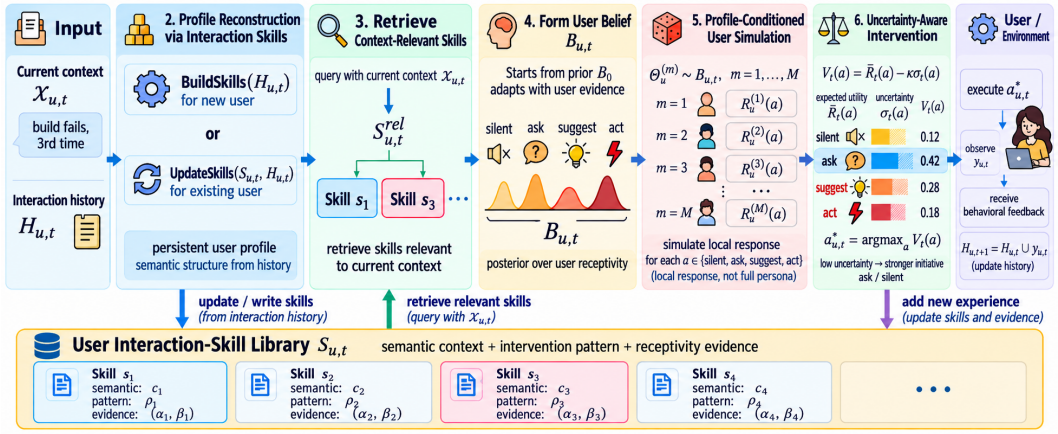


Figure 1: **Overview of PIPA.** Given the current context $x_{u,t}$ and interaction history $H_{u,t}$, PIPA reconstructs or updates a persistent user profile as an interaction-skill library $\mathcal{S}_{u,t}$, retrieves the context-relevant subset $\mathcal{S}^{\text{rel}}_{u,t}$, and forms a probabilistic belief $B_{u,t}$ over user receptivity. It then samples plausible user states, performs profile-conditioned simulation for candidate interventions, and selects $a^\star_{u,t}$ using expected utility penalized by uncertainty. After execution, behavioral feedback $y_{u,t}$ updates the interaction history, posterior belief, and skill library for future decisions.

it simulates plausible user responses to candidate interventions and selects an action according to both expected utility and uncertainty:

$$H_{u,t} \to \mathcal{S}_{u,t} \to B_{u,t} \to \text{User Simulation} \to a^\star_{u,t}.$$

Here, $\mathcal{S}_{u,t}$ denotes the interaction-skill library and $B_{u,t}$ the current probabilistic user belief.

2.2 Interaction Skills

PIPA converts interaction history into a compact library $\mathcal{S}_{u,t} = \{s_{u,1}, \ldots, s_{u,J}\}$, where

$$s_{u,j} = (c_j, \rho_j, \{\alpha_{u,j,a}, \beta_{u,j,a}\}_{a \in \mathcal{A}}). \tag{1}$$

Here, c_j denotes a recurring behavioral context, ρ_j summarizes the interaction pattern, and $(\alpha_{u,j,a}, \beta_{u,j,a})$ stores evidence about receptivity to intervention a.

Unlike a descriptive user profile, an interaction skill captures *how the user prefers the agent to behave* in a recurring situation, e.g., welcoming suggestions after repeated errors or preferring confirmation before external actions. We separate semantic abstraction from preference estimation: an LLM identifies recurring contexts and interaction patterns, while receptivity is estimated only from observed behavioral evidence. Unlabeled histories can therefore initialize the semantic structure of the profile without being treated as preference labels (Appendix D.1). At each decision, only skills relevant to the current context are retrieved. Construction, retrieval, and update details are given in Appendix B.

2.3 Bayesian User Adaptation

For each relevant skill j and intervention a, PIPA models user receptivity $\theta_{u,j,a} \in [0, 1]$ with a Beta–Bernoulli belief:

$$\theta_{u,j,a} \mid H_{u,t} \sim \text{Beta}\left(\alpha^{(0)}_{j,a} + n^+_{u,j,a},\ \beta^{(0)}_{j,a} + n^-_{u,j,a}\right), \tag{2}$$

where $(\alpha^{(0)}, \beta^{(0)})$ are population-level priors and (n^+, n^-) summarize positive and negative behavioral evidence.

The posterior provides both an estimate of the user's preference and the uncertainty in that estimate. Population priors support cold start, while user-specific evidence progressively shifts and concentrates the belief as more interactions are observed.

2.4 PROFILE-CONDITIONED USER SIMULATION

Given the relevant skills $\mathcal{S}_{u,t}^{\text{rel}}$, PIPA follows posterior-sampling approaches to decision making under uncertainty (Daniel et al., 2018; Guez et al., 2012) and samples plausible user states

$$\Theta_u^{(m)} \sim B_{u,t}, \qquad m = 1, \ldots, M, \tag{3}$$

and evaluates each candidate intervention through a local, profile-conditioned simulation:

$$R_t^{(m)}(a) = \mathbb{E}\left[r_t \mid x_{u,t}, a, \Theta_u^{(m)}, \mathcal{S}_{u,t}^{\text{rel}}\right]. \tag{4}$$

The simulator predicts the user's response to the current intervention rather than attempting to model a complete human persona. The sampled state $\Theta_u^{(m)}$, the reward r_t, and the values of M and κ are specified in Appendix B.6 and Table 9. PIPA selects

$$a_{u,t}^{\star} = \arg\max_{a \in \mathcal{A}} \left[\frac{1}{M}\sum_{m=1}^{M} R_t^{(m)}(a) - \kappa\ \text{Std}_m\left[R_t^{(m)}(a)\right]\right], \tag{5}$$

a mean–variance criterion (Markowitz, 1952), where κ controls sensitivity to uncertainty. The first term favors interventions with high expected utility, while the second penalizes actions whose predicted utility is unstable across plausible user states. Importantly, Equation 5 does not impose an explicit commitment ordering over actions: the same coefficient κ is applied to every candidate. Any tendency toward lower-commitment behavior under uncertainty arises through the predicted task benefit, interruption cost, and autonomy cost of the candidate interventions. We therefore treat this behavior as an empirical property to be measured rather than one guaranteed by the form of the objective.

3 EXPERIMENTAL SETUP

We evaluate whether PIPA **(RQ1)** learns user-specific representations predictive of future behavior, **(RQ2)** improves proactive assistance under limited or ambiguous evidence, and **(RQ3)** benefits from uncertainty-aware intervention.

Benchmarks and protocol. We evaluate on three complementary benchmarks: **ProAgentBench** (Tang et al., 2026) for proactive timing, intention prediction, and personalized suggestions; **π-Bench** (Zhang et al., 2026) for profile reconstruction and transfer across agent harnesses; and **KnowU-Bench** (Chen et al., 2026) for independent profile recovery. Following PARE (Nathani et al., 2026), interactions are processed chronologically: only past behavior is available at each decision, while later interactions are held out for evaluation. We vary observed history to study cold-start adaptation. Dataset and split details are in Appendix C.1.

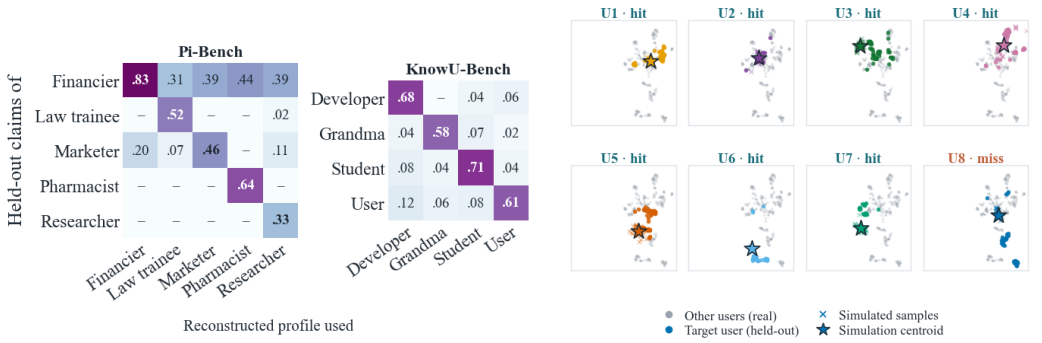
Models and baselines. We evaluate Qwen3.7-Plus, DeepSeek-V4-Flash, and GPT-4o-mini. Baselines include ReAct (Yao et al., 2022), RAG (Lewis et al., 2020), Long-Context (full history) (Wu et al., 2024), MemGPT (Packer et al., 2023), and a structured knowledge-graph memory baseline following Hogan et al. (2021). We further test population-only and shuffled profiles, PIPA-Mean, and PIPA with $\kappa = 0$. Details are provided in Appendix C.3.

Metrics and evaluation. We evaluate profile recovery and held-out behavioral prediction for user modeling; AUROC, AUPRC, F1, FAR, ECE (Guo et al., 2017), and accuracy for proactive timing; and macro classification metrics and SemanticPass@k for intention prediction and suggestion generation, respectively. Semantic outputs are evaluated by Claude Sonnet 5 using a fixed LLM-as-a-judge rubric (Zheng et al., 2023); directly observable outcomes use ground-truth labels. Metric definitions and the judge prompt are given in Appendices C.5 and C.6.

4 RESULTS

4.1 PIPA LEARNS USER-SPECIFIC BEHAVIORAL PROFILES

We first ask whether PIPA learns information that is specific to an individual user and remains predictive of future behavior. Figure 2 evaluates both properties. For profile reconstruction, the corresponding user's profile achieves the highest held-out claim recovery for all $5/5$ users on π-Bench and all $4/4$ users on KnowU-Bench ($p = 3.2 \times 10^{-4}$ and $p = 3.9 \times 10^{-3}$, the probability that every row maximum falls on the diagonal by chance). This diagonal structure indicates that the reconstructed profiles preserve individual-specific information rather than only generic population regularities. The learned representation also remains predictive beyond the claims used to evaluate



(a) Profile reconstruction.

(b) Profile-conditioned user simulation.

