WHAT TO ASK, WHEN TO ROUTE: VALUE-GUIDED SKILL ROUTING UNDER JOINT TASK–PROFILE UNCERTAINTY

Anonymous authors
Paper under double-blind review

ABSTRACT

Skill routing for LLM-based agents is typically performed from a single observed request, implicitly assuming that the available information is sufficient to identify the appropriate skill. In practice, the intended task, relevant user profile conditions, or both may be underspecified, making one-shot routing unreliable. We formulate personalized skill routing under joint task–profile uncertainty as a sequential clarify-or-route problem and propose VISTA, a training-free framework for resolving such decision-relevant uncertainty. VISTA identifies candidate-distinguishing task and profile factors, constructs a compact candidate-conditioned joint belief over plausible routing states, and evaluates task clarification, profile clarification, and immediate routing under a shared downstream utility. Based on this belief, value-of-information criteria determine whether clarification is worthwhile and which question offers the greatest expected routing benefit. We further construct a controlled counterfactual benchmark seeded by real user queries and profiles, comprising 213 cases and 2,160 simulated interactive episodes across task-only, profile-only, and joint ambiguity. The benchmark also includes explicit clarification-stopping controls and independent human validation. Experiments on two base LLMs show that VISTA improves cost-aware routing performance by 13.8% on average over interactive baselines while reducing clarification turns by 40.1%.

1 INTRODUCTION

Skills are reusable, task-oriented packages of instructions, procedures, and optional supporting resources for LLM-based agents (OpenAI, 2026a). As skill libraries grow in scale, selecting the appropriate skill for a given request becomes more challenging, increasing the importance of explicit skill routing (Wang et al., 2026b; Ding et al., 2026). Existing skill-routing methods include task-based matching (Wang et al., 2026b; Chen et al., 2026; Zheng et al., 2026; Kang et al., 2026) and personalized routing that jointly considers the task and user profile (Wang et al., 2026a). Despite these advances, a common limitation is that existing routers generally assume that the routing-relevant information available at inference time is sufficient for one-shot selection.

In practice, this assumption often fails. As illustrated in Figure 1(a), a request may be underspecified with respect to the intended task, relevant profile conditions, or both. This phenomenon is prevalent in real usage: among 995 real-user queries collected from the SkillFeed platform (SkillFeed, 2026), 87.1% exhibit routing-relevant ambiguity (Figure 1(b)). Such underspecification allows the same observed request to correspond to multiple plausible routing states that favor different skills, making one-shot routing unreliable. This motivates efforts to resolve such uncertainty before routing.

A natural response to underspecification is to ask clarifying questions. Recent work has explored clarification based on uncertainty-, similarity-, and value-based criteria (Dong et al., 2026; Kobalczyk et al., 2025; Zhang & Choi, 2025; Suri et al., 2026; Testoni & Fernández, 2024). Yet uncertainty alone does not justify clarification. If different plausible task–profile states lead to the same preferred skill, resolving that uncertainty does not improve the routing decision. Conversely, even a modest reduction in uncertainty may be valuable if it changes the preferred skill. Moreover, every clarification imposes interaction cost on the user. Therefore, the central problem is not uncertainty reduction itself, but identifying and resolving decision-relevant uncertainty under a limited interaction budget.

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.



(a) Three Sources of Routing Uncertainty (b) Real Website Query Statistics

Figure 1: **Real-World Skill Requests Are Often Underspecified. (a)** Three sources of routing uncertainty: task ambiguity, profile ambiguity, and joint ambiguity. Missing task or profile information can lead to different preferred skills and unreliable one-shot routing. **(b)** Statistics from real SkillFeed queries showing that most requests contain at least one form of routing-relevant ambiguity.

From a value-of-information perspective (Howard, 1966; Raiffa & Schlaifer, 1961), clarification is worthwhile only when its expected improvement to the downstream routing decision justifies its cost.

This challenge is particularly pronounced in personalized skill routing because task and profile factors interact in determining the preferred route. Before clarification, multiple plausible task–profile states may remain consistent with the current observation, motivating a joint representation of routing uncertainty. However, explicitly enumerating all task–profile combinations leads to a combinatorial explosion. For routing, it is sufficient to represent only the distinctions that separate the current candidate skills, rather than representing every possible user state. This motivates a compact, candidate-conditioned representation of joint task–profile uncertainty.

Based on these observations, we formulate personalized skill routing under joint task–profile uncertainty as a finite-horizon clarify-or-route problem and propose VISTA, a training-free framework for value-guided interactive skill routing under task–profile ambiguity. We also construct a controlled benchmark with 213 cases and 2,160 interactive episodes, and experiments on two base LLMs show improved cost-aware routing with fewer unnecessary clarifications. Our contributions are threefold:

- We formulate personalized skill routing under joint task–profile uncertainty as a finite-horizon sequential decision problem. Based on this joint uncertainty representation, we cast clarification as a value-guided allocation of limited interaction resources across task clarification, profile clarification, and immediate routing, enabling the router to resolve decision-relevant uncertainty while avoiding unnecessary interaction.
- We propose VISTA, a training-free value-guided routing framework that constructs a compact candidate-conditioned joint belief to control combinatorial growth of task–profile hypothesis space while preserving routing-relevant distinctions. Clarification is valued by its expected impact on downstream routing rather than uncertainty reduction alone. Expected value of perfect information provides a sufficient bound for stopping and pruning, while cost-aware expected value of sample information selects the most valuable clarification.
- We construct a controlled benchmark for evaluating route-changing ambiguity and clarification stopping across profile-only, task-only, and joint counterfactual states. The benchmark includes controlled observation views and explicit clarification-stopping controls and is independently audited by two human reviewers. Experiments across two base LLMs demonstrate stronger cost-aware routing with fewer unnecessary clarification turns.

2 Related Work

Skill Libraries and Routing. Public skill libraries include SkillNet (Liang et al., 2026) and collections maintained by Anthropic (Anthropics, 2026), OpenAI (OpenAI, 2026b), and Vercel (Vercel-labs, 2026). Recent work studies large-scale skill retrieval and routing from task requests (Liang et al.,

2026; Zheng et al., 2026; Kang et al., 2026; Wang et al., 2026b; Chen et al., 2026) and profile-aware requests (Wang et al., 2026a). Our work instead studies whether missing task or profile information is worth acquiring before routing, jointly considering clarification value and interaction cost.

Personalization and Preference Elicitation. Personalized tool use and skill selection incorporate interaction histories or user profiles (Xu et al., 2025; Hao et al., 2025; Wang et al., 2026a). Related work elicits preferences through sequential questions (Montazeralghaem et al., 2025) or clarifies both information needs and user profiles (Feng et al., 2023). Building on these settings, VISTA allocates costly clarification between task and profile uncertainty, comparing both question types with immediate routing under a shared utility.

Clarification and Value of Information. Prior work studies clarification and information acquisition with expected answer usefulness (Rao & Daumé III, 2018), information gain (Kobalczyk et al., 2025; Deng et al., 2026), and intent uncertainty (Zhang & Choi, 2025). Building on classical information valuation (Raiffa & Schlaifer, 1961; Howard, 1966), recent frameworks balance clarification benefits against interaction costs (Suri et al., 2026; Dong et al., 2026). Our work specializes this decision principle to personalized skill routing, using candidate-conditioned joint task–profile beliefs and value-guided decisions over what to clarify and when to stop.

Skill Evaluation Benchmarks. Existing benchmarks assess skill retrieval and compatibility (Kang et al., 2026; Wang et al., 2026b), risk exposure (Ding et al., 2026), profile sensitivity (Wang et al., 2026a), and skill incorporation and downstream effectiveness (Su et al., 2026; Li et al., 2026a). Our benchmark focuses on clarification allocation and stopping for skill selection, combining task-only, profile-only, and joint counterfactuals with controlled observations and explicit route-invariant and task-sufficient controls.

3 Problem Formulation

Based on the decision-theoretic view of adaptive clarification in Dong et al. (2026); Suri et al. (2026), interactive skill routing is formulated as a finite-horizon sequential decision process. At each turn, the agent either asks a targeted question to refine its belief over the user's task and profile, or terminates the interaction by routing to the skill with the highest expected utility under the current belief.

Interaction model. For each request, the latent routing state $z^\star = (t^\star, p^\star) \in \mathcal{T} \times \mathcal{P}$ consists of the intended task and routing-relevant profile conditions and remains fixed throughout the episode. The agent observes a query x and profile memory m, and at turn i maintains a belief $b_i(z) = \Pr(Z = z \mid x, m, h_i)$ conditioned on the clarification history h_i. Before interaction begins, an upstream retriever constructs a fixed candidate set $\mathcal{S} = \{s_j\}_{j=1}^{N}$, where each skill is represented by its name, description, and `SKILL.md` content. At each turn, the agent either asks a task-side question $q \in \mathcal{Q}_{\mathrm{T},i}$, asks a profile-side question $q \in \mathcal{Q}_{\mathrm{P},i}$, or routes to a skill $s \in \mathcal{S}$ and terminates. Each answer updates the interaction history and belief, and the agent may ask at most B clarification questions. Full definitions of the state, observations, and actions are provided in Appendix A.1.

Terminal routing utility. Given a latent routing state $z = (t, p)$, the terminal utility of routing to the jth candidate skill s_j is defined by its feasibility and compatibility with both the intended task and the applicable profile conditions:

$$U(s_j, z) = F(s_j, z)\left[\alpha R_T(s_j, t) + (1-\alpha) R_P(s_j, p)\right], \qquad \alpha \in [0,1], \tag{1}$$

where $F(s_j, z) \in \{0, 1\}$ indicates whether skill s_j is feasible for the requirements specified by latent state z, $R_T(s_j, t)$ measures the compatibility between the skill s_j and the intended task t, while $R_P(s_j, p)$ measures the compatibility between the skill s_j and the profile conditions p, and α is a parameter controlling their relative contribution.

Communication cost and objective. Clarification can improve the routing decision, but each question incurs a communication cost that reflects the additional interaction burden on the user. If the agent routes at turn τ after asking $q_0, \ldots, q_{\tau-1}$, the cumulative communication cost is $C_{\mathrm{comm}}(h_\tau) = \sum_{i=0}^{\tau-1} c(q_i)$, where $c(q_i) \geq 0$ is the cost of asking question q_i. Let Π_B denote the set of admissible policies that ask at most B clarification questions before committing to a skill. The objective is to maximize the terminal routing utility after accounting for the cumulative communication cost:

$$\pi^\star \in \arg\max_{\pi \in \Pi_B} \mathbb{E}_\pi[U(s_\tau, Z) - C_{\mathrm{comm}}(h_\tau) \mid x, m], \tag{2}$$

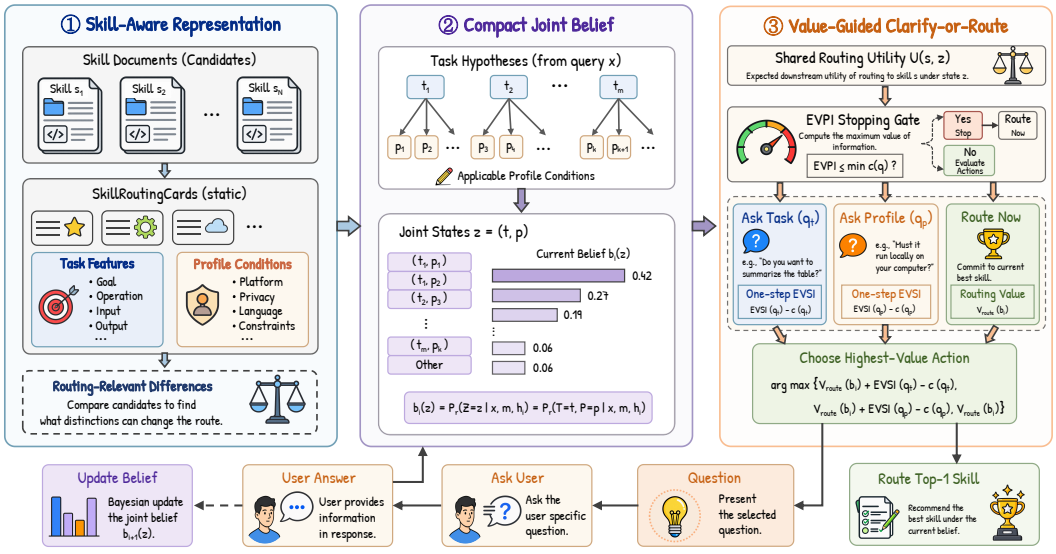


Figure 2: **Overview of VISTA. (1) Skill-aware representation** summarizes candidate skills with static *SkillRoutingCards* and identifies routing-relevant task and profile distinctions. **(2) Compact joint belief construction** organizes these distinctions into a compact support over plausible task–profile states. **(3) Value-guided clarify-or-route** compares clarification and routing under a shared utility, updating the joint belief after each user response.

where $Z = (T, P)$ is the latent joint routing state, τ is the stopping time induced by policy π, and $s_\tau \in \mathcal{S}$ is the skill selected at termination.

4 Value-Guided Interactive Skill Routing

As illustrated in Figure 2, VISTA adopts a candidate-conditioned view of decision-relevant uncertainty. Given the retrieved candidates, it first identifies task- and profile-side distinctions that can change their relative preference, and uses these distinctions to construct a compact joint support over plausible task–profile states. This support defines the belief space on which task clarification, profile clarification, and immediate routing are evaluated under a shared downstream routing utility. To approximate the finite-horizon objective, VISTA uses one-step expected value of sample information (EVSI) to select clarification according to its expected decision value, while expected value of perfect information (EVPI) provides a stopping and pruning bound when further clarification cannot justify its cost. After each answer, the belief and decision are recomputed.

Skill-Aware State Construction. An offline extraction step maps each candidate's name, description, and `SKILL.md` content to a static *SkillRoutingCard* r_j, summarizing its task scope, applicable profile conditions, and execution- or risk-related properties. Given the cards, we identify the candidate-conditioned contrast set $\mathcal{D}(\mathcal{S})$ containing task- and profile-side dimensions that vary across candidates. Task-side dimensions include goals, operations, inputs, outputs, and non-goals. Profile-side dimensions include platform, privacy, language, and execution constraints. These differences provide a compact contrastive representation for constructing routing-relevant hypotheses. Formal definitions are provided in Appendix A.2.

4.1 Compact Joint Belief Construction

Naively enumerating plausible task–profile combinations leads to combinatorial growth in the joint hypothesis space, much of which is decision-irrelevant for the candidate set. We therefore construct a compact, candidate-conditioned support in a task-first manner. A bounded set of task hypotheses is retained first, and only the profile conditions applicable to each task are subsequently instantiated.