Figure 2: **From profile reconstruction to user simulation.** **(a)** Fraction of each user's held-out claims (row) recovered by each reconstructed profile (column). The corresponding-user profile is the row maximum for every user on both π-Bench and KnowU-Bench. **(b)** Profile-conditioned predictions compared with the target user's held-out behavior. Hit/miss is computed in the original 1024-D embedding space; the 2-D UMAP projection (McInnes et al., 2018) is shown for visualization only.

reconstruction. Using only the observed user history, PIPA's profile-conditioned behavioral predictions identify the correct user's held-out behavioral distribution for $4/8$ users (randomly choose from ProAgentBench and KnowU-Bench) in the original 1024-D embedding space, against a chance level of $1/8$ (one-sided binomial test, $p \approx 0.011$). Together, these results show that interaction skills contain user-specific information that remains predictive of held-out behavior.

4.2 PIPA Improves When and What Assistance to Provide

When to intervene. Table 1 evaluates proactive timing on ProAgentBench. Across Qwen3.7-Plus, DeepSeek-V4-Flash, and GPT-4o-mini, PIPA achieves the highest AUROC ($0.891/0.854/0.910$), AUPRC ($0.885/0.840/0.910$), F1 ($0.821/0.898/0.812$), and accuracy ($0.793/0.886/0.798$), and the lowest ECE on all three backbones. These gains do not come from intervening indiscriminately. PIPA's FARs are $0.296/0.145/0.200$: the lowest with Qwen3.7-Plus, close to the lowest with DeepSeek-V4-Flash, and tied for the lowest with GPT-4o-mini. Personalized user modeling therefore improves both the recognition of useful assistance opportunities and restraint when intervention is unwanted. Nor are the gains bought with a larger context budget: PIPA lies on the cost–quality Pareto frontier on every backbone (Figure 4).

What assistance to provide. Personalized proactivity also requires predicting what form of assistance the user is likely to want. Table 2 evaluates 14-way intention classification and personalized suggestion generation. PIPA substantially improves intention prediction with Qwen3.7-Plus (macro-F1 $27.0 \rightarrow 40.8$) and DeepSeek-V4-Flash ($21.6 \rightarrow 45.5$); these gains, and the corresponding gains in macro-precision and accuracy, all exceed 2σ. With GPT-4o-mini the difference (22.1 versus 20.4) is within noise, so we do not claim an improvement for that backbone. For free-form suggestions, PIPA attains the highest mean SemanticPass@3 on all three backbones and the highest mean SemanticPass@1 on two of three, but none of these gaps exceeds 2σ; we therefore regard suggestion quality as comparable to the strongest baseline rather than improved. The learned user representation thus clearly supports deciding *when* to assist and predicting what the user intends, whereas gains on open-ended suggestion generation are not yet distinguishable from the strongest baselines.

4.3 Personalization Matters Most under Ambiguous and Limited Evidence

When does personalization matter? The benefit of user modeling depends on how informative the current observation already is. In the clear-signal setting, current context alone is highly predictive of whether assistance is wanted (AUROC 0.896), slightly exceeding an identity-only probe (0.889). In the vague-signal setting, this relationship reverses: context-only AUROC drops to 0.674, while identity reaches 0.717. This reversal suggests that personalization contributes most when the current context alone is insufficient to determine the appropriate interaction: the same observable sit-

Table 1: **Proactive timing on ProAgentBench.** Results are evaluated over 904 decisions from 17 users. AUROC and AUPRC use the continuous non-silent-versus-silent score in Equation 10. ECE uses a leave-one-user-out calibrated probability, and F1/FAR/accuracy use a threshold selected exclusively from the corresponding non-held-out calibration users (Appendix C.4).

Backbone	Method	Ranking		Decision			
		AUROC	AUPRC	F1	FAR	ECE	Acc
Qwen3.7-Plus	RAG	0.819	0.844	0.603	0.457	0.115	0.571
	ReAct	0.484	0.531	0.677	0.998	0.422	0.512
	Long-Context	0.742	0.725	0.786	0.433	0.153	0.738
	MemGPT	0.436	0.535	0.721	0.939	0.460	0.575
	Knowledge Graph	0.849	0.873	0.754	0.411	0.137	0.710
	PIPA (ours)	**0.891**	**0.885**	**0.821**	**0.296**	**0.095**	**0.793**
DeepSeek-V4-Flash	RAG	0.757	0.811	0.585	0.628	0.401	0.512
	ReAct	0.497	0.636	0.708	0.998	0.437	0.549
	Long-Context	0.852	0.811	0.860	**0.120**	0.393	0.852
	MemGPT	0.601	0.606	0.708	1.000	0.398	0.548
	Knowledge Graph	0.643	0.629	0.689	0.714	0.375	0.587
	PIPA (ours)	**0.854**	**0.840**	**0.898**	0.145	**0.344**	**0.886**
GPT-4o-mini	RAG	0.893	0.894	0.807	0.284	0.142	0.781
	ReAct	0.429	0.691	0.610	0.908	0.399	0.521
	Long-Context	0.776	0.747	0.801	0.531	0.118	0.739
	MemGPT	0.409	0.509	0.708	1.000	0.475	0.548
	Knowledge Graph	0.875	0.894	0.784	**0.200**	0.134	0.773
	PIPA (ours)	**0.910**	**0.910**	**0.812**	**0.200**	**0.049**	**0.798**

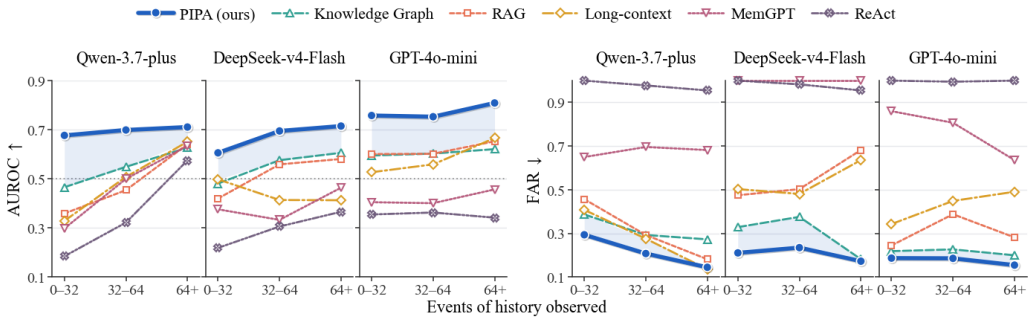


Figure 3: **Cold-start personalization as user history accumulates.** We evaluate proactive timing after observing different amounts of user history (0–32, 32–64, and 64+ events). The top row reports AUROC (↑), while the bottom row reports FAR (↓), across three LLM backbones. PIPA performs strongly in the earliest-history regime and remains comparatively stable as additional evidence accumulates, while retrieval- and context-based baselines generally benefit more from longer histories.

uation may warrant different actions for different users, making user-specific behavioral information particularly valuable under ambiguity.

Cold-start adaptation. Figure 3 examines performance as user-specific evidence accumulates. PIPA performs strongly from the earliest-history regime and remains comparatively stable as additional interactions are observed, whereas retrieval- and context-based baselines generally improve more substantially only after accumulating larger histories. PIPA also maintains low false-alarm rates across most history regimes, indicating that its cold-start advantage does not arise from intervening more aggressively.

What drives PIPA's gains? Table 5 first isolates whether performance depends on modeling the *correct individual*. Replacing the target user's profile with a shuffled profile reduces AUROC from

Table 2: **Intention classification and personalized suggestion quality.** *Intention*: 14-class classification, 17 users. *Suggestion*: SemanticPass@1/@3, judged by Claude Sonnet 5 under a fixed rubric shared across methods and backbones (Appendix C.6). Percentages, mean $\pm$ std over 5 seeds; the std reflects seed variation only and, given 17 users per corpus, understates user-level uncertainty. **Bold**: best mean per backbone and metric. A gap is claimed as an improvement only if it exceeds 2σ, where $\sigma = \sqrt{\sigma_1^2 + \sigma_2^2}$ is the std of the difference between the two methods. The two tasks use different corpora, so scores are not directly comparable.

		User intention classification			Suggestion quality	
Backbone	Method	macro-F1 ↑	macro-P ↑	Acc ↑	pass@1 ↑	pass@3 ↑
Qwen3.7-Plus	RAG	26.2 ±1.4	31.3 ±2.7	32.7 ±0.9	19.3 ±3.1	30.9 ±2.2
	ReAct	17.3 ±2.1	25.1 ±1.3	22.2 ±3.4	18.2 ±0.7	30.1 ±2.8
	Long-Context	27.0 ±0.6	30.5 ±3.2	31.3 ±2.0	19.5 ±1.7	29.3 ±3.6
	MemGPT	15.7 ±2.9	21.3 ±0.4	22.4 ±1.8	16.5 ±3.3	22.4 ±1.1
	Knowledge Graph	17.4 ±1.2	18.3 ±2.4	32.1 ±0.8	**19.8** ±2.6	41.8 ±3.5
	PIPA (ours)	**40.8** ±0.5	**42.4** ±1.9	**55.9** ±2.3	18.9 ±3.0	**44.4** ±1.6
DeepSeek-V4-Flash	RAG	17.3 ±3.4	19.5 ±1.0	32.5 ±2.5	28.6 ±0.6	40.8 ±3.2
	ReAct	12.3 ±1.8	23.2 ±2.9	23.0 ±0.3	27.9 ±3.6	35.6 ±1.4
	Long-Context	21.6 ±2.2	27.2 ±0.9	31.5 ±1.5	28.0 ±2.7	43.4 ±3.1
	MemGPT	16.1 ±3.0	27.6 ±1.7	22.2 ±2.8	24.7 ±0.5	33.3 ±2.1
	Knowledge Graph	17.6 ±1.1	20.1 ±3.3	31.5 ±2.6	31.0 ±1.9	45.1 ±0.8
	PIPA (ours)	**45.5** ±0.7	**44.5** ±2.0	**55.1** ±1.3	**38.5** ±3.4	**46.7** ±2.5
GPT-4o-mini	RAG	20.4 ±2.6	24.5 ±1.2	27.1 ±3.5	18.8 ±0.9	27.3 ±2.0
	ReAct	16.0 ±1.9	26.5 ±3.1	23.0 ±0.4	18.2 ±2.8	30.1 ±1.5
	Long-Context	20.2 ±3.2	28.6 ±0.8	22.4 ±2.4	16.5 ±1.6	26.0 ±3.7
	MemGPT	19.5 ±1.3	**33.3** ±2.2	19.2 ±0.6	14.1 ±3.0	25.3 ±1.8
	Knowledge Graph	15.5 ±2.7	20.2 ±1.4	24.0 ±3.3	19.8 ±0.5	30.8 ±2.3
	PIPA (ours)	**22.1** ±0.9	31.4 ±2.5	**29.9** ±1.7	**24.3** ±3.1	**33.6** ±1.2

Table 3: **What does the interaction-skill representation contribute?** All rows share the same ProAgentBench decisions and utility function and differ only in the user representation: *RAG+Policy* substitutes retrieved raw history for skills, and *Contextual-Beta* replaces skills with frozen embedding clusters while keeping the full Bayesian pipeline. Qwen3.7-Plus.