Task-First Joint Support. Conditioned on the query x, profile memory m, candidate cards, and $\mathcal{D}(\mathcal{S})$, we construct plausible task hypotheses $\mathcal{T}_0$ and applicable profile hypotheses $\mathcal{P}_0(t)$ for each task. The construction functions are defined in Appendix A.3. The resulting joint support is

$$\mathcal{H}_0 = \{z_{\text{obs}}\} \cup \bigcup_{t \in \mathcal{T}_0} \{(t, p) \mid p \in \mathcal{P}_0(t)\}, \qquad |\mathcal{H}_0| \le L_0, \tag{3}$$

where z_{obs} is the anchor supported by the observed task and profile evidence in x and m, and L_0 bounds the support size. We provide further discussion of the compactness and dependence properties of this task-first construction in Appendix A.4.

Joint Belief Initialization. Given an explicit support $\mathcal{H}_0^{(L)}$ with at most L hypotheses, we initialize an augmented belief over the retained states and a residual state z_{other}. Its estimated probability mass $\rho_0^{(L)}$ captures states outside the explicit support. For routing and clarification, we condition the belief on the retained support. Full initialization and normalization details are provided in Appendix A.5.

Adaptive Support Expansion. A fixed support size may allocate unnecessary capacity to well-specified requests while failing to cover sufficiently ambiguous ones. VISTA therefore adapts L according to the residual mass $\rho_0^{(L)}$. When $\rho_0^{(L)} > \epsilon_{\text{cov}}$, it introduces additional hypotheses from candidate-distinguishing dimensions not yet covered by the current support, where $\epsilon_{\text{cov}} \in [0, 1]$ is the maximum residual probability mass tolerated outside the explicit support. Expansion terminates once $\rho_0^{(L)} \le \epsilon_{\text{cov}}$ or the maximum support size $L_{\max}$ is reached. This allows the belief support to grow adaptively with the unresolved ambiguity of each request.

4.2 Value-Guided Clarify-or-Route

Given the finalized $\mathcal{H}_0$ and estimated joint belief b_i, VISTA chooses among three types of actions: task-side clarification, profile-side clarification, and immediate routing. All three are evaluated against the same routing utility in Eq. 1.

Shared Routing Utility. For each joint state $z \in \mathcal{H}_0$ and candidate skill $s_j \in \mathcal{S}$, we evaluate the routing utility $\widehat{U}(s_j, z)$. Under the current belief b_i, the value of immediate routing V_{route} and the corresponding skill s_i^{route} are

$$V_{\text{route}}(b_i) = \max_{s_j \in \mathcal{S}} \mathbb{E}_{z \sim b_i}\left[\widehat{U}(s_j, z)\right], \qquad s_i^{\text{route}} \in \arg\max_{s_j \in \mathcal{S}} \mathbb{E}_{z \sim b_i}\left[\widehat{U}(s_j, z)\right]. \tag{4}$$

This routing value serves as the common baseline against which clarification actions are evaluated.

EVPI Stopping Gate. At turn i, the candidate clarification set is $\mathcal{Q}_i = \mathcal{Q}_{\text{T},i} \cup \mathcal{Q}_{\text{P},i}$, where each question $q \in \mathcal{Q}_i$ has a finite answer space $\mathcal{Y}_q$ and communication cost $c(q)$. Evaluating every question requires considering its possible answers and the resulting posterior beliefs. To avoid unnecessary question-specific evaluations, we compute the expected value of perfect information (EVPI) under the current belief b_i as a global pruning criterion:

$$\text{EVPI}(b_i) = \mathbb{E}_{z \sim b_i}\left[\max_{s_j \in \mathcal{S}} \widehat{U}(s_j, z)\right] - V_{\text{route}}(b_i), \tag{5}$$

where $\text{EVPI}(b_i)$ upper-bounds the gross value of any clarification under the retained belief and shared routing utility (proven in Appendix B.1). For a nonempty question set $\mathcal{Q}_i$, let $c_{\min}(\mathcal{Q}_i) = \min_{q \in \mathcal{Q}_i} c(q)$. Whenever $\text{EVPI}(b_i) \le c_{\min}(\mathcal{Q}_i)$, all candidate questions have non-positive net value, so clarification is pruned and the router selects s_i^{route}.

Question-Specific EVSI Evaluation. If the EVPI gate remains open, VISTA evaluates each task- and profile-side question individually. For a candidate question $q \in \mathcal{Q}_i$ and a possible response $y \in \mathcal{Y}_q$, the predictive response probability and the corresponding hypothetical posterior are

$$p(y \mid q, b_i) = \sum_{z \in \mathcal{H}_0} p(y \mid z, q, h_i) b_i(z), \qquad b_i^{q,y}(z) = \frac{p(y \mid z, q, h_i) b_i(z)}{\sum_{z' \in \mathcal{H}_0} p(y \mid z', q, h_i) b_i(z')}. \tag{6}$$

Based on these hypothetical posteriors, the one-step expected value of sample information (EVSI) is

$$\text{EVSI}(q \mid b_i) = \sum_{y \in \mathcal{Y}_q} p(y \mid q, b_i) V_{\text{route}}(b_i^{q,y}) - V_{\text{route}}(b_i). \tag{7}$$

We then define the net value of clarification as $\mathrm{NetVOI}(q \mid b_i) = \mathrm{EVSI}(q \mid b_i) - c(q)$, which accounts for the communication cost of asking q.

Clarify-or-Route Decision. The router selects the highest-value clarification across both task- and profile-side questions:

$$q_i^\star \in \arg \max_{q \in \mathcal{Q}_{\mathrm{T},i} \cup \mathcal{Q}_{\mathrm{P},i}} \mathrm{NetVOI}(q \mid b_i). \tag{8}$$

If $\mathrm{NetVOI}(q_i^\star \mid b_i) > 0$, the router asks $q_i^\star$; otherwise, it commits to the current best skill s_i^{route}.

Belief Update and Iteration. When clarification is selected, the router asks $q_i^\star$ and observes the user response y_i. The interaction history and joint belief are then updated as

$$h_{i+1} = (h_i, q_i^\star, y_i), \qquad b_{i+1}(z) = b_i^{q_i^\star, y_i}(z). \tag{9}$$

The updated belief b_{i+1} is used in the next decision round, where task-side clarification, profile-side clarification, and immediate routing are evaluated again. The process terminates when routing is selected or the clarification budget is exhausted, returning the current s_i^{route}.

The complete online inference procedure and implementation details are provided in Appendix B.

5 Benchmark Construction

We construct a controlled benchmark from real skill-routing queries and user profiles collected from SkillFeed (SkillFeed, 2026), with an independent pool of $228{,}432$ skill documents. The benchmark is designed to disentangle variation in the underlying task–profile state from variation in the information revealed to the router. To achieve this, we construct controlled counterfactual state families, keep the candidate pool fixed within each family, independently measure task–skill and profile–skill compatibility, and deterministically materialize reference routing labels. Track-specific criteria then retain valid ambiguity cases, from which controlled observation views and clarification-stopping controls are constructed. This design distinguishes route-changing uncertainty from residual uncertainty that can remain unresolved, enabling controlled evaluation of terminal routing, clarification direction, and stopping behavior. The full construction pipeline is summarized in Appendix Figure 4, with additional details provided in Appendix C.

Controlled States and Observation Views. We construct three types of counterfactual families. Profile-only cases vary the profile while keeping the task fixed, task-only cases vary the task under a fixed profile, and joint cases vary both dimensions. All states in a family share the same candidate pool, fixed before observation masking. After reference routes are materialized, track-specific filtering retains only families with the intended routing structure. For joint cases, each latent state is paired with four observation views: fully revealed, joint hidden, profile hidden, and task hidden. Route-invariant controls test whether the router avoids unnecessary clarification, while task-sufficient controls test whether it stops once task information is sufficient to determine the route.

Compatibility and Reference Labels. Following LLM-as-a-judge evaluation (Zheng et al., 2023; Liu et al., 2023; Wang et al., 2026b; Hossain et al., 2026), we measure skill compatibility with a fixed evaluator $U^\dagger$, together with hard-violation judgments. The task-side evaluator observes only the task, whereas the profile-side evaluator observes only the profile. These measurements are then converted by fixed rules into the acceptable skill set $\mathcal{A}(T, P)$ and the reference route $s^\star(T, P) = \arg\max_{s \in \mathcal{A}(T,P)} U^\dagger(s, (T, P))$. A latent state is eligible for a route-changing ambiguity case only when its acceptable set is nonempty and its reference route is unique. Track-specific criteria then determine whether the corresponding profile-only, task-only, or joint family is retained.

Composition, Split, and Human Audit. The benchmark contains 138 ambiguity cases and 75 controls, yielding 2,160 episodes. The development and test splits contain 704 and 1,456 episodes, respectively, with all variants from each counterfactual family kept in the same split. Evaluator-only information, including latent states, hidden answers, acceptable sets, and reference routes, is withheld from the router. Two independent human reviewers audit 150 stratified episodes. For preferred-route, candidate-acceptability, and pairwise judgments, agreement with the benchmark ranges from 80.0% to 88.0%, while inter-reviewer Cohen's κ (Cohen, 1960) ranges from 0.624 to 0.823.

Table 1: Main results on the full test split across different base LLMs. PRA, ARR, and FAA are reported in percentages (%), while NR and GU are normalized scores. Avg. Q and Token denote the average numbers of clarification questions and tokens (in thousands) per episode, respectively. Best and second-best results according to the indicated direction are shown in **bold** and underlined.

Method	NR↑	PRA↑ (%)	ARR↑ (%)	FAA↑ (%)	GU↑	Avg. Q↓ (Q/ep.)	Token↓ (k/ep.)
Base LLM: DeepSeek-v4-Flash							
Retriever Top-1	0.257 ± 0.000	20.19 ± 0.00	38.74 ± 0.00	39.84 ± 0.00	0.257 ± 0.000	0.00 ± 0.00	0.00 ± 0.00
Direct Router	0.374 ± 0.006	36.41 ± 0.91	54.27 ± 0.83	39.84 ± 0.00	0.374 ± 0.006	0.00 ± 0.00	2.43 ± 0.16
Fixed-Round Clarification	0.362 ± 0.006	47.86 ± 0.97	67.18 ± 0.86	46.57 ± 1.02	0.462 ± 0.006	2.00 ± 0.00	33.86 ± 2.83
Adaptive LLM Prompting	0.426 ± 0.007	47.21 ± 0.87	66.52 ± 0.81	50.85 ± 1.04	0.458 ± 0.006	0.64 ± 0.05	4.13 ± 0.35
Random-Ask@Matched	0.409 ± 0.007	44.52 ± 0.93	62.86 ± 0.88	44.73 ± 1.17	0.436 ± 0.007	0.54 ± 0.05	4.82 ± 0.42
Confidence Thresholding	0.434 ± 0.006	48.74 ± 0.84	67.93 ± 0.78	53.48 ± 0.95	0.470 ± 0.006	0.71 ± 0.06	11.63 ± 0.91
Intent-Sim Clarification	0.433 ± 0.006	49.18 ± 0.86	68.37 ± 0.79	54.63 ± 0.93	0.472 ± 0.006	0.78 ± 0.06	10.74 ± 0.86
SAGE	0.441 ± 0.006	50.43 ± 0.81	69.77 ± 0.73	56.28 ± 0.89	0.484 ± 0.005	0.85 ± 0.07	14.27 ± 1.08
Generic VoI Clarification	0.450 ± 0.006	51.18 ± 0.78	70.48 ± 0.71	57.01 ± 0.87	0.491 ± 0.005	0.81 ± 0.07	18.64 ± 1.47
VISTA-ER (Our variant)	$\underline{0.465 \pm 0.007}$	$\mathbf{53.12 \pm 0.94}$	$\mathbf{73.04 \pm 0.79}$	$\underline{59.56 \pm 1.07}$	$\mathbf{0.514 \pm 0.006}$	0.99 ± 0.08	15.83 ± 1.24
VISTA (Ours)	$\mathbf{0.481 \pm 0.005}$	$\underline{52.62 \pm 0.61}$	$\underline{72.57 \pm 0.51}$	$\mathbf{62.68 \pm 0.76}$	$\underline{0.508 \pm 0.004}$	0.54 ± 0.04	9.83 ± 0.73
Base LLM: Mimo-v2.5							
Retriever Top-1	0.257 ± 0.000	20.19 ± 0.00	38.74 ± 0.00	39.84 ± 0.00	0.257 ± 0.000	0.00 ± 0.00	0.00 ± 0.00
Direct Router	0.348 ± 0.007	33.16 ± 1.03	50.82 ± 0.94	39.84 ± 0.00	0.348 ± 0.007	0.00 ± 0.00	2.57 ± 0.19
Fixed-Round Clarification	0.335 ± 0.007	44.29 ± 1.07	63.12 ± 0.95	43.53 ± 1.12	0.435 ± 0.007	2.00 ± 0.00	73.13 ± 5.81
Adaptive LLM Prompting	0.397 ± 0.008	43.82 ± 1.01	62.47 ± 0.91	47.32 ± 1.14	0.432 ± 0.007	0.68 ± 0.06	4.53 ± 0.43
Random-Ask@Matched	0.379 ± 0.008	41.72 ± 1.06	59.84 ± 0.98	42.36 ± 1.23	0.407 ± 0.007	0.57 ± 0.05	5.26 ± 0.48
Confidence Thresholding	0.410 ± 0.007	45.63 ± 0.95	65.18 ± 0.87	50.16 ± 1.04	0.447 ± 0.006	0.75 ± 0.07	14.83 ± 1.14
Intent-Sim Clarification	0.407 ± 0.007	46.21 ± 0.96	65.93 ± 0.88	50.84 ± 1.02	0.449 ± 0.006	0.82 ± 0.07	13.17 ± 1.02
SAGE	0.417 ± 0.007	47.36 ± 0.91	67.28 ± 0.82	52.61 ± 0.98	0.461 ± 0.006	0.89 ± 0.08	16.84 ± 1.31
Generic VoI Clarification	0.422 ± 0.007	48.04 ± 0.88	68.23 ± 0.81	54.16 ± 0.96	0.468 ± 0.006	0.94 ± 0.08	22.71 ± 1.84
VISTA-ER (Our variant)	$\underline{0.439 \pm 0.008}$	$\mathbf{50.56 \pm 1.09}$	$\mathbf{70.63 \pm 0.91}$	$\underline{56.83 \pm 1.15}$	$\mathbf{0.490 \pm 0.007}$	1.03 ± 0.09	19.43 ± 1.61
VISTA (Ours)	$\mathbf{0.449 \pm 0.006}$	$\underline{48.83 \pm 0.71}$	$\underline{69.16 \pm 0.59}$	$\mathbf{59.15 \pm 0.84}$	$\underline{0.477 \pm 0.005}$	0.57 ± 0.05	12.33 ± 0.86

6 EXPERIMENTS

6.1 EXPERIMENTAL SETUP

Evaluation Protocol. We evaluate all methods on the profile-only, task-only, and joint tracks. For each episode, methods receive the same query, visible profile memory, candidate SkillRoutingCards, candidate order, and clarification budget, while latent states and reference routes remain evaluator-only. Results are averaged over five independent runs for $N \in \{3, 5, 10\}$. We consider two base LLMs, `deepseek-v4-flash` (DeepSeek-AI et al., 2026) and `mimo-v2.5` (mim, 2026).