Method	AUROC ↑	AUPRC ↑	F1 ↑	FAR ↓	ECE ↓
RAG	0.819	0.844	0.603	0.457	0.115
RAG+Policy	0.823	0.846	0.610	0.481	0.111
Contextual-Beta	0.801	0.795	0.609	0.514	0.123
PIPA	**0.891**	**0.885**	**0.821**	**0.296**	**0.095**

Table 4: **Uncertainty-aware intervention**, four-action utility setting. F1/FAR/Acc use the same binary proactive-timing evaluation as Tables 1 and 5; Utility additionally scores the four-action choice under Eq. 16. All variants share the same interaction skills and evaluation subset.

Method	F1 ↑	FAR ↓	Acc ↑	Utility ↑
PIPA-Mean	0.803	0.315	0.729	0.372
PIPA ($\kappa = 0$)	0.745	0.301	0.594	0.175
PIPA	**0.821**	**0.296**	**0.793**	**0.395**

Table 5: **Profile-specificity ablation**, ProAgentBench (Qwen3.7-Plus). *Population only* drops the individual profile and decides from the population prior; *Shuffled profile* substitutes another user's, observation unchanged. AUROC is threshold-free; F1/FAR/Acc use the leave-one-user-out threshold.

User representation	AUROC ↑	F1 ↑	FAR ↓	Acc ↑
Population only	0.752	0.609	0.594	0.541
Shuffled profile	0.492	0.709	0.995	0.550
PIPA	**0.891**	**0.821**	**0.296**	**0.793**

0.891 to 0.492 and increases FAR from 0.296 to 0.995; using only population-level information also degrades AUROC to 0.752 and raises FAR to 0.594. PIPA's gains therefore depend on user-specific behavioral information rather than merely supplying additional profile context. Table 3 next isolates the representation itself while holding the downstream decision mechanism fixed. Giving PIPA's candidate-action simulator access to retrieved raw history instead of interaction skills (RAG+Policy)

Table 6: **PIPA improves every harness under every backbone on π-Bench.** *Base* is the unmodified harness and *+PIPA* (tinted) is the same harness with PIPA attached, with the gain over Base shown in green. All 30 paired comparisons (5 harnesses $\times$ 3 backbones $\times$ 2 metrics) favour PIPA, with mean gains of $+3.55$ PROC and $+3.79$ COMP. PROC measures the percentage of hidden user intents resolved proactively, either by direct completion or targeted elicitation; COMP measures the percentage of verifiable checklist requirements satisfied by the final trajectory. Both metrics range from 0 to 100 and are averaged over the full π-Bench evaluation of 100 multi-turn tasks across 5 personas (Zhang et al., 2026). Higher is better. **Bold**: best configuration per backbone and metric.

	Qwen3.7-Plus				**DeepSeek-V4-Flash**				**GPT-4o-mini**			
	PROC ↑		COMP ↑		PROC ↑		COMP ↑		PROC ↑		COMP ↑	
Harness	Base	**+PIPA**	Base	**+PIPA**	Base	**+PIPA**	Base	**+PIPA**	Base	**+PIPA**	Base	**+PIPA**
OpenHands	61.01	64.91 +3.90	50.11	56.02 +5.91	63.63	65.43 +1.80	52.08	52.38 +0.30	46.81	47.61 +0.80	53.90	54.44 +0.54
Nanobot	64.51	**69.05** +4.54	74.32	**77.40** +3.08	67.93	69.88 +1.95	76.73	79.46 +2.73	51.27	**54.79** +3.52	52.77	**56.62** +3.85
Pi	51.07	56.22 +5.15	36.77	39.62 +2.85	61.81	68.39 +6.58	79.07	**84.22** +5.15	49.12	53.32 +4.20	44.20	47.45 +3.25
Codex	49.66	54.31 +4.65	23.27	29.76 +6.49	66.33	**71.09** +4.76	48.77	55.93 +7.16	46.95	47.55 +0.60	32.08	35.87 +3.79
OpenCode	45.95	51.61 +5.66	14.33	20.00 +5.67	61.91	63.03 +1.12	57.10	60.30 +3.20	42.94	46.94 +4.00	7.88	10.69 +2.81
Average	*54.44*	*59.22* +4.78	*39.76*	*44.56* +4.80	*64.32*	*67.56* +3.24	*62.75*	*66.46* +3.71	*47.42*	*50.04* +2.62	*38.17*	*41.01* +2.85

leaves proactive timing essentially unchanged relative to RAG (0.823 versus 0.819 AUROC), and replacing semantic skills with frozen embedding-space clusters while retaining the full Bayesian pipeline (Contextual-Beta) does not help either (0.801). Only the complete interaction-skill representation reaches 0.891, indicating that the gains follow from the representation rather than from the decision mechanism or from Bayesian personalization alone. Table 4 separately isolates uncertainty-aware decision making. Collapsing the posterior to its mean reduces utility from 0.395 to 0.372, while removing uncertainty-sensitive action selection ($\kappa = 0$) reduces it to 0.175. Full PIPA attains the best F1, false-alarm rate, accuracy, and utility among the three variants. These results indicate that effective proactive personalization requires both learning what the user prefers and reasoning about how confidently that preference is known.

Does the belief state improve with evidence? We further examine PIPA's probabilistic belief as interaction history accumulates. Posterior variance decreases consistently across history regimes, while PIPA's explicit belief achieves the lowest ECE in all nine backbone–history settings compared with probabilities inferred from the same profile and alternative memory methods (Appendix D.2). Thus, additional evidence makes the belief more concentrated without sacrificing calibration.

4.4 PIPA Transfers across Agent Harnesses

Finally, we test whether PIPA acts as a reusable personalization layer rather than depending on one particular agent architecture. Table 6 reports per-harness results and averages across five agent harnesses on π-Bench. PIPA improves both PROC and COMP on average for every tested backbone, with overall mean gains of $+3.55$ and $+3.79$ points, respectively, and all 30 harness–backbone–metric comparisons are positive. Individual gains range from $+0.30$ to $+7.16$ points. These results suggest that the learned user representation can benefit diverse agent systems without modifying their underlying execution policy.

5 Discussion

Our results highlight an additional challenge introduced by personalized proactivity. Beyond generating an appropriate response, the agent must decide whether the available evidence about a particular user is sufficient to justify initiating assistance. Errors therefore affect not only response quality, but also interruption, user attention, and control. PIPA addresses this challenge by representing recurring user behavior through interaction skills and maintaining an explicit probabilistic belief over user receptivity.

From user memory to intervention policy. Our results suggest that proactive personalization requires more than retrieving or summarizing user history. The profile-specificity ablation shows that modeling the correct individual matters, while the uncertainty ablations and calibration analysis show that the strength of the supporting evidence also matters. A useful user model should therefore

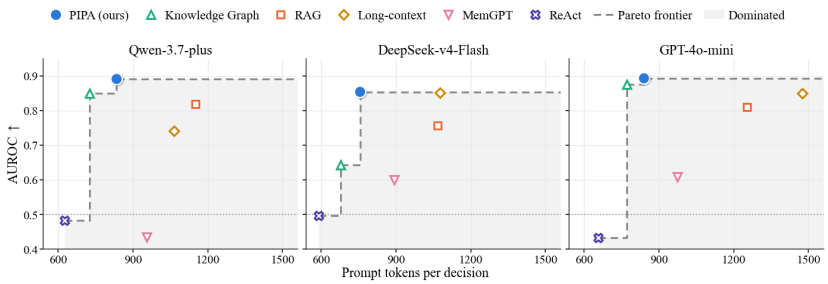


Figure 4: **Cost–quality trade-off.** Prompt tokens per decision vs. proactive-timing AUROC on three backbones; the shaded region is Pareto-dominated. PIPA lies on the frontier and attains the highest AUROC on every backbone. Cheaper methods (ReAct, Knowledge Graph) trade away accuracy, while Long-Context comes close to PIPA only at 1.4–1.8× its token cost, so PIPA's gains do not come from a larger context budget.

expose not only *what* the agent believes about the user, but also how strongly that belief is supported, so that uncertainty can directly influence the level of initiative.

Interaction skills as a behavioral user representation. PIPA's interaction skills are intentionally narrower than a general textual persona. They organize recurring contexts around a decision-relevant question: how does this user tend to respond to different forms of assistance? Semantic patterns are extracted from interaction history, while receptivity is grounded in observed behavior. Our reconstruction, unlabeled-profile, and held-out simulation results provide evidence that this representation preserves user-specific information that remains useful beyond the interactions from which it was constructed.

When personalization matters. Personalization is not equally valuable in every interaction. When the current context already provides a strong signal, user-specific information adds comparatively little. Its value increases under ambiguity, where the observable context alone is less predictive of whether assistance is wanted. Together with the cold-start results, this suggests a natural adaptation strategy: begin from population-level regularities, specialize as individual evidence accumulates, and increase reliance on personalization where contextual evidence is insufficient.

Limitations. Several limitations remain. First, behavioral feedback is an imperfect proxy for user preference: acceptance, rejection, or subsequent actions may also reflect task constraints, transient goals, or external factors. Second, PIPA's user simulation is local and intervention-conditioned; it predicts how a user may respond to candidate actions rather than attempting to model a complete human persona. Third, the quality of learned interaction skills depends on the coverage of observed history, and rare or changing preferences may remain poorly estimated. Fourth, while PIPA improves structured prediction tasks, its gains on open-ended suggestion generation are not distinguishable from the strongest baselines, and its intention-prediction gains depend on the backbone. Fifth, several evaluations involve a small number of users (4–17 per benchmark), so per-user variation is only partially captured. Finally, our evaluation relies on logged traces and benchmark environments; a longitudinal study with live users would more directly measure the interruption costs that motivate this work. Future work could study richer feedback signals, non-stationary user preferences, and continual belief revision.

6 Conclusion

We introduced **PIPA**, a personalized proactive agent that learns behaviorally grounded interaction skills, maintains probabilistic beliefs over user receptivity, and uses profile-conditioned simulation to decide whether and how strongly to intervene. Across profile reconstruction, held-out user simulation, proactive timing, intention prediction, cold-start adaptation, and multiple agent harnesses, PIPA improves personalized proactive behavior; the exception is open-ended suggestion quality, where differences from the strongest baselines are within noise. Ablations further show that these gains depend both on modeling the correct individual and on explicitly accounting for uncertainty during intervention selection. Overall, our results suggest that effective proactive personalization requires more than remembering what a user has done before: an agent should also reason about how confidently it understands that user before taking initiative.

AI USE STATEMENT

In this work, we used generative AI tools for (i) language editing, i.e., grammar and clarity revisions of author-written text; (ii) literature search, to help identify potentially relevant prior work; and (iii) coding and formatting assistance, including plotting scripts for figures and LaTeX formatting of equations and the judge-prompt figure. Large language models are also part of our methodology: they serve as the agents under study and as the automatic judge for SemanticPass@k, as described in Appendix C.6 and Figure 6.

We have not used generative AI tools to generate, modify, or select experimental data or results, or to fabricate citations.

We have reviewed all AI-assisted work. Every reference surfaced with AI assistance was located and checked against the original source by the authors before inclusion. AI-suggested text edits were reviewed and revised by the authors. AI-assisted plotting code was reviewed by all authors, and each figure was checked against the raw exported results. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

PIPA builds models of individual users from their interaction histories, which raises privacy considerations. All experiments use existing benchmarks (ProAgentBench, π-Bench, and KnowU-Bench) under their released terms, and no new human-subject data were collected. In deployment, interaction-skill profiles would constitute personal data: they should remain under user control, be inspectable and deletable, and not be shared across services without consent. Because PIPA adjusts how much initiative an agent takes, an inaccurate profile could lead to unwanted interruptions or actions. The uncertainty penalty is intended to bias decisions toward lower-commitment actions when evidence is thin, but this is an empirical tendency rather than a guarantee (Section 2.4), and it does not replace explicit user controls such as confirmation before external actions. Finally, profiles inferred from behavior may indirectly encode sensitive attributes; we do not evaluate this risk, and we flag it for future work.

REPRODUCIBILITY STATEMENT

Section 2 and Appendix B describe the method, with the complete procedure in Algorithm 1. Appendix C specifies the benchmarks, backbones, baselines, ablations, evaluation protocol, and metrics, and Figure 6 gives the exact judge prompt. Code will be released upon publication.

REFERENCES

Craig Boutilier. A pomdp formulation of preference elicitation problems. *AAAI/IAAI*, 1:239–246, 2002.

Joel QL Chang. Asymptotic optimality of thompson sampling for risk-averse bandits with subgaussian rewards. *arXiv preprint arXiv:2606.09191*, 2026.

Tongbo Chen, Zhengxi Lu, Zhan Xu, Guocheng Shao, Shaohan Zhao, Fei Tang, Yong Du, Kaitao Song, Yizhou Liu, Yuchen Yan, et al. Knowu-bench: Towards interactive, proactive, and personalized mobile agent evaluation. *arXiv preprint arXiv:2604.08455*, 2026.

J Russo Daniel, Van Roy Benjamin, Kazerouni Abbas, Osband Ian, and Wen Zheng. A tutorial on thompson sampling. *Foundations and trends® in machine learning*, 11(1):1–99, 2018.

Arthur Guez, David Silver, and Peter Dayan. Efficient bayes-adaptive reinforcement learning using sample-based search. *Advances in neural information processing systems*, 25, 2012.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International conference on machine learning*, pp. 1321–1330. PMLR, 2017.

Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia d'Amato, Gerard De Melo, Claudio Gutierrez, Sabrina Kirrane, José Emilio Labra Gayo, Roberto Navigli, Sebastian Neumaier, et al. Knowledge graphs. *ACM Computing Surveys (Csur)*, 54(4):1–37, 2021.

Eric Horvitz. Principles of mixed-initiative user interfaces. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, pp. 159–166, 1999.

Jiho Kim, Junseong Choi, Woosog Chay, Daeun Kyung, Yeonsu Kwon, Yohan Jo, and Edward Choi. Propersim: Developing proactive and personalized ai assistants through user-assistant simulation. In *International Conference on Learning Representations*, volume 2026, pp. 110843–110873, 2026.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33: 9459–9474, 2020.

Stella Li, Avinandan Bose, Faeze Brahman, Simon Du, Pang Wei Koh, Maryam Fazel, and Yulia Tsvetkov. Prefdisco: Benchmarking proactive personalized reasoning. In *International Conference on Learning Representations*, volume 2026, pp. 80844–80908, 2026.

Anthony Zhe Liu, Jongwook Choi, Sungryull Sohn, Yao Fu, Jaekyeom Kim, Dong-Ki Kim, Xinhe Wang, Jaewon Yoo, and Honglak Lee. Skillact: Using skill abstractions improves llm agents. In *ICML 2024 Workshop on LLMs and Cognition*, 2024.

Fan Liu, Zhiyong Cheng, Changchang Sun, Yinglong Wang, Liqiang Nie, and Mohan Kankanhalli. User diverse preference modeling by multimodal attentive metric learning. In *Proceedings of the 27th ACM international conference on multimedia*, pp. 1526–1534, 2019.

Harry Markowitz. Modern portfolio theory. *Journal of Finance*, 7(11):77–91, 1952.

Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

Qirui Mi, Zhijian Ma, Mengyue Yang, Haoxuan Li, Yisen Wang, Haifeng Zhang, and Jun Wang. Procmem: Learning reusable procedural memory from experience via non-parametric ppo for llm agents. *arXiv e-prints*, pp. arXiv–2602, 2026.

Deepak Nathani, Cheng Zhang, Chang Huan, Jiaming Shan, Yinfei Yang, Alkesh Patel, Zhe Gan, William Yang Wang, Michael Saxon, and Xin Eric Wang. Proactive agent research environment: Simulating active users to evaluate proactive assistants. *arXiv preprint arXiv:2604.00842*, 2026.

Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G Patil, Ion Stoica, and Joseph E Gonzalez. Memgpt: Towards llms as operating systems. *arXiv preprint arXiv:2310.08560*, 2023.

Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th annual acm symposium on user interface software and technology*, pp. 1–22, 2023.

Alireza Salemi, Sheshera Mysore, Michael Bendersky, and Hamed Zamani. Lamp: When large language models meet personalization. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7370–7392, 2024.

Taiwei Shi, Zhuoer Wang, Longqi Yang, Ying-Chun Lin, Zexue He, Mengting Wan, Pei Zhou, Sujay Kumar Jauhar, Sihao Chen, Shan Xia, et al. Wildfeedback: Aligning llms with in-situ user interactions and feedback. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 36701–36725, 2026.

Weiwei Sun, Xuhui Zhou, Weihua Du, Xingyao Wang, Sean Welleck, Graham Neubig, Maarten Sap, and Yiming Yang. Training proactive and personalized llm agents. *arXiv preprint arXiv:2511.02208*, 2025.

Yuanbo Tang, Huaze Tang, Tingyu Cao, Lam Nguyen, Anping Zhang, Xinwen Cao, Chunkang Liu, Wenbo Ding, and Yang Li. Proagentbench: Evaluating llm agents for proactive assistance with real-world data. *arXiv preprint arXiv:2602.04482*, 2026.

Chenxi Wang, Zhuoyun Yu, Xin Xie, Wuguannan Yao, Runnan Fang, Shuofei Qiao, Kexin Cao, Guozhou Zheng, Xiang Qi, Peng Zhang, et al. Skillx: Automatically constructing skill knowledge bases for agents. *arXiv preprint arXiv:2604.04804*, 2026.

Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. Longmemeval: Benchmarking chat assistants on long-term interactive memory. *arXiv preprint arXiv:2410.10813*, 2024.

Qinglong Yang, Haoming Li, Haotian Zhao, Xiaokai Yan, Jingtao Ding, Fengli Xu, and Yong Li. Fingertip 20k: A benchmark for proactive and personalized mobile llm agents. *arXiv preprint arXiv:2507.21071*, 2025.

Yutao Yang, Junsong Li, Qianjun Pan, Bihao Zhan, Yuxuan Cai, Lin Du, Jie Zhou, Kai Chen, Qin Chen, Xin Li, et al. Autoskill: Experience-driven lifelong learning via skill self-evolution. *arXiv preprint arXiv:2603.01145*, 2026.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2022.

Haoran Zhang, Luxin Xu, Zhilin Wang, Runquan Gui, Shunkai Zhang, Haodi Lei, Zihao He, Bingsu He, Chicheng Qin, Tong Zhu, et al. pi-bench: Evaluating proactive personal assistant agents in long-horizon workflows. *arXiv preprint arXiv:2605.14678*, 2026.

Siyan Zhao, Mingyi Hong, Yang Liu, Devamanyu Hazarika, and Kaixiang Lin. Do llms recognize your preferences? evaluating personalized preference following in llms. In *International Conference on Learning Representations*, volume 2025, pp. 15888–15931, 2025.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623, 2023.