Baselines. We compare VISTA against a range of routing and clarification methods. *Two non-interactive routing baselines:* **(1) Retriever Top-1**, which directly selects the top-ranked retrieved skill without clarification, and **(2) Direct Router**, which performs one-shot LLM routing from the observed request. *Three general interaction baselines:* **(3) Fixed-Round Clarification**, which asks a fixed number of clarification questions, **(4) Adaptive LLM Prompting**, which lets the LLM implicitly decide whether and what to ask, and **(5) Random-Ask@Matched** (Testoni & Fernández, 2024), which randomly selects clarification questions while matching the interaction budget. *Three uncertainty-aware clarification baselines:* **(6) Confidence Thresholding** (Geifman & El-Yaniv, 2019; Gu & Hopkins, 2023), **(7) Intent-Sim Clarification** (Zhang & Choi, 2025), and **(8) adapted SAGE** (Suri et al., 2026), which use confidence-, similarity-, or uncertainty-based signals to determine clarification. *One generic value-aware baseline:* **(9) Generic VoI Clarification** (Dong et al., 2026), which applies value-of-information-based clarification without the candidate-conditioned joint task–profile state representation of VISTA. *One controlled variant:* **(10) VISTA-ER**, which retains the joint state representation and question space of VISTA but replaces value-guided selection and stopping with entropy reduction (Settles, 2009; Yu et al., 2010).

Metrics. We evaluate routing quality using Preferred Route Accuracy (PRA), Acceptable Route Rate (ARR), and Gold Utility (GU). Interaction quality is measured by First-Action Accuracy (FAA), Average Questions (Avg. Q), and Unnecessary-Ask Rate (UAR), while cost-aware performance is measured by Net Reward (NR) and Tokens per Episode (Token/ep.). We use NR as the primary cost-aware metric because it directly balances terminal routing utility against clarification cost.

Full inference and offline interaction settings are provided in Appendices B and D, respectively. Detailed baseline definitions are given in Appendix D.4, and complete metric definitions in Appendix D.5.

Table 2: Ablation results of VISTA on the full test split with DeepSeek-v4-Flash. ΔNR denotes the relative percentage change in NR from the full model. Best and second-best results according to the indicated direction are shown in **bold** and underlined.

Variant	ΔNR (%)	NR↑	PRA↑ (%)	ARR↑ (%)	FAA↑ (%)	GU↑	Avg. Q↓ (Q/ep.)	Token↓ (k/ep.)
VISTA (Full)	0.00%	**0.4813**	**52.62**	**72.57**	**62.68**	**0.5082**	0.5372	9.83
Decision-Aware State Modeling								
Independent T/P	-4.24%	0.4609	50.21	69.84	58.23	0.4894	0.5696	9.31
Fixed hypotheses	-6.00%	0.4524	49.57	69.12	56.74	0.4826	0.6038	9.72
w/o Candidate-distinguishing dimensions	-8.27%	0.4415	48.17	67.34	51.41	0.4662	0.4946	8.66
Value-Guided Clarification								
w/o EVPI stopping	0.00%	**0.4813**	**52.62**	**72.57**	**62.68**	**0.5082**	0.5372	13.53
Task-only utility	-5.59%	0.4544	49.22	68.31	54.16	0.4798	0.5076	8.91
Skill Representation								
name+description Card	-9.39%	0.4361	47.12	66.48	49.68	0.4587	0.4528	7.29
name-only Card	-16.04%	0.4041	43.24	62.17	45.12	0.4248	**0.4136**	**5.81**

6.2 Main Results

Overall Performance. Table 1 shows that VISTA achieves the highest NR and FAA under both base LLMs, reaching $0.481/62.68\%$ with DeepSeek-v4-Flash and $0.449/59.15\%$ with Mimo-v2.5. These consistent gains demonstrate stronger cost-aware clarify-or-route decisions.

Adaptive Clarification Is Necessary. Fixed-Round Clarification improves terminal routing quality but incurs high interaction cost by always asking two questions, resulting in lower NR. This highlights the need for adaptive clarify-or-route decisions that balance routing quality and interaction cost.

Structured Joint States Improve Value Estimation. Generic VoI improves over uncertainty-agnostic baselines, while VISTA further raises NR from 0.450 to 0.481 and from 0.422 to 0.449 with fewer questions. This indicates that candidate-conditioned joint task–profile states provide a better basis for estimating clarification value.

Value Guidance Improves the Routing–Interaction Trade-off. VISTA-ER attains slightly stronger terminal routing quality but asks substantially more questions. VISTA nearly halves Avg. Q while retaining comparable PRA and GU, yielding higher NR and FAA. This shows that VoI improves the trade-off by clarifying only when the expected routing gain justifies the interaction cost.

6.3 Clarification Behavior

Complete clarification-behavior results are reported in Appendix E.1.

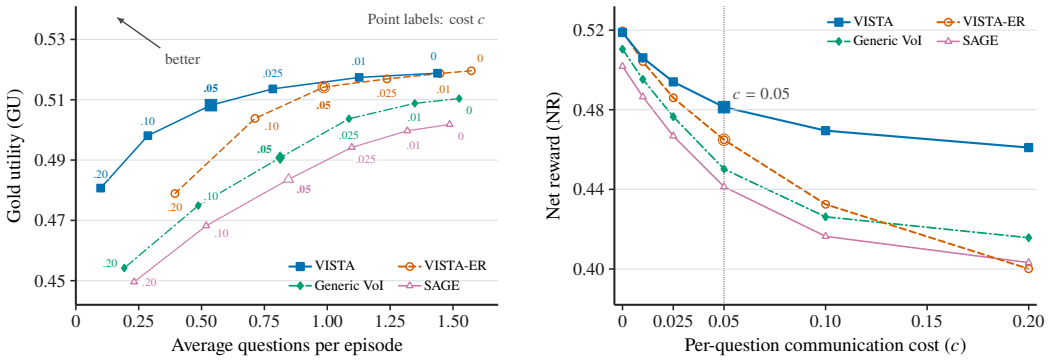
From Uncertainty Reduction to Decision-Relevant Clarification. Fixed-Round Clarification improves routing over non-interactive methods but incurs excessive interaction, showing that resolving more uncertainty is not necessarily better. More importantly, VISTA-ER retains our candidate-conditioned joint state representation while replacing VoI with entropy reduction. Its strong task- and profile-side question-selection accuracy shows that the structured state effectively identifies what is uncertain and relevant to the candidate skills.

Value-Guided Clarify-or-Route Decisions. Compared with VISTA-ER, VISTA achieves stronger Route/No-Ask accuracy and FAA with substantially lower UAR. Since the two variants share the same state representation and question space, this comparison isolates the role of the decision criterion: structured states identify what may need clarification, while VoI determines whether resolving it is worth the interaction cost. Thus, VISTA allocates the limited clarification budget according to downstream routing value rather than uncertainty alone. Qualitative case studies in Appendix F illustrate how VISTA selectively clarifies task- or profile-side uncertainty and stops once further clarification no longer changes the preferred route.

6.4 Ablation and Factorial Analysis

Table 2 examines the main components of VISTA using DeepSeek-v4-Flash.

Routing-Relevant State Representation. Independent task/profile modeling, fixed hypothesis support, and removing candidate-distinguishing dimensions all degrade performance, with the largest



(a) Utility–interaction trade-off **(b) Net reward under varying communication costs**

Figure 3: **Sensitivity to communication cost** $c \in \{0, 0.01, 0.025, 0.05, 0.10, 0.20\}$. (a) Gold utility versus average clarification frequency. (b) Net Reward across different per-question communication costs. VISTA maintains a favorable utility–interaction trade-off across the sweep, with a clearer advantage as clarification becomes more costly. The main experiments use $c = 0.05$.

drop occurring without candidate conditioning. These results support the central representation principle of VISTA, showing that the belief need not cover the full task–profile space, but should preserve the joint distinctions that can change the route among the current candidates.

Value-Guided Decision Components. Task-only utility degrades both routing and first-action decisions, demonstrating that clarification value must reflect both task and profile compatibility. Removing EVPI preserves routing quality but increases inference cost, supporting its role as an efficient early stopping and pruning mechanism.

Complementary Roles of State and Value. The controlled factorial analysis in Appendix E.2 demonstrates the same pattern, with structured joint modeling improving the representation of routing-relevant uncertainty and VoI improving how that uncertainty is translated into clarification and stopping decisions. Their combination yields the strongest cost-aware clarify-or-route behavior.

Skill Representation. Simplifying *SkillRoutingCards* consistently degrades performance, indicating that documented task capabilities and profile conditions are important for identifying candidate-relevant distinctions and supporting reliable downstream routing decisions.

6.5 Sensitivity to Communication Cost

As shown in Figure 3, VISTA maintains the strongest utility–interaction trade-off across the evaluated communication costs. As c increases, it progressively reduces clarification while preserving comparatively high routing utility, indicating that its asking behavior adapts to the cost of information acquisition. Consequently, VISTA achieves the highest NR throughout the sweep, with its advantage becoming more pronounced when clarification is costly. This suggests that VISTA is robust to different interaction-cost settings. Detailed utility–cost analyses are provided in Appendix E.3.

7 Conclusion

We formulated personalized skill routing under joint task–profile uncertainty as a finite-horizon clarify-or-route problem. The key principle is that a router need not resolve all uncertainty, but only decision-relevant uncertainty whose resolution can improve the downstream skill decision enough to justify its interaction cost. VISTA operationalizes this principle through a candidate-conditioned joint task–profile belief and value-guided clarification and stopping. We also constructed a controlled benchmark that evaluates both route-changing ambiguity and unnecessary clarification. Across two base LLMs, VISTA achieved stronger cost-aware routing and stopping behavior while maintaining competitive terminal routing quality. We hope this work advances personalized skill routing toward resolving decision-relevant uncertainty while avoiding unnecessary user interruption in real-world agent systems. Future work will extend the framework to dynamic agentic workflows and long-term user memory (Li et al., 2026b; Xu et al., 2026).

AI USE STATEMENT

Generative AI tools were used to assist with language polishing, code development and review, and selected stages of benchmark construction. Code assistance included benchmark preprocessing and construction scripts, experimental pipelines, implementations of VISTA and the baselines, model/API interaction, and evaluation and analysis scripts. For benchmark construction, generative AI supported the organization of source instances, construction of controlled counterfactual cases, and consistency checking of intermediate artifacts. All AI-assisted text and code were manually reviewed, revised, and verified by the authors.

ETHICS STATEMENT

This work follows the ICLR Code of Ethics. We use real-world skill-routing queries and user profiles collected from a deployed skill-routing platform, which may contain routing-relevant information such as language, region, platform, and payment preferences. Direct identifiers, including user names and IP addresses, are excluded from the research data and are not included in the released benchmark. We retain only information necessary for studying task–profile-dependent skill routing and release processed benchmark instances rather than directly identifying user information. We also recognize the privacy risks associated with personalized routing systems and recommend appropriate data minimization, access control, and retention safeguards in practical deployments.

REPRODUCIBILITY STATEMENT

We provide detailed descriptions of the VISTA algorithm, benchmark construction, evaluation protocol, and experimental configuration in the main text and appendices. The appendices document the LLM-based semantic instantiation, offline interaction protocol, runtime and validation procedures, baseline implementations, evaluation metrics, benchmark construction process, data splits, and human-audit protocol. We also report the model configurations and key hyperparameters required to reproduce the experiments. The supplementary material includes the complete source code and dataset, both of which will be released in a public GitHub repository upon acceptance.

REFERENCES

Mimo-v2.5. https://huggingface.co/collections/XiaomiMiMo/mimo-v25, 2026.

Anthropics. Skills. https://github.com/anthropics/skills, 2026. Accessed: 2026-09-16.

Ruishuo Chen, Xun Wang, Yu Chen, Zhuoran Li, and Longbo Huang. The router within: Eliciting native skill routing from a frozen LLM, 2026. URL https://arxiv.org/abs/2609.15982.

Jacob Cohen. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46, 1960. doi: 10.1177/001316446002000104. URL https://doi.org/10.1177/001316446002000104.

DeepSeek-AI, Anyi Xu, Bangcai Lin, Bing Xue, et al. DeepSeek-V4: Towards highly efficient million-token context intelligence, 2026. URL https://arxiv.org/abs/2606.19348.

Mengyi Deng, Zhiwei Li, Xin Li, Tingyu Zhu, Ying Zhao, Zhijiang Guo, and Wei Wang. Uncertainty-aware clarification in LLM agents with information gain, 2026. URL https://arxiv.org/abs/2606.03135.

Jiandong Ding, Honglei Ji, Ming Liu, and Tao Duan. Right family, wrong skill: Evaluating risk exposure in agent skill retrieval, 2026. URL https://arxiv.org/abs/2606.10388.

Yijiang River Dong, Tiancheng Hu, Zheng Hui, Caiqi Zhang, Ivan Vulić, Andreea Bobu, and Nigel Collier. Value of information: A framework for human–agent communication. In *Proceedings of*

the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 42879–42896, San Diego, California, United States, 2026. Association for Computational Linguistics. doi: 10.18653/v1/2026.acl-long.1987. URL https://aclanthology.org/2026.acl-long.1987/.

Yue Feng, Hossein A. Rahmani, Aldo Lipani, and Emine Yilmaz. Towards asking clarification questions for information seeking on task-oriented dialogues, 2023. URL https://arxiv.org/abs/2305.13690.

Yonatan Geifman and Ran El-Yaniv. SelectiveNet: A deep neural network with an integrated reject option. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2151–2159. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/geifman19a.html.

Zhengyao Gu and Mark Hopkins. On the evaluation of neural selective prediction methods for natural language processing. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7888–7899, Toronto, Canada, 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.437. URL https://aclanthology.org/2023.acl-long.437/.

Yupu Hao, Pengfei Cao, Zhuoran Jin, Huanxuan Liao, Yubo Chen, Kang Liu, and Jun Zhao. Evaluating personalized tool-augmented LLMs from the perspectives of personalization and proactivity. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 21897–21935, Vienna, Austria, 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.acl-long.1064. URL https://aclanthology.org/2025.acl-long.1064/.

Ismail Hossain, Sai Puppala, Md Jahangir Alam, Tanzim Ahad, and Sajedul Talukder. SkillVetBench: LLM-as-judge for multi-dimensional security risk evaluation in open-source LLM agent skills, 2026. URL https://arxiv.org/abs/2606.15899.

Ronald A. Howard. Information value theory. *IEEE Transactions on Systems Science and Cybernetics*, 2(1):22–26, 1966. doi: 10.1109/TSSC.1966.300074. URL https://doi.org/10.1109/TSSC.1966.300074.

Ryangkyung Kang, Hongcheol Cho, and Youngeun Kim. SkillRet: A large-scale benchmark for skill retrieval in LLM agents, 2026. URL https://arxiv.org/abs/2605.05726.

Katarzyna Kobalczyk, Nicolás Astorga, Tennison Liu, and Mihaela van der Schaar. Active task disambiguation with LLMs. In *International Conference on Learning Representations*, volume 2025, pp. 37823–37847, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/5e07476b6bd2497e1fbd11b8f0b2de3c-Paper-Conference.pdf.

K. Krippendorff. *Content Analysis: An Introduction to Its Methodology*. Sage, 2004. ISBN 9780761915454. URL https://books.google.com/books?id=q657o3M3C8cC.

Xiangyi Li, Yimin Liu, Wenbo Chen, Bingran You, Zonglin Di, Yifeng He, Shenghan Zheng, Kyoung Whan Choe, Jiankai Sun, Shuyi Wang, Chujun Tao, Binxu Li, Xuandong Zhao, Hejia Geng, Xiaojun Wu, Junwei Zhou, Xiaokun Chen, Hanwen Xing, Yubo Li, Qunhong Zeng, Di Wang, Yuanli Wang, Roey Ben Chaim, Penghao Jiang, Haotian Shen, Luyang Kong, Xinyi Liu, Runhui Wang, Xuanqing Liu, Jiachen Li, Xin Lan, Yueqian Lin, Wengao Ye, Junwei He, Songlin Li, Yue Zhang, Yipeng Gao, Yijiang Li, Ze Ma, Liqiang Jing, Tianyu Wang, Kaixin Li, Yiqi Xue, Haoran Lyu, Yizhuo He, Yuchen Tian, Shutong Wu, Bowei Wang, Yixuan Gao, Bo Chen, Litong Liu, Sikai Cheng, Jiajun Bao, Shuaicheng Tong, Shuwen Xu, Terry Yue Zhuo, Tinghan Ye, Qi Qi, Miao Li, Longtai Liao, Zelin Tan, Chang Shi, Xilin Tang, Srinath Tankasala, Boqin Yuan, Yaoyao Qian, Jianhong Tu, Chenguang Wang, Yizhou Sun, Wei Wang, Aaron Taylor, Ziyue Yang, Changkun Guan, Zhikang Dong, Xinyu Zhang, Steven Dillmann, Han chung Lee, and Dawn Song. Skillsbench: Benchmarking how well agent skills work across diverse tasks, 2026a. URL https://arxiv.org/abs/2602.12670.

Zhuofeng Li, Haoxiang Zhang, Seungju Han, Sheng Liu, Jianwen Xie, Yu Zhang, Yejin Choi, James Y Zou, and Pan Lu. In-the-flow agentic system optimization for effective planning and tool use. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 50524–50570, 2026b. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/52c4a736cd63b3c7bf52011a1f66f3f6-Paper-Conference.pdf.

Yuan Liang, Ruobin Zhong, Haoming Xu, Chen Jiang, Yi Zhong, Runnan Fang, Jia-Chen Gu, Shumin Deng, Yunzhi Yao, Mengru Wang, Shuofei Qiao, Yida Xue, Xin Xu, Tongtong Wu, Kun Wang, Yang Liu, Zhen Bi, Jungang Lou, Yuchen Eleanor Jiang, Hangcheng Zhu, Gang Yu, Haiwen Hong, Longtao Huang, Hui Xue, Chenxi Wang, Yijun Wang, Zifei Shan, Xi Chen, Zhaopeng Tu, Feiyu Xiong, Xin Xie, Peng Zhang, Zhengke Gui, Lei Liang, Jun Zhou, Chiyu Wu, Jin Shang, Yu Gong, Junyu Lin, Changliang Xu, Hongjie Deng, Wen Zhang, Keyan Ding, Qiang Zhang, Fei Huang, Ningyu Zhang, Jeff Z. Pan, Guilin Qi, Haofen Wang, and Huajun Chen. Skillnet: Create, evaluate, and connect ai skills, 2026. URL https://arxiv.org/abs/2603.04448.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. G-eval: NLG evaluation using GPT-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 2511–2522, Singapore, 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.153. URL https://aclanthology.org/2023.emnlp-main.153/.

Ali Montazeralghaem, Guy Tennenholtz, Craig Boutilier, and Ofer Meshi. Asking clarifying questions for preference elicitation with large language models, 2025. URL https://arxiv.org/abs/2510.12015.

OpenAI. Build skills. https://learn.chatgpt.com/docs/build-skills, 2026a. Accessed: 2026-08-22.

OpenAI. Agent Skills. https://github.com/openai/skills, 2026b. Accessed: 2026-09-16.

Howard Raiffa and Robert Schlaifer. *Applied Statistical Decision Theory*. Harvard Business School Publications. Division of Research, Graduate School of Business Administration, Harvard University, 1961. ISBN 9780875840178. URL https://books.google.com/books?id=T_hQAAAAMAAJ.

Sudha Rao and Hal Daumé III. Learning to ask good questions: Ranking clarification questions using neural expected value of perfect information. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2737–2746, Melbourne, Australia, 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1255. URL https://aclanthology.org/P18-1255/.

Burr Settles. Active learning literature survey. Technical Report 1648, University of Wisconsin–Madison, 2009. URL http://digital.library.wisc.edu/1793/60660.

SkillFeed. Discover, match, and personalize skills for every task. http://www.aiskillfeed.com/, 2026. Accessed: 2026-08-22.

Weihang Su, Jianming Long, Qingyao Ai, Qiaozhi He, Yichen Tang, Changyue Wang, Yiteng Tu, Yingbo Wang, and Yiqun Liu. Skill retrieval augmentation for agentic AI, 2026. URL https://arxiv.org/abs/2604.24594.

Manan Suri, Puneet Mathur, Nedim Lipka, Franck Dernoncourt, Ryan A. Rossi, and Dinesh Manocha. Structured uncertainty guided clarification for LLM agents. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 40811–40838, San Diego, California, United States, 2026. Association for Computational Linguistics. doi: 10.18653/v1/2026.findings-acl.2028. URL https://aclanthology.org/2026.findings-acl.2028/.

Alberto Testoni and Raquel Fernández. Asking the right question at the right time: Human and model uncertainty guidance to ask clarification questions. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*,

pp. 258–275, St. Julian's, Malta, March 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.eacl-long.16. URL https://aclanthology.org/2024.eacl-long.16/.

Vercel-labs. Agent Skills. https://github.com/vercel-labs/agent-skills, 2026. Accessed: 2026-09-16.

Tianle Wang, Yanghe Zou, Xiang Liu, Ziyao Huang, Chenchen Fu, and Weiwei Wu. Beyond task-only matching: Personalized skill routing with counterfactual evaluation, 2026a. URL https://arxiv.org/abs/2608.28241.

Zifei Wang, Wei Wen, Qiang Ji, Keyu Chen, Ruizhi Qiao, and Xing Sun. Skill is not document: Query-conditioned compatibility for LLM agent skill routing, 2026b. URL https://arxiv.org/abs/2606.03565.

Edwin B Wilson. Probable inference, the law of succession, and statistical inference. *Journal of the American Statistical Association*, 22(158):209–212, 1927.

Derong Xu, Yi Wen, Pengyue Jia, Yingyi Zhang, Wenlin Zhang, Yichao Wang, Huifeng Guo, Ruiming Tang, Xiangyu Zhao, Enhong Chen, and Tong Xu. From single to multi-granularity: Toward long-term memory association and selection of conversational agents. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 91581–91617, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/93e444b4a1a64815631c3c74977ee200-Paper-Conference.pdf.

Qiancheng Xu, Yongqi Li, Heming Xia, Fan Liu, Min Yang, and Wenjie Li. PEToolLLM: Towards personalized tool learning in large language models. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 21488–21503, Vienna, Austria, 2025. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.1107. URL https://aclanthology.org/2025.findings-acl.1107/.

Dong Yu, Balakrishnan Varadarajan, Li Deng, and Alex Acero. Active learning and semi-supervised learning for speech recognition: A unified framework using the global entropy reduction maximization criterion. *Computer Speech & Language*, 24(3):433–444, 2010. doi: 10.1016/j.csl.2009.03.004. URL https://www.sciencedirect.com/science/article/pii/S0885230809000187.

Michael JQ Zhang and Eunsol Choi. Clarify when necessary: Resolving ambiguity through interaction with LMs. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 5541–5558, Albuquerque, New Mexico, 2025. Association for Computational Linguistics. ISBN 979-8-89176-195-7. doi: 10.18653/v1/2025.findings-naacl.306. URL https://aclanthology.org/2025.findings-naacl.306/.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph Gonzalez, and Ion Stoica. Judging LLM-as-a-Judge with MT-bench and chatbot arena. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 46595–46623. Curran Associates, Inc., 2023. doi: 10.52202/075280-2020. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/91f18a1287b398d378ef22505bf41832-Paper-Datasets_and_Benchmarks.pdf.

YanZhao Zheng, ZhenTao Zhang, Chao Ma, YuanQiang Yu, JiHuai Zhu, Yong Wu, Tianze Xu, Baohua Dong, Hangcheng Zhu, Ruohui Huang, and Gang Yu. SkillRouter: Skill routing for LLM agents at scale, 2026. URL https://arxiv.org/abs/2603.22455.

APPENDIX TABLE OF CONTENTS

A Problem Formulation and Joint State Representation **16**

A.1 State, Observations, and Actions 16

A.2 Skill Cards and Candidate Contrasts 16

A.3 Task and Profile Hypothesis Functions 17

A.4 Compactness and Task–Profile Dependence 17

A.5 Augmented Belief Initialization 17

B Inference Procedure and Implementation **18**

B.1 EVPI Bound and One-Step Approximation 18

B.2 Online Inference Algorithm 18

B.3 LLM-Based Semantic Instantiation 20

B.4 Runtime Validation and Deterministic Decision Computation 21

C Benchmark Construction **21**

C.1 Source Inputs 21

C.2 Controlled Task–Profile Construction 21

C.2.1 Profile-Only Construction 22

C.2.2 Task-Only Construction 23

C.2.3 Joint Task–Profile Construction 23

C.3 Independent Compatibility Measurement 24

C.4 Gold Materialization, Filtering, and Observation Views 25

C.4.1 Deterministic Gold Materialization 25

C.4.2 Track-Specific Filtering 25

C.4.3 Controlled Observation Views 26

C.5 Independent Human Validation and Group-Aware Release 27

C.5.1 Benchmark Composition and Group-Aware Split 27

C.5.2 Independent Human Validation 28

C.5.3 Release Format and Information Separation 30

D Experimental Setup and Evaluation **31**

D.1 Evaluation Protocol and Shared Conditions 31

D.2 Offline User Simulation and Information Isolation 31

D.3 Model Configurations and Hyperparameters 32

D.4 Baselines and Controlled Variants 32

D.5 Evaluation Metrics 34

D.5.1 Routing Quality 34

D.5.2 Interaction Quality 34

D.5.3 Cost-Aware Performance 35

E Additional Experimental Results **35**

E.1 Clarification Behavior 35

E.2 Factorial Analysis of State Representation and Clarification Criterion 35

E.2.1 Experimental Design 35

E.2.2 Full Factorial Results 37

E.2.3 Descriptive Main Effects 37

E.2.4 Within-Representation Effect of VoI 38

E.3 Sensitivity to Communication Cost 38

F Qualitative Case Studies **38**

F.1 Case I: Clarification Changes the Preferred Route 39

F.2 Case II: Residual Uncertainty Does Not Require Clarification 40

F.3 Comparison and Implications 41

A Problem Formulation and Joint State Representation

A.1 State, Observations, and Actions

Latent routing state. For each user request, we assume an underlying routing state

$$z^\star = (t^\star, p^\star) \in \mathcal{Z} := \mathcal{T} \times \mathcal{P}, \tag{10}$$

where $z^\star$ is the user's true latent routing state, $t^\star$ is the task the user intends to accomplish, $p^\star$ denotes the profile conditions relevant to skill selection, $\mathcal{T}$ and $\mathcal{P}$ are the task and profile state spaces, respectively, and their Cartesian product $\mathcal{Z}$ defines the joint routing state space. We assume that $z^\star$ remains fixed within an episode.

Observations and belief. The agent acquires information about $z^\star$ when it can improve the routing decision. It receives the current query x and a profile memory m, both of which provide partial evidence about the latent state. Clarification history h at turn i is defined as

$$h_i = (q_0, y_0, q_1, y_1, \ldots, q_{i-1}, y_{i-1}), \qquad h_0 = \varnothing, \tag{11}$$

where q_i is the clarification question asked and y_i denotes the corresponding user response at turn i. The agent maintains a belief over possible routing states, defined as

$$b_i(z) = \Pr(Z = z \mid x, m, h_i) = \Pr(T = t, P = p \mid x, m, h_i), \tag{12}$$

where $z = (t, p)$ is a hypothesis about the user's true state, and t and p are hypotheses about the user's task and profile, respectively.

Candidate skills. Let $\mathcal{L}$ denote the large-scale skill library. Before clarification begins, an upstream retriever constructs a finite candidate set using the observed request,

$$\mathcal{S} = \mathrm{Retrieve}_N(x, m) = \{s_j\}_{j=1}^N \subseteq \mathcal{L}, \tag{13}$$

where $\mathrm{Retrieve}_N(x, m)$ is the retrieval function that returns the top-N candidate skills from the skill library given the user query x and profile memory m. Each candidate skill $s_j = (n_j, d_j, \kappa_j)$ consists of a name n_j, a description d_j, and `SKILL.md` content κ_j.

Clarify-or-route process. At each turn i, the agent may either gather additional information or terminate the interaction by selecting a skill. Candidate clarification questions are partitioned into a task-side set $\mathcal{Q}_{\mathrm{T},i}$ and profile-side set $\mathcal{Q}_{\mathrm{P},i}$. The action set at turn i is defined as

$$\mathcal{A}_i = \{\mathrm{AskTask}(q) : q \in \mathcal{Q}_{\mathrm{T},i}\} \cup \{\mathrm{AskProfile}(q) : q \in \mathcal{Q}_{\mathrm{P},i}\} \cup \{\mathrm{Route}(s) : s \in \mathcal{S}\}, \tag{14}$$

where an ask action observes an answer $y_i \in \mathcal{Y}_{q_i}$, augments the history, and induces an updated belief b_{i+1}; a route action selects one skill s and terminates the episode. The interaction is additionally bounded by a clarification budget B, yielding a finite-horizon clarify-or-route process.