A Related Work

Personalized LLM agents and memory. Personalized language models and agents commonly represent users through retrieved histories, long-context interaction records, textual profiles, or persistent memories. LaMP demonstrates the value of retrieving user-specific history for personalized generation (Salemi et al., 2024), while PrefEval shows that even strong LLMs remain unreliable at inferring and following preferences over long conversations (Zhao et al., 2025). Memory architectures such as MemGPT extend this paradigm by maintaining information beyond the immediate context window (Packer et al., 2023). These approaches primarily address *what information about the user should be retained and made available to the model*. PIPA instead focuses on how behavioral evidence should be converted into a decision-relevant model of *how a particular user prefers the agent to interact*.

Proactive and personalized agents. Recent work increasingly studies personalization together with agent initiative. ProAgentBench and FingerTip 20K evaluate proactive assistance from real behavioral traces and long-term device interaction (Tang et al., 2026; Yang et al., 2025), while PARE studies active users in stateful interactive environments (Nathani et al., 2026). ProPerSim learns personalized proactive behavior through interaction with simulated users instantiated from rich personas (Kim et al., 2026), and UserVille provides configurable user preferences for jointly optimizing productivity, proactivity, and personalization (Sun et al., 2025). PrefDisco considers the complementary cold-start problem of identifying missing preferences and strategically eliciting them through interaction (Li et al., 2026). These works establish the importance of user-aware initiative, but differ in how user knowledge becomes available: preferences may be instantiated through personas or simulators, explicitly elicited, or recovered from interaction history. PIPA focuses on the latter setting, learning user-specific behavioral regularities from sparse implicit feedback and using the resulting uncertainty directly in intervention decisions.

Mixed-initiative interaction and user control. The question of when an intelligent system should take initiative predates LLM agents. Mixed-initiative interaction studies how automation and direct user control should be combined so that the system contributes at appropriate moments without unnecessarily overriding the user (Horvitz, 1999). This literature emphasizes uncertainty about user goals, the costs and benefits of automated action, and the importance of selecting an appropriate level of initiative. PIPA brings this perspective to personalized LLM agents: the agent's intervention policy depends not only on predicted task utility, but also on how strongly the individual user's behavioral evidence supports taking initiative.

Probabilistic user modeling. Uncertainty over user preferences has long been modeled using probabilistic preference learning and decision making under partial observability. Bayesian approaches maintain distributions over uncertain preferences rather than committing to point estimates (Liu et al., 2019), while preference elicitation has been formulated as maintaining and updating a belief over unknown user utility (Boutilier, 2002). PIPA builds on this general principle rather than proposing a new Bayesian estimator. Its contribution is to connect probabilistic user modeling with *interaction skills* abstracted from natural behavioral histories, and to propagate uncertainty in those skills into the amount of initiative taken by a proactive LLM agent.

B Method Details

This appendix specifies every operation in Algorithm 1. Table 7 summarizes the notation. Throughout, every operation uses only information available before the current decision, and the LLM never produces a probability, a count, a utility scalar, or the final action. Its role is limited to (i) writing the semantic description of a new skill and (ii) predicting local response quantities during simulation. All other steps are deterministic given the random seed.

Table 7: Notation.

Symbol	Meaning
$H_{u,t} = \{(x_{u,\tau}, a_{u,\tau}, y_{u,\tau})\}_{\tau<t}$	interaction history of user u before step t
$\mathcal{A} = \{\texttt{silent}, \texttt{ask}, \texttt{suggest}, \texttt{act}\}$	intervention space; commitment cost $c(a)$ in Eq. 16
$s_{u,j} = (c_j, \rho_j, \{\alpha_{u,j,a}, \beta_{u,j,a}\}_{a\in\mathcal{A}})$	interaction skill: context, pattern, receptivity counts
$z_{u,t} \in \{1, 0, \varnothing\}$	receptivity evidence mapped from feedback $y_{u,t}$
$(\alpha_a^{(0)}, \beta_a^{(0)})$	population prior for action a
$q_j,\ S_{u,t}^{\mathrm{rel}},\ w_j$	retrieval score, retrieved skills, composition weight
$\Theta_u^{(m)} = \{\theta_{u,a}^{(m)}\}_{a\in\mathcal{A}}$	m-th sampled user state, $m = 1, \ldots, M$
$R_t^{(m)}(a),\ \bar{R}_t(a),\ \sigma_t(a)$	simulated utility, its mean and std. over samples
$V_t(a) = \bar{R}_t(a) - \kappa\,\sigma_t(a)$	uncertainty-penalized score used for selection

B.1 Interaction-Skill Construction and Update

Two views of each event. PIPA keeps two views of every historical event. The *semantic view* is shown to the skill-construction model. It contains the event identifier, the chronological index, task or application metadata when the benchmark provides it, and the pre-intervention observation and user activity. It excludes the post-intervention outcome and any derived receptivity label. The *evidence view* additionally contains the action $a_{u,\tau}$ and the observed outcome $y_{u,\tau}$, and is read only by the deterministic evidence mapper (Appendix B.2). This separation guarantees that behavioral outcomes can affect a skill's counts but never its text. Histories are replayed in chronological order, one event at a time, so no future event is visible when a skill or posterior is updated.

Matching an event to a skill. For an incoming event with context $x_{u,t}$, PIPA computes

$$r_j = \cos\big(e(x_{u,t}),\, e(c_j)\big), \qquad j^* = \arg\max_j r_j,$$

where $e(\cdot)$ is the ℓ_2-normalized `qwen3.7-text-embedding` embedding. The event is assigned to s_{u,j^*} if $r_{j^*} \geq \tau_{\mathrm{match}} = 0.80$. Otherwise, a new skill is created. Matching is purely embedding-based; the LLM is not consulted.

Creating a new skill. For an unmatched event, the skill-construction model writes a short context description c_j and interaction-pattern description ρ_j from the semantic view (prompt in Appendix E). This model is the same backbone as the evaluated agent (Qwen3.7-Plus, DeepSeek-V4-Flash, or GPT-4o-mini), so PIPA never benefits from a stronger auxiliary model. The new skill's counts are initialized to the population prior (Appendix B.2).

Updating a matched skill. For a matched skill, the text (c_j, ρ_j) is immutable: the event identifier is appended to the skill's support list and the text is left unchanged. Its counts change only through mapped behavioral evidence. We impose no cap on the number of skills per user ($J_{\max} = \infty$), and skill merging and eviction are disabled. This avoids introducing a second, hard-to-audit memory-management policy.

B.2 Behavioral Evidence and Population Prior

Evidence mapping. Each observed outcome $y_{u,t}$ is mapped to $z_{u,t} \in \{1, 0, \varnothing\}$ according to Table 8. Positive evidence ($z = 1$) means the user's behavior supported the intervention, negative evidence ($z = 0$) means it contradicted it, and $\varnothing$ marks an uninformative or ambiguous observation. For the matched skill j^* and executed action a,

$$z_{u,t} = 1 : \; n^{+}_{u,j^*,a} \mathrel{+}= 1, \qquad z_{u,t} = 0 : \; n^{-}_{u,j^*,a} \mathrel{+}= 1, \qquad z_{u,t} = \varnothing : \text{ no change.}$$

Not forcing every observation into a label keeps ambiguous behavior out of the belief. Because `silent` produces no user-facing intervention, it yields no post-intervention signal. Its receptivity therefore stays at the prior, and the value of remaining silent enters the decision only through the simulated goal progress (Appendix B.6).

Table 8: Deterministic mapping from observable feedback to receptivity evidence.

Observed behavioral event	**Interpretation**	$z_{u,t}$
Explicit acceptance or confirmation, or direct follow-through on the intervention	behavior supports the intervention	1
Benchmark marks the intervention as warranted and the user proceeds along the proposed path	consistent with the observed goal	1
Explicit rejection, dismissal, correction, undo, or immediate override	behavior contradicts the intervention	0
Benchmark marks assistance as unwanted, or the intervention conflicts with the reference intent	inconsistent with the demonstrated need	0
No post-intervention signal, exogenous termination, missing feedback, or conflicting behavior	no reliable evidence	$\varnothing$

Population prior. PIPA uses a dataset-specific, action-specific empirical-Bayes prior estimated from training users only. Let N^{+}_{a} and N^{-}_{a} be the numbers of positive and negative mapped events for action a among training users. With the Laplace-smoothed rate $\hat{p}_a = (N^{+}_{a}+1)/(N^{+}_{a}+N^{-}_{a}+2)$, every new skill is initialized with

$$\alpha^{(0)}_{a} = 1 + c_0\,\hat{p}_a, \qquad \beta^{(0)}_{a} = 1 + c_0\,(1-\hat{p}_a), \qquad c_0 = 4.$$

The prior has a total pseudo-count of six. It shrinks sparse users toward the population rate while remaining weak enough that a few user-specific observations dominate it. The posterior after n^{+} positive and n^{-} negative observations is $\mathrm{Beta}(\alpha^{(0)}_{a} + n^{+},\ \beta^{(0)}_{a} + n^{-})$. Priors are shared across skills and backbones, and no held-out user contributes to them.

B.3 Retrieval and Contextual Belief Composition

Retrieval. The query is the embedding of the current context string: task or application metadata plus the current observation, excluding any future outcome. Each skill is represented by the concatenation $[c_j; \rho_j]$, and

$$q_j = \cos\big(e(x_{u,t}),\, e([c_j; \rho_j])\big).$$

PIPA keeps the top $K = 3$ skills with $q_j \geq \tau_{\mathrm{ret}} = 0.40$. If no skill passes the threshold, the retrieved set is empty and PIPA falls back to the population prior.