A.2 Skill Cards and Candidate Contrasts

SkillRoutingCards. To ground subsequent routing decisions in the documented properties of each candidate skill, we construct a static *SkillRoutingCard* for each candidate skill s_j. Specifically, an offline extraction step maps its name n_j, description d_j, and `SKILL.md` content κ_j to

$$r_j = \phi_{\mathrm{card}}(s_j) = \phi_{\mathrm{card}}(n_j, d_j, \kappa_j), \tag{15}$$

where r_j is the static *SkillRoutingCard* of s_j, summarizing its task scope, applicable profile conditions, and execution- or risk-related properties that may affect routing.

Routing-Relevant Differences. Given the candidate cards, we identify the task- and profile-side dimensions that distinguish the available skills. Let $v_d(r_j)$ denote the value of dimension d exposed by card r_j. We define the candidate-conditioned contrast set as

$$\mathcal{D}(\mathcal{S}) = \left\{ \left(\sigma, d, \{v_d(r_j)\}_{j=1}^N\right) : \exists\, j \neq k,\ v_d(r_j) \neq v_d(r_k) \right\}, \qquad \sigma \in \{\mathrm{Task}, \mathrm{Profile}\}, \tag{16}$$

where σ indicates whether dimension d is task- or profile-side. Task-side dimensions include documented goals, operations, inputs, outputs, and non-goals and profile-side dimensions capture applicable conditions such as platform, privacy, language, and execution constraints.

By retaining only dimensions that vary across the candidate set, $\mathcal{D}(\mathcal{S})$ provides a compact contrastive representation for constructing routing-relevant task and profile hypotheses.

A.3 Task and Profile Hypothesis Functions

Conditioned on the observed query x, profile memory m, *SkillRoutingCards* $\{r_j\}_{j=1}^N$, and candidate-distinguishing dimensions $\mathcal{D}(\mathcal{S})$, we first construct a set of plausible task hypotheses and then instantiate only the profile hypotheses applicable to each task:

$$\mathcal{T}_0 = G^T\big(x, m, \{r_j\}_{j=1}^N, \mathcal{D}(\mathcal{S})\big), \qquad \mathcal{P}_0(t) = G^P\big(t, x, m, \{r_j\}_{j=1}^N, \mathcal{D}(\mathcal{S})\big), \quad t \in \mathcal{T}_0, \tag{17}$$

where $\mathcal{T}_0$ is the set of plausible task hypotheses, $\mathcal{P}_0(t)$ is the set of profile hypotheses applicable to task t, and G^T and G^P are the corresponding task- and profile-hypothesis construction functions, respectively.

A.4 Compactness and Task–Profile Dependence

Compactness of the Joint Support. A naive construction of the latent state space would form the Cartesian product of all plausible task and profile hypotheses. Let $\mathcal{T}_0$ denote the task hypothesis set and let $\mathcal{P}_0$ denote the union of all plausible profile hypotheses. Such a construction can contain up to $|\mathcal{T}_0|\,|\mathcal{P}_0|$ task–profile combinations, including many profile conditions that are inapplicable to particular tasks. Our task-first construction instead associates each task $t \in \mathcal{T}_0$ with only its applicable profile hypotheses $\mathcal{P}_0(t)$. Before deduplication, the resulting explicit support satisfies

$$|\mathcal{H}_0| \le 1 + \sum_{t \in \mathcal{T}_0} |\mathcal{P}_0(t)|, \tag{18}$$

where the additional state is the observation-consistent anchor z_{obs}. Because $\mathcal{P}_0(t)$ contains only profile hypotheses applicable to task t, the construction avoids unnecessary task–profile combinations that cannot influence the current candidate-routing problem. We additionally impose a support-size budget L_0 to keep subsequent belief updates and value computations tractable.

Task–Profile Dependence. The task-first construction specifies how the joint support is organized. It does not impose independence between task and profile. By the chain rule, the joint distribution can be factorized as

$$p(T, P \mid x, m) = p(T \mid x, m)\, p(P \mid T, x, m). \tag{19}$$

The second factor explicitly conditions the profile state on the task state. Thus, different task hypotheses may induce different sets of applicable profile hypotheses as well as different conditional probabilities over them. The task-first construction therefore preserves task–profile dependence while restricting the explicit support to combinations that are relevant to the candidate routing decision.

When Joint Structure Matters. With a purely additive routing utility and fixed task and profile marginals, the expected compatibility of a skill can be determined from the marginals alone. Joint modeling becomes important, however, once clarification and state-dependent feasibility are considered. Because the profile distribution is conditioned on the task, a task-side observation can also update the posterior over profile states and thereby change the value of subsequent clarification. Moreover, the task-first support determines which task–profile combinations are explicitly represented, and the feasibility term $F(s, z)$ may depend on the joint state, making the resulting routing utility non-additive. Thus, the distinction between joint and independent modeling extends beyond the initial marginals to the support structure, posterior updates, and state-dependent feasibility.

A.5 Augmented Belief Initialization

Joint Belief Initialization. Given an explicit joint support $\mathcal{H}_0^{(L)}$ with at most L hypotheses, we initialize an augmented belief over the retained states together with a residual state z_{other}, which captures plausible task–profile states not represented in the current support:

$$\bar{b}_0^{(L)} = \Big(\{\bar{b}_0^{(L)}(z) : z \in \mathcal{H}_0^{(L)}\}, \bar{b}_0^{(L)}(z_{\text{other}})\Big), \qquad \rho_0^{(L)} = \bar{b}_0^{(L)}(z_{\text{other}}), \tag{20}$$

where $\bar{b}_0^{(L)}$ is the augmented initial belief over the explicit support and the residual state z_{other}, and $\rho_0^{(L)}$ is the residual probability mass assigned to states outside the explicit support. For downstream routing and clarification, we condition the belief on the retained support:

$$b_0^{(L)}(z) = \frac{\bar{b}_0^{(L)}(z)}{1 - \rho_0^{(L)}}, \qquad z \in \mathcal{H}_0^{(L)}. \tag{21}$$

B Inference Procedure and Implementation

B.1 EVPI Bound and One-Step Approximation

Let Y denote the answer to a candidate question q, with predictive distribution $p(y \mid q, b_i)$ induced by the current belief and the state-conditioned answer likelihood. Assume a fixed retained support and well-defined Bayesian posteriors for all answers with positive predictive probability. Under the same belief and estimated utility used in Eqs. 4–7,

$$\begin{aligned}\mathbb{E}_{Y\sim p(\cdot|q,b_i)}\left[V_{\mathrm{route}}(b_i^{q,Y})\right] &= \mathbb{E}_Y\left[\max_{s_j\in\mathcal{S}}\mathbb{E}\left[\widehat{U}(s_j,Z)\mid Y\right]\right]\\ &\le \mathbb{E}_Y\left[\mathbb{E}\left[\max_{s_j\in\mathcal{S}}\widehat{U}(s_j,Z)\mid Y\right]\right]\\ &= \mathbb{E}_{Z\sim b_i}\left[\max_{s_j\in\mathcal{S}}\widehat{U}(s_j,Z)\right],\end{aligned} \tag{22}$$

subtracting $V_{\mathrm{route}}(b_i)$ from both sides gives $\mathrm{EVSI}(q \mid b_i) \le \mathrm{EVPI}(b_i)$. Non-negativity follows because the router can ignore Y and reuse its pre-question skill. Hence, if $\mathrm{EVPI}(b_i) \le \min_q c(q)$, every legal question satisfies $\mathrm{NetVOI}(q \mid b_i) \le 0$.

One-Step Approximation and Stopping. The EVPI gate can prune all candidate questions when no positive-net-value clarification is possible under the retained belief model. Otherwise, VISTA uses one-step EVSI, evaluating one clarification followed by routing. Complementary questions may therefore have positive multi-step value even when their individual one-step NetVOI values are non-positive. Thus, one-step stopping does not guarantee optimality for the full finite-horizon objective.

B.2 Online Inference Algorithm

Algorithm 1 summarizes the complete online inference procedure of VISTA. The procedure consists of two stages. First, episode-level quantities are initialized once, including the candidate representations, candidate-conditioned joint support, initial belief, and routing utilities. Second, the router repeatedly compares immediate routing with task- and profile-side clarification under the current belief until it commits to a skill or the clarification budget is exhausted.

Episode-Level Initialization (Lines 1–13). Line 1 obtains a static *SkillRoutingCard* for each candidate skill, and Line 2 identifies the task- and profile-side dimensions that distinguish the candidates. Lines 3–5 initialize the support capacity, construct the corresponding task-first joint support, and estimate the augmented belief together with the residual probability mass assigned to states not yet represented explicitly. Lines 6–9 implement adaptive support expansion. While the residual mass exceeds the coverage threshold and additional capacity remains, the support capacity is increased, new hypotheses are introduced from previously unrepresented candidate-distinguishing dimensions, and the belief and residual mass are re-estimated. Expansion terminates when the residual mass is at or below the threshold or $L_{\max}$ is reached. Line 10 fixes the resulting support as $\mathcal{H}_0$, and Line 11 conditions the augmented belief on this explicit support to obtain the initial decision belief b_0. Finally, Line 12 computes the routing utility for each candidate–state pair, and Line 13 initializes the interaction history.

Immediate Routing and Budget Check (Lines 14–17). Line 14 begins the clarify-or-route loop. At each turn, Line 15 computes the value of immediate routing and the corresponding current best skill. Line 16 checks whether the maximum clarification budget has been consumed. If so, Line 17 returns the current best skill. Hence, B specifies an upper bound on clarification rather than a required number of questions.

Candidate Questions and EVPI Stopping (Lines 18–23). Line 18 constructs the currently available clarification set by combining task- and profile-side questions. Line 19 checks whether any valid clarification question remains. If the set is empty, Line 20 returns the current best skill. When candidate questions are available, Line 21 computes $\mathrm{EVPI}(b_i)$. Line 22 compares this upper bound

Algorithm 1 VISTA Online Inference

Input: query x, profile memory m, candidate skills $\mathcal{S} = \{s_j\}_{j=1}^{N}$, clarification budget B
Output: selected skill $s^\star$

1: Construct SkillRoutingCards $\{r_j\}_{j=1}^{N}$ using Eq. 15;
2: Identify candidate-distinguishing dimensions $\mathcal{D}(\mathcal{S})$;
3: $L \leftarrow L_{\text{init}}$;
4: Construct the initial task-first support $\mathcal{H}_0^{(L)}$ using Eq. 3;
5: Estimate augmented belief $\bar{b}_0^{(L)}$ and residual mass $\rho_0^{(L)}$;
6: **while** $\rho_0^{(L)} > \epsilon_{\text{cov}}$ and $L < L_{\max}$ **do**
7: Increase L subject to $L \leq L_{\max}$;
8: Expand $\mathcal{H}_0^{(L)}$ with additional hypotheses from candidate-distinguishing dimensions not yet represented;
9: Re-estimate $\bar{b}_0^{(L)}$ and $\rho_0^{(L)}$;
10: Set the finalized support $\mathcal{H}_0 \leftarrow \mathcal{H}_0^{(L)}$;
11: Obtain the explicit belief b_0 using Eq. 21;
12: Compute $\widehat{U}(s_j, z)$ for all $s_j \in \mathcal{S}$ and $z \in \mathcal{H}_0$ using Eq. 1;
13: $h_0 \leftarrow \varnothing$;
14: **for** $i = 0, \ldots, B$ **do**
15: Compute $V_{\text{route}}(b_i)$ and s_i^{route} using Eq. 4;
16: **if** $i = B$ **then**
17: **return** s_i^{route};
18: Construct $\mathcal{Q}_i = \mathcal{Q}_{\text{T},i} \cup \mathcal{Q}_{\text{P},i}$;
19: **if** $\mathcal{Q}_i = \varnothing$ **then**
20: **return** s_i^{route};
21: Compute $\text{EVPI}(b_i)$ using Eq. 5;
22: **if** $\text{EVPI}(b_i) \leq c_{\min}(\mathcal{Q}_i)$ **then**
23: **return** s_i^{route};
24: **for all** $q \in \mathcal{Q}_i$ **do**
25: Compute $p(y \mid q, b_i)$ and $\{b_i^{q,y}\}_{y \in \mathcal{Y}_q}$ using Eq. 6;
26: Compute $\text{EVSI}(q \mid b_i)$ and $\text{NetVOI}(q \mid b_i)$ using Eq. 7;
27: $q_i^\star \leftarrow \arg\max_{q \in \mathcal{Q}_i} \text{NetVOI}(q \mid b_i)$;
28: **if** $\text{NetVOI}(q_i^\star \mid b_i) \leq 0$ **then**
29: **return** s_i^{route};
30: Ask $q_i^\star$ and observe user response y_i;
31: **if** y_i indicates refusal or termination **then**
32: **return** s_i^{route};
33: $h_{i+1} \leftarrow (h_i, q_i^\star, y_i)$;
34: $b_{i+1}(z) \leftarrow b_i^{q_i^\star, y_i}(z)$ using Eq. 9;

on the gross value of additional information with the minimum communication cost among the available questions. If even perfect information cannot justify the least costly clarification, Line 23 routes immediately. This gate avoids question-specific evaluation whenever clarification cannot achieve positive net value.

Question-Specific Value Evaluation (Lines 24–29). Line 24 begins question-specific evaluation over $q \in \mathcal{Q}_i$. For each candidate question, Line 25 computes the predictive answer distribution and the hypothetical posterior associated with each possible response. Line 26 then evaluates the corresponding one-step EVSI and subtracts the communication cost to obtain NetVOI. After all candidate questions have been evaluated, Line 27 selects the task- or profile-side question with the largest NetVOI. Line 28 checks whether this maximum net value is positive. If not, Line 29 returns the current best skill rather than acquiring information whose expected routing benefit does not justify its cost.

Interaction and Belief Update (Lines 30–34). When clarification has positive net value, Line 30 presents $q_i^\star$ and observes the user response y_i. Line 31 checks whether the user declines further clarification or requests termination. In that case, Line 32 returns the current best skill. Otherwise, Line 33 appends the selected question and observed response to the ordered interaction history, and Line 34 updates the joint belief with the corresponding posterior. The updated belief is then used in the next clarify-or-route round.

Additional details on LLM-based semantic estimation and runtime execution are provided in Appendices B.3 and B.4. Offline interaction is described in Appendix D.2.

B.3 LLM-Based Semantic Instantiation

Semantic Estimation and Decision Control. VISTA separates semantic estimation from value-based decision making. Following prior work that uses language models to instantiate structured uncertainty for downstream value calculations (Dong et al., 2026), a language model provides the semantic quantities required by the routing policy, while the clarify-or-route decision itself is computed programmatically.