Composition. Retrieved skills are combined with softmax weights

$$w_j = \frac{\exp(q_j/\tau_B)}{\sum_{k \in S^{\mathrm{rel}}_{u,t}} \exp(q_k/\tau_B)}, \qquad \tau_B = 0.10.$$

For posterior sample m, PIPA draws $\theta^{(m)}_{u,j,a} \sim \mathrm{Beta}(\alpha_{u,j,a}, \beta_{u,j,a})$ independently for every retrieved skill and action, and forms

$$\theta^{(m)}_{u,a} = \sum_{j \in S^{\mathrm{rel}}_{u,t}} w_j \, \theta^{(m)}_{u,j,a}.$$

The sampled user state $\Theta^{(m)}_u$ consists of these values together with the retrieved skill text. When the retrieved set is empty, each $\theta^{(m)}_{u,a}$ is drawn from the population prior. Given the Beta draws, the map from skill-level evidence to the contextual state is deterministic.

B.4 Skill-Free Contextual Bayesian Baseline

To isolate the contribution of PIPA's LLM-generated interaction skills from the contribution of Bayesian personalization itself, we introduce **Contextual-Beta**, a skill-free control.

Contextual-Beta uses the same event embeddings and the same behavioral evidence mapping as PIPA, but replaces the semantic skill library with context clusters constructed without an LLM. Context embeddings from training users are partitioned into K_{ctx} clusters by k-means. Cluster centroids $\{\mu_k\}_{k=1}^{K_{\mathrm{ctx}}}$ are frozen before evaluation.

For a held-out interaction context $x_{u,t}$,

$$k^\star = \arg\max_k \cos(e(x_{u,t}), \mu_k). \tag{6}$$

For every user, cluster, and candidate action, Contextual-Beta maintains

$$\theta_{u,k,a} \sim \mathrm{Beta}\left(\alpha^{(0)}_a + n^+_{u,k,a}, \beta^{(0)}_a + n^-_{u,k,a}\right), \tag{7}$$

using exactly the same population prior and feedback mapping as PIPA.

At decision time, the posterior associated with $k^\star$ is passed through the same posterior-sampling, simulation, utility, and uncertainty-sensitive selection pipeline as full PIPA. Thus the only material difference between Contextual-Beta and PIPA is the user representation: embedding-defined coarse context clusters versus LLM-generated recurring interaction skills.

We tune K_{ctx} only on training users from

$$K_{\mathrm{ctx}} \in \{8, 16, 32, 64\},$$

and freeze the selected value for held-out evaluation.

B.5 Shared-Policy Retrieval Control

To separate PIPA's user representation from the effect of its downstream candidate-action evaluation, we additionally evaluate **RAG+Policy**. RAG+Policy retrieves the top-K historical user interactions most similar to the current context using the same embedding model and retrieval budget as PIPA, but does not construct interaction skills or explicit receptivity posteriors.

The retrieved raw interactions and current context are provided to the same candidate-action simulator used by PIPA. The simulator predicts goal progress, interruption cost, and autonomy cost for each candidate action, and utilities are computed by Equation 9. Because RAG+Policy has no posterior distribution, it uses the expected simulated utility without an uncertainty penalty:

$$a^\star_t = \arg\max_{a \in \mathcal{A}} \bar{R}_t(a). \tag{8}$$

This control answers whether PIPA improves over raw-history retrieval because of the interaction-skill and probabilistic user representation rather than merely because its downstream policy evaluates candidate interventions more explicitly.

B.6 Profile-Conditioned Simulation and Decision Utility

Sampling. For each decision, PIPA draws $M = 16$ user states. The same states are reused for all four candidate actions (common random numbers), so differences between actions are not driven by sampling noise. Posterior draws use a deterministic seed derived from (run seed, u, t, m), with run seeds $\{13, 29, 43, 71, 97\}$.

Simulator input and output. For each sample m and action a, the simulator receives four inputs: the current context $x_{u,t}$, the candidate action a, the retrieved skills $S^{\text{rel}}_{u,t}$, and the sampled state $\Theta^{(m)}_u$. It uses the same backbone as the evaluated agent, with temperature 0. It returns a JSON object (Figure 8) with goal progress $g \in [-1, 1]$, interruption cost $c_{\text{int}} \in [0, 1]$, autonomy cost $c_{\text{aut}} \in [0, 1]$, and a one-sentence `likely_response`. For `goal_progress`, -1 means the action moves the user away from the immediate goal, 0 means no expected effect, and 1 means strong progress. The simulator predicts only the *local* response to one candidate action; it does not role-play a complete persona.

Decision utility. The JSON fields are converted to a scalar by a fixed function,

$$R^{(m)}_t(a) = g^{(m)}_t(a) - 0.6\, c^{(m)}_{\text{int},t}(a) - 0.4\, c^{(m)}_{\text{aut},t}(a), \tag{9}$$

with both costs set to zero for `silent`. The coefficients are frozen before held-out evaluation and shared across datasets, backbones, and ablations.

Invalid outputs. Outputs are validated against the schema. An invalid output triggers one retry with a schema-repair message. After a second failure, PIPA uses a conservative fallback that sets $g = 0$ and $c_{\text{int}} = c_{\text{aut}} = 1$ for a non-silent action, and all three fields to zero for `silent`. A parsing failure therefore pushes the decision toward silence rather than toward intervening.

B.7 Uncertainty-Aware Selection

PIPA selects

$$a^\star_{u,t} = \arg\max_{a \in \mathcal{A}} V_t(a), \qquad V_t(a) = \bar{R}_t(a) - \kappa\, \sigma_t(a),$$

where $\bar{R}_t(a)$ and $\sigma_t(a)$ are the mean and standard deviation of $R^{(m)}_t(a)$ over the M samples. The coefficient κ is selected from $\{0, 0.25, 0.5, 0.75, 1.0\}$ by maximizing mean environment utility (Eq. 16) over training users on ProAgentBench. Ties are broken by lower FAR and then by smaller κ. This selects $\kappa = 0.5$, which is then frozen for all datasets and backbones. The same coefficient is applied to every action, so the rule does not impose a commitment ordering by construction (Section 2.4); in practice, an action whose simulated value is unstable across plausible user states is penalized more, and because higher-commitment actions tend to carry larger interruption and autonomy costs, weak evidence tends to favor lower-commitment actions.

Myopic, conservative selection. The rule optimizes one decision at a time and is deliberately risk-averse, in the spirit of mean–variance bandits (Chang, 2026). This matches the deployment setting, where every intervention is visible to the user and an unwanted one has an immediate cost. The price is limited exploration. Receptivity to an action that is rarely selected can stay close to the population prior, because its uncertainty keeps it from being tried. The empirical-Bayes prior mitigates this at cold start, since it is informed by other users' responses to the same action. Optimistic or information-directed variants that trade a bounded amount of interruption for faster learning are a natural extension that we leave to future work.

B.8 Complete Algorithm

Algorithm 1 gives the complete interaction loop. Skill text is created only for unmatched events. Receptivity counts change only through mapped evidence on the matched skill. Selection and all arithmetic happen outside the LLM.

Algorithm 1 PIPA: Profile-Inferred Personalization Adaptation (one decision step)

Require: history $H_{u,t}$; context $x_{u,t}$; skill library $\mathcal{S}_{u,t}$ (empty for a new user); population prior $\{(\alpha_a^{(0)}, \beta_a^{(0)})\}$; actions $\mathcal{A}$; samples M; coefficient κ
Ensure: intervention $a^\star_{u,t}$; updated library $\mathcal{S}_{u,t+1}$

Stage 1: Retrieve context-relevant skills App. B.3
$S^{\text{rel}}_{u,t} \leftarrow \textsc{RetrieveRelevant}(x_{u,t}, \mathcal{S}_{u,t})$ ▷ top-K, $q_j \geq \tau_{\text{ret}}$
$\{w_j\} \leftarrow \text{softmax}_j(q_j/\tau_B)$
Stage 2: Sample plausible user states App. B.3
for $m = 1, \ldots, M$ **do**
  for each $a \in \mathcal{A}$ **do**
    $\theta^{(m)}_{u,a} \leftarrow \sum_j w_j\, \theta^{(m)}_{u,j,a},\ \theta^{(m)}_{u,j,a} \sim \text{Beta}(\alpha_{u,j,a}, \beta_{u,j,a})$ ▷ prior if $S^{\text{rel}}_{u,t} = \emptyset$
  end for
end for
Stage 3: Profile-conditioned simulation App. B.6
for $m = 1, \ldots, M$ **and** each $a \in \mathcal{A}$ **do**
  $(g, c_{\text{int}}, c_{\text{aut}}) \leftarrow \textsc{SimulateResponse}(x_{u,t}, a, \Theta^{(m)}_u, S^{\text{rel}}_{u,t})$
  $R^{(m)}_t(a) \leftarrow g - 0.6\, c_{\text{int}} - 0.4\, c_{\text{aut}}$ ▷ Eq. 9
end for
Stage 4: Uncertainty-aware selection App. B.7
$V_t(a) \leftarrow \bar{R}_t(a) - \kappa\, \sigma_t(a)$ for each $a \in \mathcal{A}$
$a^\star_{u,t} \leftarrow \arg\max_{a \in \mathcal{A}} V_t(a)$
Stage 5: Execute and observe
$\textsc{Execute}(a^\star_{u,t})$; observe feedback $y_{u,t}$
$H_{u,t+1} \leftarrow H_{u,t} \cup \{(x_{u,t}, a^\star_{u,t}, y_{u,t})\}$
Stage 6: Update the profile App. B.1–B.2
$j^* \leftarrow \arg\max_j \cos\big(e(x_{u,t}), e(c_j)\big)$
if $\mathcal{S}_{u,t} = \emptyset$ **or** $\cos\big(e(x_{u,t}), e(c_{j^*})\big) < \tau_{\text{match}}$ **then**
  $(c_{j^*}, \rho_{j^*}) \leftarrow \textsc{BuildSkill}(\text{semantic view of } x_{u,t})$ ▷ only LLM call outside simulation
  $(\alpha_{u,j^*,a}, \beta_{u,j^*,a}) \leftarrow (\alpha_a^{(0)}, \beta_a^{(0)})$ for all $a \in \mathcal{A}$
end if
append event t to the support list of s_{u,j^*} ▷ skill text unchanged
$z_{u,t} \leftarrow \textsc{MapFeedback}(y_{u,t})$ ▷ Table 8
if $z_{u,t} = 1$ **then**
  $\alpha_{u,j^*,a^\star} \leftarrow \alpha_{u,j^*,a^\star} + 1$
else if $z_{u,t} = 0$ **then**
  $\beta_{u,j^*,a^\star} \leftarrow \beta_{u,j^*,a^\star} + 1$
end if
return $a^\star_{u,t}, \mathcal{S}_{u,t+1}$

B.9 Worked End-to-End Example

Figure 5 traces one decision of Algorithm 1 on a small synthetic debugging scenario from our implementation tests. It is illustrative only and is not part of any reported result. The priors follow Appendix B.2 with $\hat{p}_{\texttt{suggest}} = 0.85$ and $\hat{p}_{\texttt{act}} = 0.30$. The user once accepted a suggestion and once declined an offer of help. PIPA is therefore fairly confident about `suggest` and still has only the population prior for `act`, which it has never tried. When the same failure recurs, `suggest` has the best simulated utility, and `act` pays the largest uncertainty penalty. The user accepts the suggestion, and this single observation sharpens the belief.