Specifically, the language model is used to construct task and profile hypotheses, initialize the joint belief, estimate skill compatibility and feasibility, construct clarification questions, and estimate state-conditioned answer likelihoods. Belief normalization, Bayesian updates, routing-utility aggregation, EVPI, EVSI, communication-cost adjustment, stopping, and final action selection are performed outside the language model. Thus, the language model serves as a semantic estimator rather than directly deciding whether to ask or route.

Structured Semantic Interfaces. The semantic interfaces correspond directly to the quantities in the mathematical formulation. For hypothesis construction, the model produces $\mathcal{T}_0$ and the task-conditioned profile sets $\mathcal{P}_0(t)$. For each candidate–state pair, it estimates task compatibility $R_T(s_j, t)$, profile compatibility $R_P(s_j, p)$, and the feasibility judgment required by the terminal routing utility in Eq. 1. For clarification, the model constructs task-side and profile-side question sets, $\mathcal{Q}_{\mathrm{T},i}$ and $\mathcal{Q}_{\mathrm{P},i}$, together with a finite answer space $\mathcal{Y}_q$ for each question. It further estimates the state-conditioned answer likelihood

$$p(y \mid z, q, h_i), \qquad y \in \mathcal{Y}_q, \tag{23}$$

which is used to construct the predictive answer distribution and hypothetical posteriors in Eq. 6. Importantly, these hypothetical likelihoods are estimated before the realized user response is observed.

Belief Estimation. For an explicit support $\mathcal{H}_0^{(L)}$, the language model produces non-negative scores over both the retained hypotheses and the residual state z_{other}:

$$\widetilde{b}_0^{(L)} = \left(\{\widetilde{b}_0^{(L)}(z) : z \in \mathcal{H}_0^{(L)}\}, \widetilde{b}_0^{(L)}(z_{\mathrm{other}}) \right). \tag{24}$$

These scores are optionally temperature-calibrated and normalized over the augmented support:

$$\bar{b}_0^{(L)}(z) = \frac{\left(\widetilde{b}_0^{(L)}(z)\right)^{1/\tau_b}}{\sum_{z' \in \mathcal{H}_0^{(L)} \cup \{z_{\mathrm{other}}\}} \left(\widetilde{b}_0^{(L)}(z')\right)^{1/\tau_b}}, \tag{25}$$

where $\tau_b > 0$ is the belief-calibration temperature and $\tau_b = 1$ corresponds to ordinary normalization. The resulting residual mass is

$$\rho_0^{(L)} = \bar{b}_0^{(L)}(z_{\mathrm{other}}). \tag{26}$$

Support expansion then follows the criterion defined in Section 4.1: if $\rho_0^{(L)} > \epsilon_{\mathrm{cov}}$, additional hypotheses are introduced from candidate-distinguishing dimensions not yet sufficiently represented. After expansion terminates, the belief is conditioned on the final explicit support according to Eq. 21. We then omit the superscript (L) and denote the resulting support and belief by $\mathcal{H}_0$ and b_0.

Answer-Model Boundary. The posterior in Eq. 6 is evaluated only for answers with positive predictive probability. Zero-probability hypothetical answers contribute nothing to EVSI, while out-of-support observed answers require a recovery mechanism beyond the current formulation.

Training-Free Instantiation. No parameters of the underlying language model or routing policy are learned. The same language model is used throughout a model condition. Development data are used only to select scalar configuration parameters such as α, τ_b and ϵ_{cov}, and the support-size limits. Accordingly, VISTA is training-free at both the language-model and policy levels.

B.4 Runtime Validation and Deterministic Decision Computation

Structured Inference and Validation. Within each experimental condition, all router-side semantic interfaces use the same fixed language-model configuration. The model outputs structured estimates for hypotheses, beliefs, compatibility, clarification questions, and answer likelihoods. Before entering the routing computation, these outputs are validated for required fields, admissible values, identifier consistency, and numerical ranges, with probability-valued outputs normalized programmatically. Invalid outputs are retried under the same semantic input according to a fixed retry policy.

Deterministic Decision Computation. Once the semantic estimates are obtained, belief normalization, Bayesian updates, routing utility, V_{route}, EVPI, EVSI, NetVOI, stopping, and action selection are computed deterministically by the routing program. Thus, stochasticity is limited to semantic estimation and simulated user responses (Appendix D.2).

Interaction Scope. Candidates are retrieved before clarification and remain fixed within an episode; clarification only updates the belief and reranks the existing candidates. The current implementation returns a single skill and does not model long-term memory write-back or multi-skill planning. Although the interaction is sequential, VISTA uses one-step EVSI for question evaluation rather than solving a full finite-horizon POMDP. The main experiments use $B = 2$, while the policy itself supports any finite clarification budget.

C Benchmark Construction

This appendix details the benchmark construction protocol underlying Section 5. As summarized in Figure 4, the pipeline consists of five stages: source inputs, controlled task–profile construction, independent compatibility measurement, deterministic gold materialization with track-specific filtering and observation construction, and independent human validation followed by group-aware release. The following subsections describe these stages.

C.1 Source Inputs

Real User Seeds. Each source seed consists of a real skill-routing query and its associated user profile collected from the SkillFeed platform (SkillFeed, 2026). We normalize each query into a structured task representation containing its name, description, and core requirements. The accompanying profile retains reusable user-side conditions such as region, language, platform, payment preferences, and other cross-request constraints. This normalization changes only the representation of the source input and preserves its original semantic content.

Skill Document Pool. The candidate library is constructed from a fixed snapshot containing 228,432 documented skills, each represented by its name, description, and complete `SKILL.md` content. Using a fixed repository snapshot ensures that candidate construction and compatibility measurement operate over a stable skill space and are unaffected by subsequent changes to the underlying repository.

C.2 Controlled Task–Profile Construction

We construct controlled counterfactual families along the task and profile dimensions while keeping the candidate space fixed within each family. The resulting families correspond to profile-only, task-only, and joint task–profile uncertainty.

Counterfactual State Families. Profile-only cases vary the profile while holding the task fixed, yielding (T, P_0) and (T, P_1). Task-only cases vary the task under a fixed profile, yielding (T_0, P)

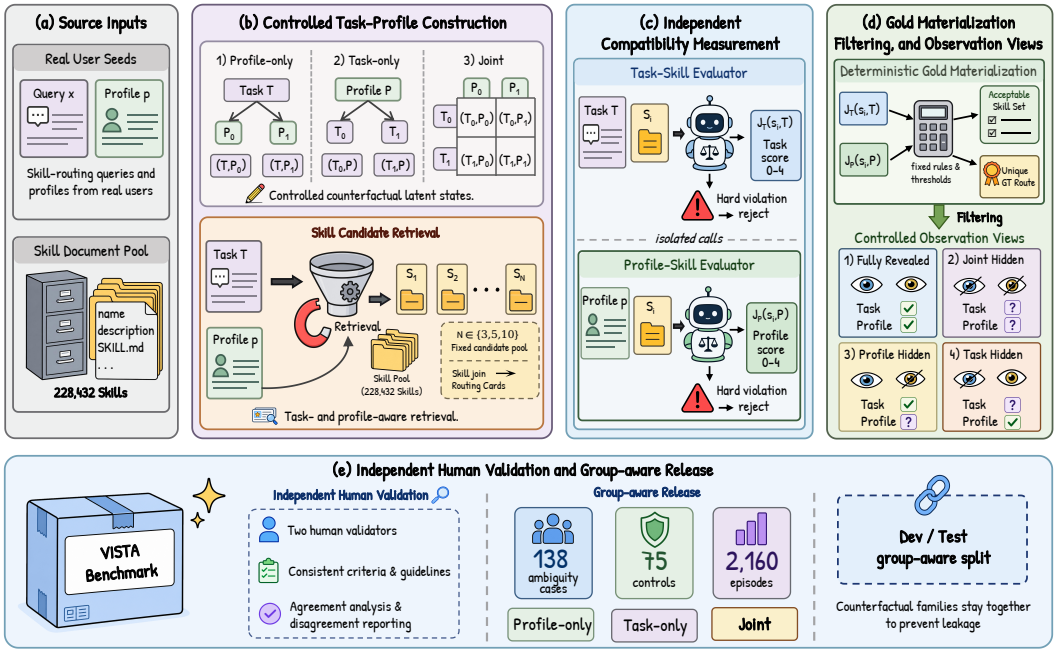


Figure 4: **Construction of the VISTA benchmark. (a)** Real user queries and profiles provide source seeds, while skill documents form the candidate library. **(b)** Controlled interventions construct profile-only, task-only, and joint latent states, together with a shared candidate pool obtained through task- and profile-aware retrieval. **(c)** Task–skill and profile–skill compatibility are measured independently. **(d)** Fixed rules deterministically materialize acceptable skill sets and reference routes, after which track-specific criteria select valid cases and controlled observation views are constructed. **(e)** Independent human validation audits the constructed labels before counterfactual families are released under group-aware development/test splits.

and (T_1, P). Joint cases vary both dimensions to form the four-state grid (T_a, P_b) with $a, b \in \{0, 1\}$. These constructions isolate profile uncertainty, task uncertainty, and their interaction while maintaining explicit control over the information that can affect the preferred skill.

Controlled Interventions. Task interventions modify only routing-relevant properties of the requested task, such as its goal, operation, input, workflow, deliverable, or required output. Profile interventions instead modify reusable user-side conditions without changing the task semantics. For each counterfactual pair, we compute exact field-level differences and group them into answerable semantic factors. The underlying field–value differences, rather than free-form paraphrases, are retained as evaluator-side facts and later provide the canonical information underlying clarification answers and hidden-state annotations.

Shared Candidate Pool. Candidate retrieval follows the task- and profile-aware SkillFeed retrieval pipeline (SkillFeed, 2026). Candidate lists from the task–profile variants within a counterfactual family are consolidated into a single ordered pool and frozen before compatibility measurement and observation masking. We evaluate candidate-set sizes $N \in \{3, 5, 10\}$ using prefixes of this frozen ordering. Consequently, all latent states within the same family are evaluated over the same candidate set $\mathcal{S}$. This prevents variation in upstream candidate recall from confounding the effect of the controlled task–profile intervention. Each candidate is represented by a static *SkillRoutingCard*, independently of benchmark labels.

C.2.1 Profile-Only Construction

State Construction. Profile-only cases are defined by

$$z_0 = (T, P_0), \qquad z_1 = (T, P_1), \qquad P_0 \neq P_1, \tag{27}$$

where the requested task is fixed and only reusable profile conditions vary. We begin with 200 source profile pairs sharing the same task. A preliminary routing screen partitions the 200 profile pairs into 75 pairs whose profile variation changes the provisional route and 125 whose route remains unchanged. The former are used to seed profile-only ambiguity construction, while the latter provide candidate route-invariant controls. Final routes are recomputed using the frozen candidate pool and independent compatibility measurements.

Candidate-Pool Construction. For each task and its associated profile variants, retrieval results are consolidated into a shared ordered ten-skill pool and joined to the fixed skill-document snapshot. One pool is discarded because one returned document cannot be mapped to a valid *SkillRoutingCard*. The remaining 199 ten-skill pools contain 1,898 unique skills. Candidate sets for $N \in \{3, 5, 10\}$ are obtained from prefixes of the corresponding frozen pool.

C.2.2 Task-Only Construction

State Construction. Task-only cases vary the task under a fixed profile:

$$z_0 = (T_0, P), \qquad z_1 = (T_1, P), \qquad T_0 \neq T_1. \tag{28}$$

The source task T_0 is preserved, while the counterfactual task modifies a minimal set of routing-relevant task properties under the same profile P. Candidate retrieval is performed for both task variants, and the returned candidates are merged into a shared ordered pool before subsequent measurement.

Initial Seed Filtering. We start from 79 task lineages and form 111 fixed-profile seeds. A semantic screen removes cases in which the shared profile explicitly repeats or entails the intended task-side distinction. Such cases would allow profile information to reveal the hidden task difference directly, undermining the intended task-only uncertainty. After filtering, 69 candidate cases remain.

Counterfactual Task Refinement. A minimal task edit does not necessarily change the preferred skill. When an initial counterfactual remains compatible with the same candidate, an eligible target skill is selected from the already frozen candidate pool and a revised counterfactual task T_1' is constructed so that its goal, input, workflow, deliverable, or output requirements better match the documented capabilities of that target.

The construction procedure may access compact candidate *SkillRoutingCards*, but it does not observe benchmark route labels, numerical compatibility scores, or profile information. Generated tasks are additionally prohibited from mentioning skill identifiers or introducing reusable user-side properties. These constraints ensure that the intervention remains task-side and does not directly encode the eventual routing label.

C.2.3 Joint Task–Profile Construction

Task-Only Backbones. Joint cases are constructed from coherent task-only backbones rather than by arbitrarily pairing independently generated task-only and profile-only examples. Each backbone provides two task states T_0/T_1, a shared frozen candidate pool, and a source profile P_0. We then construct a minimally modified profile P_1 within the same candidate pool, producing

$$\{T_0, T_1\} \times \{P_0, P_1\}. \tag{29}$$

We start from the 31 retained task-only cases. One top-10 case is excluded because its reference route under P_0 is not unique, leaving 30 joint backbones: 10 for $N = 3$, 11 for $N = 5$, and 9 for $N = 10$.

Blind Profile Counterfactual Proposal. Profile counterfactuals are first proposed without exposing reference routes, acceptable sets, numerical compatibility scores, or a desired winner. The construction input contains only T_0/T_1, the source profile P_0, and the documented profile conditions represented in the candidate *SkillRoutingCards*. The proposed profile P_1 is restricted to small modifications of reusable profile fields, implemented through `add` or `replace` operations. This restriction limits semantic drift and preserves the interpretability of the profile intervention.

Targeted Proposal Fallback. Blind profile modifications do not always induce a valid joint routing pattern. When blind proposals fail, a targeted proposal stage may use coarse candidate roles derived from measurements already available under P_0. Even in this stage, numerical compatibility scores, reference-route labels, and all outcomes under the proposed P_1 remain hidden. Thus, targeted generation may identify potentially relevant candidate distinctions but cannot directly observe the routing outcome produced by the new profile.

Structural and Semantic Validation. Every proposed P_1 undergoes structural validation followed by an independent semantic screen. The screen evaluates whether the profile modification is natural, answerable through clarification, clearly separable from task semantics, and grounded in documented candidate conditions. Only proposals passing these checks proceed to compatibility measurement.

Proposal Statistics. Blind generation produces 120 profile proposals, of which 119 pass the semantic screen. Two subsequent targeted rounds generate 150 and 309 proposals, of which 148 and 300 pass, respectively. After deduplication, 566 candidate joint grids remain: 109 for $N = 3$, 207 for $N = 5$, and 250 for $N = 10$.