C Experimental Details

C.1 Benchmarks and Data Splits

ProAgentBench (Tang et al., 2026) is used for proactive timing (904 decisions from 17 users), 14-way intention classification (495 examples from 11 users), personalized suggestion generation, cold-start analysis, and the profile-specificity and uncertainty ablations.

① What happened

Test fails	`silent`	user keeps debugging
Fails again	`ask`	declined ✗
Fails again	`suggest`	fix accepted ✓

② What PIPA learned

Skill s_1**:** *repeated debugging after a test failure; the user edits code in short local cycles.*

`suggest`: Beta(5.4, 1.6), mean 0.77 evidence
`act`: Beta(2.2, 3.8), mean 0.37 prior only

③ Now: the same test fails again

Retrieve s_1 (relevance 0.91), simulate each action, and score it.

	`silent`	`ask`	`suggest`	`act`
$\bar{R}_t(a)$	−0.10	0.29	0.50	0.20
$\sigma_t(a)$	0.04	0.07	0.08	0.16
$V_t(a)$	−0.12	0.26	**0.46**	0.12

④ Act, observe, update

PIPA chooses `suggest`. The user applies the fix and the test passes, so $z_{u,t} = 1$.

$$\text{Beta}(5.4, 1.6) \longrightarrow \text{Beta}(6.4, 1.6)$$

mean 0.77 → 0.80; the skill text is unchanged.

Figure 5: **One PIPA decision, step by step** (synthetic example). ① Past interactions provide behavioral evidence. ② PIPA summarizes them as an interaction skill with a Beta belief per action. ③ For the current context, it simulates each action over $M = 16$ posterior samples and scores it by $V_t(a) = \bar{R}_t(a) - \kappa\,\sigma_t(a)$ with $\kappa = 0.5$. ④ The outcome updates the matched skill's belief.

π-Bench (Zhang et al., 2026) is used for two purposes: profile reconstruction from user-specific histories, and testing whether PIPA improves existing agent harnesses when attached as an external personalization layer (Table 6).

KnowU-Bench (Chen et al., 2026) provides an independent profile-reconstruction setting. Profiles are built from observed history and evaluated on held-out user-specific claims.

Users and tuning. All hyperparameters, the population prior, and the decision threshold are estimated on training users only. Each held-out user is evaluated once with the frozen configuration.

C.2 Implementation Configuration

Table 9 lists the complete configuration. All values were fixed on training users before held-out evaluation and are shared across datasets and backbones unless noted.

C.3 Backbones, Baselines, and Ablations

All methods are run with three backbones: Qwen3.7-Plus, DeepSeek-V4-Flash, and GPT-4o-mini. Within a backbone, all methods share the task observations, action space, and evaluation protocol. They differ *only* in how past interactions are represented and used, and every memory-based method sees only interactions that precede the current decision. Table 10 summarizes the baselines and ablations.

C.4 Sequential Evaluation Protocol

Interactions are processed in chronological order. At step t, a method may use only the user's preceding history, and later interactions are held out. Following PARE (Nathani et al., 2026), the agent observes the current state, selects an intervention, receives behavioral feedback, and updates its user model before the next decision. For cold-start analysis, we vary the amount of history observed before prediction.

Continuous intervention score and calibrated probability. The four-action policy produces one scalar decision value for each candidate action. For PIPA these are the uncertainty-adjusted values

Table 9: Complete PIPA configuration. Validation grids are searched on training users only.

Component	Setting
Skill library	
Skill-construction model	same backbone as the evaluated agent; temperature 0, top-$p = 1$
History processing	chronological replay, one event at a time
Embedding model	`qwen3.7-text-embedding` (1024-d), cosine on ℓ_2-normalized vectors
Matching threshold τ_{match}	0.80 (grid $\{0.70, 0.75, 0.80, 0.85\}$)
Maximum skills per user	no cap; no merging or eviction
Belief	
Population prior	$\alpha_a^{(0)} = 1 + 4\hat{p}_a$, $\beta_a^{(0)} = 1 + 4(1 - \hat{p}_a)$, Laplace-smoothed $\hat{p}_a$
Retrieval	top-$K = 3$, threshold $\tau_{\text{ret}} = 0.40$ (grid $\{0.30, 0.40, 0.50, 0.60\}$)
Composition temperature τ_B	0.10
Simulation and selection	
Simulator	same backbone; temperature 0, top-$p = 1$, max. 192 output tokens
Posterior samples M	16 (sensitivity grid $\{4, 8, 16, 32\}$)
Uncertainty coefficient κ	0.5 (grid $\{0, 0.25, 0.5, 0.75, 1.0\}$)
Decision utility	$g - 0.6\, c_{\text{int}} - 0.4\, c_{\text{aut}}$ (Eq. 9)
Invalid-output fallback	one schema-repair retry, then $(0, 1, 1)$ for non-silent actions
Reproducibility	
Run seeds	$\{13, 29, 43, 71, 97\}$; common random numbers across actions and ablations

Table 10: Baselines and ablations. Ablations change exactly one component of PIPA.

Method	User representation / change
Baselines	
ReAct (Yao et al., 2022)	reasoning–action loop with no persistent user model
RAG (Lewis et al., 2020)	past interactions most similar to the current context
Long-Context (Wu et al., 2024)	all prior interactions in context, truncated to the backbone limit
MemGPT (Packer et al., 2023)	persistent external memory exposed dynamically at inference
Knowledge Graph (Hogan et al., 2021)	entities, attributes, and relations extracted from history; relevant records retrieved per decision
Representation controls	
RAG+Policy	PIPA's candidate-action simulator over retrieved raw history; no skills, no posterior (App. B.5)
Contextual-Beta	PIPA's full Bayesian pipeline over frozen k-means context clusters instead of skills (App. B.4)
Profile-specificity ablations	
Population only	individual profile removed; population prior and no retrieved skills
Shuffled profile	target profile replaced by another user's profile; observation unchanged
Uncertainty ablations	
PIPA-Mean	each posterior replaced by its mean; one simulator call per action
PIPA ($\kappa = 0$)	posterior sampling and simulation kept; selection by $\arg\max_a \bar{R}_t(a)$

$V_t(a)$ from Equation 5. We define the raw intervention margin as

$$s_t = \max_{a \in \{\texttt{ask},\texttt{suggest},\texttt{act}\}} V_t(a) - V_t(\texttt{silent}). \tag{10}$$

Thus $s_t > 0$ means that at least one proactive action has higher decision value than remaining silent, while the magnitude of s_t measures the strength of that preference.

Table 11: **Continuous score used for proactive-timing evaluation.** Every method ultimately produces a scalar non-silent-versus-silent margin s_t, which is then processed by the identical leave-one-user-out calibration protocol in Appendix C.4.

Method	Raw timing score
PIPA	$\max_{a \neq \texttt{silent}} V_t(a) - V_t(\texttt{silent})$ from Equation 5.
PIPA ($\kappa = 0$)	$\max_{a \neq \texttt{silent}} \bar{R}_t(a) - \bar{R}_t(\texttt{silent})$.
PIPA-Mean	Non-silent-versus-silent margin from simulator utilities evaluated at the posterior mean.
Contextual-Beta	Same score as full PIPA, but with the skill representation replaced by the non-LLM contextual Bayesian state described in Appendix B.4.
RAG+Policy	Non-silent-versus-silent utility margin from the shared candidate-action simulator conditioned on retrieved raw user interactions rather than PIPA skills.
RAG / Long-Context / MemGPT / Knowledge Graph / ReAct	Native candidate-action score or action probability emitted before the final argmax. Probabilities are converted to log-odds as defined in Appendix C.4. All methods use the identical downstream calibration procedure.

For each non-PIPA baseline, we expose the corresponding pre-decision scalar score for every candidate action under its native decision policy and construct the same non-silent-versus-silent margin. If a baseline produces normalized action probabilities rather than arbitrary scores, we use

$$s_t = \log \frac{\max_{a \neq \texttt{silent}} p_t(a) + \epsilon}{p_t(\texttt{silent}) + \epsilon}, \qquad \epsilon = 10^{-8}.$$

The exact score extraction for each baseline is listed in Table 11. No ground-truth label or future behavior is used to construct s_t.

Leave-one-user-out calibration. Ranking metrics use the raw score s_t directly. For ECE and thresholded decision metrics, the raw score is converted into a calibrated probability. For each held-out evaluation user u, we fit a two-parameter Platt calibration model using all calibration users except u:

$$p_t = \sigma(\gamma_u s_t + b_u), \tag{11}$$

where $\sigma(\cdot)$ denotes the logistic sigmoid. The parameters (γ_u, b_u) never use interactions from the held-out user.

A binary operating threshold τ_u is then chosen using the same non-held-out calibration users by maximizing macro-F1. Ties are broken first by lower false-alarm rate and then by the threshold closest to 0.5. The held-out prediction is

$$\hat{h}_t = \mathbb{1}[p_t \geq \tau_u]. \tag{12}$$

Accordingly,

$$\text{AUROC/AUPRC} : s_t, \tag{13}$$

$$\text{ECE} : p_t, \tag{14}$$

$$\text{F1/FAR/Accuracy} : \hat{h}_t. \tag{15}$$

This protocol is repeated independently for every held-out user and every method. All reported predictions are therefore out-of-user with respect to both probability calibration and threshold selection.