C.3 Independent Compatibility Measurement

Fixed Construction Model. All LLM-assisted construction stages use a fixed external model configuration. *SkillRoutingCard* extraction, counterfactual semantic processing, and compatibility judgments use `deepseek-v4-flash` with temperature zero and low reasoning effort. Compatibility evaluation is performed independently for each task–skill or profile–skill pair, and the evaluator never directly selects the final reference route.

Following LLM-as-a-judge evaluation paradigms (Zheng et al., 2023; Liu et al., 2023; Wang et al., 2026b; Hossain et al., 2026), task and profile compatibility are measured in isolation before reference routes are materialized.

Isolated Task Compatibility. For every candidate–state pair, the task-side evaluator receives exactly one task T and one candidate skill s_j and returns

$$J_T(s_j, T) \in \{0, 1, 2, 3, 4\}, \tag{30}$$

together with a task-side hard-violation flag. The evaluator observes neither the user profile nor other candidates. Thus, the judgment measures compatibility between the documented skill and the task itself without introducing profile information or pairwise candidate preference.

Isolated Profile Compatibility. Profile compatibility is measured separately. The profile-side evaluator receives exactly one profile P and one candidate skill s_j and returns

$$J_P(s_j, P) \in \{0, 1, 2, 3, 4\}, \tag{31}$$

together with the corresponding profile-side hard-violation flag. The evaluator observes neither the task nor other candidates. This strict information separation prevents either routing axis from implicitly influencing compatibility measurement on the other.

Normalized Compatibility Scores. The discrete judgments are normalized to the compatibility scores used by the routing utility:

$$R_T(s_j, T) = \frac{J_T(s_j, T)}{4}, \qquad R_P(s_j, P) = \frac{J_P(s_j, P)}{4}. \tag{32}$$

Profile-Only Measurements. Each retained ten-skill profile-only pool requires one task-side judgment and two profile-side judgments per candidate. This yields $10 \times (1 + 2) = 30$ isolated compatibility measurements per pool and 5,970 measurements in total. For each candidate-set size N, compatibility information from the corresponding candidate prefix is used independently when materializing the reference routes.

Task-Only Rescoring. All generated task states are independently rescored against the complete frozen candidate pool. Across the 69 candidate cases, this produces 2,070 case-level judgments, corresponding to 1,889 distinct evaluator requests after identical inputs are shared. The resulting compatibility measurements, rather than the intended construction target, determine whether a counterfactual can be retained.

Joint Measurements. For every joint profile proposal passing structural and semantic validation, a fresh profile–skill judgment is obtained for each candidate under P_1. Together with the task-side and P_0 measurements already associated with the backbone, these new judgments provide the compatibility information required to materialize all four routes in the joint task–profile grid.

C.4 Gold Materialization, Filtering, and Observation Views

C.4.1 Deterministic Gold Materialization

Benchmark labels are deterministically derived from the isolated compatibility scores and hard-violation flags using the routing utility in Eq. 1. We denote this benchmark-side utility by $U^\dagger$, which is distinct from the router estimate $\widehat{U}$. We set $\alpha = 0.5$ during benchmark construction.

For each latent state, the acceptable skill set is

$$\mathcal{A}(T, P) = \{s_j \in \mathcal{S} : J_T(s_j, T) \geq 2,\ J_P(s_j, P) \geq 1,\ \text{no hard violation}\}\,, \tag{33}$$

and the corresponding reference route is

$$s^\star(T, P) = \arg \max_{s_j \in \mathcal{A}(T,P)} U^\dagger(s_j, (T, P)). \tag{34}$$

We preserve both the complete acceptable set and the reference route so that the benchmark supports evaluation of strict top-1 routing and broader candidate acceptability. The track-specific criteria below determine which counterfactual families are retained.

C.4.2 Track-Specific Filtering

Profile-Only Filtering. A profile-only case is retained as route-changing ambiguity only when both profile states yield nonempty acceptable sets, unique reference routes, and

$$s^\star(T, P_0) \neq s^\star(T, P_1). \tag{35}$$

To exclude fragile route flips, each winning skill must additionally exceed its closest competitor by a normalized utility margin of at least 0.125. These criteria retain 11, 10, and 7 route-changing profile-only cases for $N = 3, 5, 10$, respectively.

Task-Only Filtering. A task-only ambiguity case must satisfy four conditions. First, the source anchor must be the task-side maximum under T_0. Second, the generated target must be the unique task-side maximum under T_1'. Third, the two anchors must have equal and non-conflicting profile compatibility. Finally, their task-side acceptable sets must be disjoint. These conditions ensure that the route change is attributable to the task intervention rather than to profile differences or weakly separated task preferences. Candidate generation and filtering are performed separately for each N. The resulting track contains 10, 11, and 10 task-only ambiguity cases for $N = 3, 5, 10$, respectively.

Joint Filtering. For each joint grid, define

$$r_{ab} = s^\star(T_a, P_b), \qquad a, b \in \{0, 1\}. \tag{36}$$

A retained joint ambiguity case must exhibit decision dependence on both uncertainty axes:

$$|\{r_{00}, r_{01}, r_{10}, r_{11}\}| \geq 3, \qquad \exists b :\ r_{0b} \neq r_{1b}, \qquad \exists a :\ r_{a0} \neq r_{a1}. \tag{37}$$

The first condition requires at least three distinct preferred routes across the four latent states. The remaining conditions require task variation to change the route for at least one profile state and profile variation to change the route for at least one task state. Every latent state must additionally have a nonempty acceptable set and a unique reference route. These constraints exclude grids in which one uncertainty axis is nominally present but has no effect on the routing decision.

Table 3: Controlled observation views used for joint ambiguity cases.

Observation view	Task	Profile
Fully revealed	Revealed	Revealed
Joint hidden	Hidden	Hidden
Profile hidden	Revealed	Hidden
Task hidden	Hidden	Revealed

Robustness to Utility Weighting. To reduce sensitivity to the balance between task and profile compatibility, the preferred route for each joint latent state must remain unchanged for

$$\alpha \in \{0.4, 0.5, 0.6\}. \tag{38}$$

Because pairwise routing-utility differences are affine in α, agreement at the endpoints 0.4 and 0.6 rules out a winner crossing within the interval $[0.4, 0.6]$. We use $\alpha = 0.5$ to materialize the benchmark labels reported in the main experiments.

Final Joint Cases. The above criteria admit 24, 25, and 34 valid joint grids for $N = 3, 5, 10$, respectively. We retain all eligible top-3 and top-5 grids and 30 top-10 grids, yielding 79 joint ambiguity cases in total. Among the retained profile counterfactuals, 74 contain a single profile-field addition, three contain two additions, and two use a replacement. Seventy-eight cases exhibit three distinct reference routes across the four latent states, while one case exhibits four. These statistics indicate that the retained joint cases generally arise from small, interpretable profile interventions rather than broad profile rewrites.

Validation and Label Finalization. Across all tracks, generated counterfactuals and structured intermediate outputs are retained only after the corresponding structural, semantic, and track-specific validity checks. Final benchmark labels are recomputed from the retained compatibility measurements using the deterministic rules above. They are neither inherited from the source data nor directly produced by a holistic route-selection prompt. Thus, generation proposes candidate counterfactual states, isolated compatibility measurement evaluates their relationship to candidate skills, and deterministic rules together with track-specific criteria determine the final benchmark cases.

C.4.3 Controlled Observation Views

Observation Construction. Each benchmark episode distinguishes the underlying latent routing state from the information exposed to the router. For profile-only and task-only cases, each of the two counterfactual states is instantiated under both revealed and hidden observations, yielding four episodes per case. For joint cases, each latent state (T_a, P_b) is paired with the four observation conditions summarized in Table 3. Crossing four latent states with four observation views yields 16 episodes per joint case.

Hidden-Profile Observation. For profile-only cases with hidden profile information, both latent states expose only profile information shared between P_0 and P_1. For every profile field f,

$$P_{\cap}[f] = P_0[f] \cap P_1[f]. \tag{39}$$

The values distinguishing the two profile states remain evaluator-only. An empty intersection is treated as a valid partial observation rather than an explicitly revealed empty profile. This distinction ensures that the absence of visible profile evidence is not conflated with evidence that the user has no relevant profile conditions.

Task Observation Views. For task-only cases with revealed observations, the full task name, description, and core requirements are serialized into the user query. Under hidden-task observations, the two latent states instead share a common underspecified query that remains compatible with both task hypotheses while omitting their exact counterfactual differences. The profile, candidate set, and remaining interaction configuration are held fixed. Hence, the hidden observation changes only the task information visible to the router.

Table 4: VISTA benchmark composition. Pos. denotes a route-changing ambiguity case and Ctrl. a control case.

Track	$N = 3$	$N = 5$	$N = 10$	Pos.	Ctrl.	Episodes
Profile-only	21	20	17	28	30	232
Task-only	15	16	15	31	15	184
Joint	34	36	39	79	30	1,744
Total	70	72	71	138	75	2,160

Information-Preserving Masking. Hidden observations remove the counterfactual distinction without introducing additional evidence about the active latent state. In the joint-hidden view, all four states share the same underspecified task query and common profile intersection. The profile-hidden view reveals the complete task while retaining only profile information common to the counterfactual profiles, whereas the task-hidden view reveals the complete profile while using the common underspecified query. Consequently, the observation itself does not reveal which counterfactual state is active. Among the 79 joint cases, 49 have an empty common profile intersection. Such cases are represented as valid partial disclosure rather than as missing data.

Clarification-Stopping Controls. We additionally construct control cases from already measured candidate pools rather than assigning synthetic control labels. Single-axis controls are route-invariant: although task or profile information is hidden, revealing the hidden information does not change the reference route. These cases test whether a routing policy avoids clarification when the unresolved uncertainty is not decision-relevant.

Joint controls are task-sufficient. Task variation changes the reference route, whereas profile variation does not. Resolving the task uncertainty is therefore sufficient to determine the preferred skill, and subsequent profile-side clarification has no additional decision value. These controls test whether an interactive router can stop once sufficient information has been acquired rather than mechanically clarifying both dimensions. The final release contains 75 controls: 30 profile-only controls, 15 task-only controls, and 30 joint task-sufficient controls.

Reference First Actions. Reference first actions are assigned from the ambiguity track, observation view, and control type. Fully revealed episodes are labeled *route*. For ambiguity cases, hidden task or profile information is labeled *ask task* or *ask profile*, respectively, while joint-hidden cases permit either clarification direction. Route-invariant controls are labeled *route* under both observation views. Joint task-sufficient controls are labeled *ask task* when the task is hidden and *route* when it is revealed, regardless of profile visibility. These labels define the valid first-action sets used in Appendix D.5.

C.5 Independent Human Validation and Group-Aware Release

C.5.1 Benchmark Composition and Group-Aware Split

Final Composition. Table 4 summarizes the complete benchmark composition. Cases are counted separately across candidate-set sizes because both the available candidate set and the resulting routing decision may vary with N. The 213 cases include multiple variants derived from shared counterfactual families or source lineages. Joint cases generate 1,744 of the 2,160 released episodes (80.7%), reflecting their larger number of latent-state and observation-view combinations.

Family-Level Partitioning. Episodes derived from the same counterfactual family are statistically dependent because they may share source tasks, latent states, candidate pools, observation variants, or candidate-size variants. We therefore group examples by source family or task lineage and assign each group intact to a single split.

The resulting partition contains a 68-case prompt-development split with 704 episodes and a 145-case test split with 1,456 episodes. Prompt selection, α, and stopping-related development choices are determined exclusively using the development split. No counterfactual sibling or candidate-size variant of a development case is allowed to appear in the test split. This partition is fixed before final evaluation and is also used to stratify the independent human audit described below.

C.5.2 INDEPENDENT HUMAN VALIDATION

Audit Sample and Protocol. To assess whether the constructed benchmark labels withstand independent human scrutiny, we conduct a fixed-format audit with two human reviewers, denoted R1 and R2. The audit contains 150 episodes drawn from 150 distinct benchmark cases, corresponding to 6.9% of the 2,160-episode benchmark. Sampling is stratified across ambiguity track (90 joint, 30 task-only, and 30 profile-only), candidate-set size (50 each for $N \in \{3, 5, 10\}$), and case role (100 positive and 50 control). The sample contains 44 episodes from `dev` and 106 from `test`. Both reviewers independently evaluate the same 150 episodes under an identical annotation protocol and without access to one another's decisions. No sampled episode is excluded.

Review Stages. The audit uses a shared codebook and evidence packet and covers four routing-related review stages:

1. **Action & ambiguity.** Given the observed query, observed profile, clarification budget, candidate briefs, and the displayed reference action and ambiguity labels, the reviewer independently judges whether each reference label is appropriate.
2. **Preferred route.** Given the complete latent task and profile, the reviewer judges whether a proposed skill should be regarded as the preferred route. Across the 150 episodes, propositions are balanced between the benchmark reference route and a strong challenger.
3. **Candidate acceptability.** For each candidate, the reviewer judges whether the skill can satisfy the core task without a task- or profile-side hard violation. Acceptability is intentionally weaker than being the preferred route.
4. **Pairwise preference.** Given candidates A and B, the reviewer answers two binary questions: "Is A strictly better than B?" and "Is B strictly better than A?" Two No responses encode a tie, whereas two Yes responses are treated as invalid. For the 96 episodes containing clarification targets, reviewers additionally assess clarification relevance and answer-state naturalness.

Annotation Coverage. All required annotation fields use binary Yes/No judgments. Optional notes and exclusion flags are not treated as analytic labels. Each reviewer completes 150 action, ambiguity, preferred-route, and pairwise assessments, together with 900 candidate-level acceptability judgments. The two clarification fields contain 96 judgments per reviewer, and all required fields are complete.

Pairwise Reference Finalization. Before agreement is computed, pairwise reference labels are finalized using the same deterministic construction rules as the benchmark. When utility values for both candidates are available, their numerical ordering determines the pairwise relation; otherwise, the episode's unique reference route resolves the comparison. The resulting references comprise 60 A wins, 86 B wins, and 4 ties. Human reviewer judgments are not modified by this procedure.

Reviewer Form Example. Table 5 provides a compact rendering of one completed review form. We abbreviate the annotation fields as ACT (action), AMB (ambiguity), PREF (preferred route), ACC (acceptable), REL (clarification relevance), and NAT (answer-state naturalness). The full review forms retain the complete query, profile, task, and skill descriptions; only the presentation below is abbreviated.