C.5 Metrics

Profile reconstruction. The fraction of held-out user-specific claims recovered by each reconstructed profile. Cross-user comparisons test whether a user's claims are better recovered by their own profile than by other users' profiles.

User simulation. Profile-conditioned predictions are compared with held-out behavior in the original 1024-d `qwen3.7-text-embedding` space.[1] The UMAP projection in Figure 2 is for display only and does not determine the reported hits.

[1] `https://www.alibabacloud.com/help/en/model-studio/embedding`

Proactive timing. AUROC and AUPRC measure threshold-free ranking. F1, accuracy, false-alarm rate, and expected calibration error (ECE, 10 bins) are computed at the calibrated operating point, with $\mathrm{FAR} = \mathrm{FP}/(\mathrm{FP} + \mathrm{TN})$, the rate of interventions when assistance is not warranted.

Intention and suggestion quality. Macro-F1, macro-precision, and accuracy for 14-way intention classification; SemanticPass@k for free-form suggestions (Appendix C.6).

Environment utility. The uncertainty ablation (Table 4) additionally reports a label-based utility that is independent of the simulator's own scores. Let $h_t \in \{0, 1\}$ indicate whether proactive help is warranted, and let the commitment cost be $c(\texttt{silent}) = 0$, $c(\texttt{ask}) = 0.25$, $c(\texttt{suggest}) = 0.5$, and $c(\texttt{act}) = 1$. Then

$$\begin{aligned} U_t = h_t \Big[\mathbf{1}[a_t \neq \texttt{silent}] \big(1 - 0.2\, c(a_t)\big) - 0.5\, \mathbf{1}[a_t = \texttt{silent}] \Big] \\ - (1 - h_t)\, \mathbf{1}[a_t \neq \texttt{silent}]\, c(a_t). \end{aligned} \tag{16}$$

A warranted intervention earns between 0.8 and 0.95 depending on its commitment, a missed opportunity costs 0.5, and an unwanted intervention is penalized in proportion to its commitment. The same utility is used for every variant, and only for evaluation and validation, never as test-time supervision.

C.6 LLM-as-a-Judge for Personalized Suggestions

Free-form assistance can be appropriate even when it does not lexically match what the user did next, so suggestion quality is judged by Claude Sonnet 5. Direct labels and observable outcomes are scored without an LLM. For each instance, the judge sees the context x, a reference intent y derived from the user's subsequent behavior, and one candidate suggestion $\hat{s}$. It does not see which method produced the suggestion, and the same prompt (Figure 6) is used for all methods and backbones.

Decision rule. The judge returns $J(x, y, \hat{s}) \in \{\textsc{Pass}, \textsc{Fail}\}$. A suggestion passes only if it (1) targets the *same underlying goal* as the reference intent, not merely the same topic or application, and (2) would make *actionable progress* toward that goal, where a necessary sub-step or an equivalent route counts. It fails if it targets a related but different goal, conflicts with the observed behavior, is generic enough to apply to most users in the same context, or hedges across alternatives whose primary recommendation fails (1) or (2). Length, politeness, and hedging are not rewarded.

SemanticPass@k. Each of the top-k suggestions $\hat{s}_{n,i}$ is judged independently, and an example succeeds if any of them passes:

$$\mathrm{SemanticPass@}k = \frac{1}{N} \sum_{n=1}^{N} \mathbf{1}\big[\exists\, i \leq k :\ J(x_n, y_n, \hat{s}_{n,i}) = \textsc{Pass}\big].$$

We report $k = 1$ and $k = 3$.

D Additional Results

D.1 Profiles from Unlabeled Histories

PIPA does not need preference labels to initialize a profile. Unlabeled history determines *which* situations recur, $H_u^{\mathrm{unlabeled}} \mapsto \{(c_j, \rho_j)\}_{j=1}^{J_u}$. Receptivity $(\alpha_{u,j,a}, \beta_{u,j,a})$ is filled in only once later interactions provide behavioral feedback.

We test whether such label-free profiles are already user-specific by scoring each user's held-out claims against their own profile and against another user's profile (Figure 7). With clear signals, the own profile reaches a recall of 0.556, versus 0.208 for another user's profile ($p = 1.3 \times 10^{-9}$). With vague signals, the values are 0.451 versus 0.165 ($p = 5.8 \times 10^{-11}$). The own profile therefore reaches $2.8\times$ and $2.3\times$ chance (0.20), while another user's profile is indistinguishable from chance. User-specific structure is thus recoverable before any receptivity evidence is available.

You are evaluating a proactive suggestion made by an AI assistant. Your task is to decide whether the suggestion targets the goal the user actually pursued.

Inputs.
Context: {`context`}
Reference intent (inferred from what the user did next): {`reference`}
Candidate suggestion: {`suggestion`}

Decision rule. Output PASS only if *both* hold:

(1) *Same goal.* The suggestion addresses the same underlying goal as the reference intent, not merely the same topic or application.
(2) *Actionable progress.* Acting on the suggestion would materially advance that goal in the given context. A necessary sub-step or an equivalent alternative route counts; the exact wording or action sequence need not match.

Otherwise output FAIL. In particular, output FAIL if the suggestion:

– targets a different or merely related goal;
– conflicts with the reference intent or the user's observed behavior;
– is generic, i.e., it would apply equally to most users in this context and does not specify what to do;
– offers several alternatives whose primary recommendation does not satisfy (1) and (2).

Guidelines. Judge only from the information given; do not assume user goals beyond the reference intent. Do not reward length, politeness, or hedging. If the reference intent is ambiguous, accept any suggestion consistent with a reasonable reading of it.

Respond with exactly one word: PASS or FAIL.

Figure 6: Verbatim instruction used by the LLM judge for all methods and backbones.

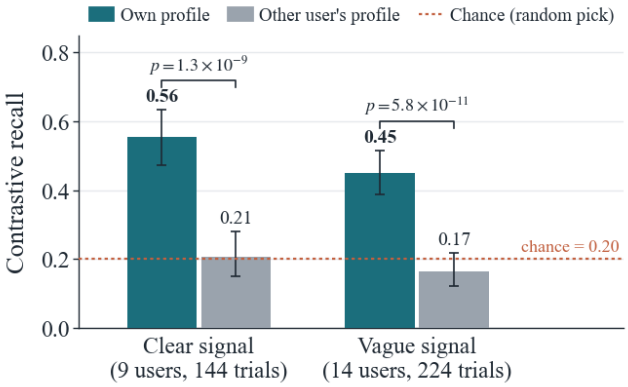


Figure 7: **Profiles built without labels are user-specific.** Contrastive recall when a user's claims are scored against their own reconstructed profile versus another user's. Another user's profile is indistinguishable from chance (p=0.80 and p=0.19). Error bars show 95% Wilson intervals over trials.

D.2 Belief Concentration and Calibration

Table 12 tests two properties the method relies on. First, the posterior should concentrate as evidence accumulates. Its variance decreases monotonically with history size on all three backbones. Second, the explicit belief should be better calibrated than the same information expressed in prose. In all nine backbone–history settings, the Beta–Bernoulli belief has the lowest ECE. This holds against the probability that the same LLM infers from the same profile (*prose channel*) and against every memory baseline. The gap to the prose channel isolates the value of keeping receptivity as explicit counts rather than text: the information is identical, and only the representation differs.

Table 12: **Belief concentration and calibration as user evidence accumulates.** Cold-start decisions are grouped by the number of history events observed before the decision ($n = 281/268/51$). *Top:* posterior variance of PIPA's Beta–Bernoulli belief. *Bottom:* ECE (10 bins) of the probability used for intervention decisions; *prose channel* is the probability inferred by the LLM from the same profile. Bold marks the lowest ECE per column. The $64+$ bucket is small and therefore noisier.

	Qwen3.7-Plus			DeepSeek-V4-Flash			GPT-4o-mini		
History size	0–32	32–64	64+	0–32	32–64	64+	0–32	32–64	64+
Posterior variance $\mathrm{Var}(\theta \mid H) \downarrow$									
PIPA belief	0.0098	0.0076	0.0031	0.0074	0.0036	0.0025	0.0109	0.0082	0.0043
Calibration error (ECE) $\downarrow$									
PIPA belief	**0.084**	**0.061**	**0.049**	**0.072**	**0.053**	**0.047**	**0.096**	**0.071**	**0.071**
PIPA (prose channel)	0.135	0.095	0.062	0.180	0.140	0.100	0.145	0.110	0.080
Knowledge Graph	0.175	0.085	0.092	0.330	0.310	0.290	0.190	0.100	0.110
RAG	0.090	0.100	0.108	0.359	0.340	0.320	0.156	0.075	0.090
Long-Context	0.119	0.125	0.130	0.359	0.345	0.330	0.195	0.105	0.115
MemGPT	0.296	0.275	0.255	0.366	0.355	0.340	0.270	0.265	0.260
ReAct	0.297	0.290	0.285	0.381	0.370	0.360	0.297	0.300	0.305

E Prompts and Output Schemas

E.1 Skill Construction

The skill-construction model receives the semantic view of one unmatched event (Appendix B.1) and returns two short strings: a context description c_j and an interaction-pattern description ρ_j. It is not shown outcomes, receptivity labels, or counts, and it is instructed not to state preferences.

E.2 Profile-Conditioned Simulator

Figure 8 gives the output schema. Values outside the stated ranges, or missing fields, count as invalid outputs (Appendix B.6).

```
{
  "goal_progress":  <float in [-1, 1]>,
  "interruption_cost":  <float in [0, 1]>,
  "autonomy_cost":  <float in [0, 1]>,
  "likely_response": <one sentence>
}
```

Figure 8: Output schema of the profile-conditioned simulator. The scalar utility (Eq. 9) is computed outside the LLM.