Human–Benchmark Agreement. Table 6 reports each reviewer's agreement with the benchmark reference labels. For action and ambiguity, the reported value is the Yes rate for the displayed reference. Preferred route, candidate acceptability, and pairwise preference are evaluated against their corresponding benchmark labels. Clarification fields are reported only as proxy Yes rates because no benchmark reference label is defined for them. All confidence intervals are Wilson 95% intervals (Wilson, 1927). Candidate-level acceptability agreement reaches 84.3–85.1% across the two reviewers, whereas exact acceptable-set agreement is lower (46.0–48.7%) because a mismatch on any individual candidate makes the complete set non-exact.

Inter-Reviewer Reliability. Table 7 reports raw agreement, Cohen's κ (Cohen, 1960), and Krippendorff's α (Krippendorff, 2004) over all shared completed items. Agreement is strongest for preferred route and pairwise preference ($\kappa = 0.823$ and 0.785, respectively) and remains substantial for candidate acceptability ($\kappa = 0.624$). Action and ambiguity judgments achieve 85.3% raw agreement

Table 5: Example of a completed human-review form. "Shown value" denotes the proposition or object presented to the reviewer. All outputs are binary Yes/No judgments.

Form	Field	Shown value	R1	R2
A&A	ACT?	`route` appropriate	Yes	Yes
A&A	AMB?	`none` appropriate	Yes	Yes
Route	PREF?	S_2 is the preferred skill	Yes	Yes
Candidate	ACC?	S_1	No	No
Candidate	ACC?	S_2	Yes	Yes
Candidate	ACC?	S_3	No	No
Pair	$A > B$?	$A = S_2$, $B = S_3$	Yes	Yes
Pair	$B > A$?	$A = S_2$, $B = S_3$	No	No
Clar.	REL?	Clarification-target summary	No	No
Clar.	NAT?	Answer-state summary	Yes	Yes

S_1=`methodology-curator`; S_2=`methodology-selection`; S_3=`neversight_customer_feedback`.

Table 6: Human–benchmark audit results. Values are percentages with Wilson 95% confidence intervals. †Yes rate for a displayed reference; ‡agreement with a benchmark reference label; §proxy Yes rate without a benchmark reference label.

Measure	R1 % [95% CI]	R2 % [95% CI]	n
Action & ambiguity			
Action appropriate†	84.7 [78.0, 89.6]	84.7 [78.0, 89.6]	150
Ambiguity appropriate†	83.3 [76.6, 88.4]	83.3 [76.6, 88.4]	150
Routing-label agreement			
Preferred route‡	82.0 [75.1, 87.3]	80.0 [72.9, 85.6]	150
Candidate acceptable‡	84.3 [81.8, 86.6]	85.1 [82.6, 87.3]	900
Acceptable set exact‡	46.0 [38.2, 54.0]	48.7 [40.8, 56.6]	150
Pairwise outcome‡	83.3 [76.6, 88.4]	88.0 [81.8, 92.3]	150
Clarification proxies			
Target relevant§	86.5 [78.2, 91.9]	66.7 [56.8, 75.3]	96
Answer state natural§	88.5 [80.6, 93.5]	71.9 [62.2, 79.9]	96

but more moderate chance-corrected agreement ($\kappa = 0.435$ and 0.472), partly reflecting the high prevalence of positive judgments.

Chance-Corrected Human–Benchmark Agreement. For labels whose benchmark references vary across examples, Table 8 additionally reports Cohen's κ and nominal Krippendorff's α between each reviewer and the benchmark reference labels.

Consensus Analysis. Rather than collapsing the two reviewers into a single adjudicated judgment, we preserve them as independent evidence sources. Table 9 separates cases in which both reviewers support the benchmark reference, both reject it, or disagree. For routing-critical fields, both reviewers support the reference in 76.7% of preferred-route judgments, 75.4% of candidate-level judgments, and 81.3% of pairwise outcomes. Reviewer disagreements are retained rather than resolved through post-hoc adjudication.

Track-Level Results. Table 10 reports agreement by ambiguity track. Preferred-route agreement is lowest for profile-only cases, at 70.0% for R1 and 66.7% for R2. After post-stratification over ambiguity track, candidate-set size, and observation variant, preferred-route agreement is 82.8% for R1 and 79.6% for R2.

Audit Interpretation. The audit provides convergent evidence for the routing-critical benchmark labels. Individual human–benchmark agreement ranges from 80.0% to 88.0% for preferred route,

Table 7: Inter-reviewer reliability over all shared completed judgments.

Label	**Raw agree. %**	**Cohen's κ**	**Kripp. α**	n
Action	85.3	0.435	0.437	150
Ambiguity	85.3	0.472	0.474	150
Preferred route	91.3	0.823	0.823	150
Candidate acceptable	81.4	0.624	0.612	900
Pairwise outcome	88.7	0.785	0.785	150
Pairwise $A > B$	94.0	0.868	0.869	150
Pairwise $B > A$	92.7	0.848	0.849	150
Clarification relevance	80.2	0.477	0.451	96
Answer-state naturalness	79.2	0.371	0.347	96

Table 8: Chance-corrected agreement between each human reviewer and the benchmark reference labels.

Label	**R1 κ**	**R1 α**	**R2 κ**	**R2 α**	n
Preferred route	0.640	0.640	0.600	0.598	150
Candidate acceptable	0.684	0.682	0.675	0.672	900
Pairwise outcome	0.688	0.688	0.763	0.763	150

Note. Chance-corrected agreement is reported only for labels with varying benchmark references. Action and ambiguity are excluded because their displayed propositions are instantiated from the benchmark reference labels.

candidate acceptability, and pairwise preference, while inter-reviewer Cohen's κ ranges from 0.624 to 0.823 for these judgments. Action and ambiguity assessments provide complementary evidence, with each reviewer endorsing 84.7% of displayed action references and 83.3% of ambiguity references. Joint endorsement is 77.3% and 76.0%, respectively.

These results support the overall reliability of the constructed routing labels but do not imply unanimous correctness. Exact acceptable-set agreement is particularly sensitive to borderline candidates because disagreement on a single candidate makes the entire set non-exact. Likewise, action and ambiguity Yes rates should be interpreted as independent human support for the displayed reference labels rather than as direct estimates of ground-truth accuracy.

Clarification-related judgments provide narrower evidence. The audit packet contains clarification-target and answer-state summaries rather than a generated natural-language question followed by an actual user response. Their Yes rates (R1: 86.5/88.5%; R2: 66.7/71.9%) and lower inter-reviewer agreement ($\kappa = 0.477/0.371$) are therefore treated only as auxiliary quality evidence and are not used as additional benchmark labels. More generally, human judgments are used solely for independent validation and do not modify the benchmark labels.

C.5.3 Release Format and Information Separation

Final Release. The family-level partition is retained unchanged after the human audit. The final release contains 213 cases and 2,160 episodes in total.

Runtime–Evaluator Separation. Each released episode is represented by separate runtime and evaluator views. The runtime view contains only information available to an interactive router: the observed query, profile memory, candidate *SkillRoutingCards*, and clarification budget. The evaluator view additionally contains the latent state, answer support, acceptable skill set, reference route, expected action, and observation type. The two views are associated only for post-inference evaluation. Latent-state identifiers, hidden factor values, reference routes, and construction targets are never exposed to the router. This separation prevents evaluator-side information from leaking into clarification or routing decisions.

Table 9: Two-reviewer consensus relative to the displayed reference or benchmark label. Counts are followed by percentages.

Label	Both support	Both reject	R1 $\neq$ R2	n
Action	116 (77.3)	12 (8.0)	22 (14.7)	150
Ambiguity	114 (76.0)	14 (9.3)	22 (14.7)	150
Preferred route	115 (76.7)	22 (14.7)	13 (8.7)	150
Candidate acceptable	679 (75.4)	54 (6.0)	167 (18.6)	900
Pairwise outcome	122 (81.3)	11 (7.3)	17 (11.3)	150

Table 10: Agreement with benchmark reference labels by ambiguity track. Each cell reports R1/R2 in percent.

Track	n	Preferred	Pairwise
Profile-only	30	70.0/66.7	66.7/80.0
Task-only	30	90.0/93.3	90.0/96.7
Joint	90	83.3/80.0	86.7/87.8

D Experimental Setup and Evaluation

D.1 Evaluation Protocol and Shared Conditions

Evaluation Conditions. We evaluate the released profile-only, task-only, and joint tracks. Each episode contains a fixed candidate set $\mathcal{S}$, and all compared methods receive the same observable query x, profile memory m, candidate *SkillRoutingCards*, candidate ordering, and clarification budget B. No method is allowed to access the evaluator-only latent task, latent profile, preferred skill, acceptable skill set, or expected action.

We evaluate candidate-set sizes $N \in \{3, 5, 10\}$. The main results aggregate the full 1,456-episode test split, with each episode weighted equally as in Appendix D.5. Tables report the mean and standard deviation across five runs. The development split is used only to select prompts and scalar configuration parameters, after which all settings are frozen. The test split is reserved for final comparison.

D.2 Offline User Simulation and Information Isolation

Offline User Interaction. The routing policy is designed for interactive deployment, where a selected clarification question $q_i^\star$ is answered by the user. For offline evaluation, we instead use an evaluator-side user simulator that generates a response only after the router has selected a question.

For each episode, the simulator has access to the latent routing state $z^\star$ and returns

$$y_i \sim p_{\text{sim}}\left(y \mid z^\star, q_i^\star, h_i\right), \qquad y_i \in \mathcal{Y}_{q_i^\star}. \tag{40}$$

The simulator supplies a valid answer from the question's finite answer space, which is passed to the router in the same form as a user-selected answer. Thus, the routing rule is unchanged for responses within this answer space. Unrestricted natural-language responses would additionally require answer mapping and out-of-support recovery, which are outside the scope of the current evaluation protocol.

Information Isolation. The router and evaluator-side simulator are separated by a strict information boundary. The latent state $z^\star$, benchmark reference routes, acceptable skill sets, expected actions, and other evaluator-only annotations are never provided directly to VISTA. The router has access only to deployment-observable information: the initial query x, profile memory m, candidate *SkillRoutingCards* r, and accumulated interaction history h_i. Router-side inference and evaluator-side simulation are maintained in separate sessions and caches, following the runtime–evaluator separation in Appendix C.5.3.

The simulator is invoked only after the router commits to $q_i^\star$. In particular, the likelihood estimator used for hypothetical EVSI computation never observes the realized response y_i in advance. The

offline information flow is therefore

$$(q_i^\star, z^\star) \longrightarrow \boxed{\text{User Simulator}} \xrightarrow{y_i} \boxed{\text{VISTA}}, \tag{41}$$

with no direct access from $z^\star$ or evaluator-only labels to the router. This separation prevents privileged evaluation information from influencing hypothesis construction, belief estimation, question selection, or routing.

D.3 Model Configurations and Hyperparameters

Base-LLM Conditions. We evaluate two independent base-LLM conditions, `deepseek-v4-flash` and `mimo-v2.5`. Within one experimental condition, the same base model configuration is used across router-side semantic interfaces.

Experimental Configuration. Table 11 summarizes the scalar settings used in the main experiments.

Table 11: Main experimental configuration.

Setting	Value
Candidate sizes	$N \in \{3, 5, 10\}$
Clarification budget	$B = 2$
Task/profile utility weight	$\alpha = 0.5$
Question cost	$c = 0.05$
Initial support budget	$L_{\text{init}} = 8$
Maximum support budget	$L_{\max} = 16$
Coverage threshold	$\epsilon_{\text{cov}} = 0.25$
Belief temperature	$\tau_b = 1.0$
EVPI numerical tolerance	10^{-4}
LLM decoding temperature	0.3
Reasoning effort	`low`
Base LLMs	`deepseek-v4-flash`, `mimo-v2.5`

D.4 Baselines and Controlled Variants

Except for Retriever Top-1, all methods receive the same observable query, profile memory, and candidate *SkillRoutingCards*. No method has access to the evaluator-only latent task, latent profile, preferred skill, acceptable skill set, or expected action. All interactive methods operate over the same fixed candidate set and share the maximum clarification budget $B = 2$.

Retriever Top-1. This baseline performs neither LLM inference nor clarification and directly returns the highest-ranked skill from the upstream retriever (SkillFeed, 2026). It measures the routing capability of retrieval alone.

Direct Router. The base LLM receives the current query, visible profile memory, and candidate *SkillRoutingCards* and directly selects one skill. It performs no explicit uncertainty modeling and cannot request clarification.

Fixed-Round Clarification. This baseline always performs two clarification rounds before routing from the resulting interaction history. It does not construct the joint belief used by VISTA and does not use EVPI, EVSI, or value-based stopping. It tests whether a fixed amount of additional interaction can substitute for adaptive clarification.

Adaptive LLM Prompting. At each turn, the LLM directly decides whether to route or continue asking and, if clarification is selected, whether to ask about the task or profile. These decisions are made through prompting alone, without explicit latent-state probabilities, EVPI, or EVSI.

Random-Ask@Matched. Following random-question controls used in clarification studies (Testoni & Fernández, 2024), this baseline matches the overall clarification rate of VISTA under the same base-LLM condition, with the matching rate determined on the development split and frozen before test evaluation. At each turn, it randomly decides whether to clarify and, when asking, randomly selects among available task- and profile-side questions that have not yet been asked. This baseline separates the benefit of asking itself from that of selecting the appropriate question and interaction time.

Confidence Thresholding. Following confidence-based selective prediction (Geifman & El-Yaniv, 2019; Gu & Hopkins, 2023), this baseline reuses the probabilistic hypotheses and routing-utility estimates of VISTA, but replaces EVPI-based stopping with a confidence threshold. It routes when the confidence of the currently preferred skill exceeds a development-tuned threshold; otherwise, clarification continues.

Intent-Sim Clarification. Following the intent-similarity clarification principle of Zhang & Choi (2025), this baseline represents the plausible task/profile interpretations under the current observation and uses their uncertainty to determine whether clarification is needed. When the corresponding entropy exceeds a development-tuned threshold, the LLM generates a clarification question; otherwise, the method routes immediately. It does not explicitly use candidate-conditioned routing utility or communication cost.

SAGE-Style Structured Uncertainty. We adapt the structured-uncertainty mechanism of SAGE-Agent (Suri et al., 2026) to skill routing. Since skills do not expose tool-style argument schemas, the task/profile dimensions extracted from *SkillRoutingCards* are treated as clarification aspects. The baseline estimates uncertainty and information value over these structured aspects and applies its EVPI/aspect-cost principle to clarification. It does not use the task-first joint state support or the shared task–profile routing utility introduced by VISTA.

Generic VoI Clarification. We adapt the generic clarify-or-commit framework of Dong et al. (2026) to skill routing. Candidate skills are treated as terminal actions $a \in \mathcal{A}$, while the user's overall latent intent is represented by a single variable $\theta \in \Theta$. The method maintains a belief $b(\theta)$, estimates answer likelihoods $p(y \mid q, \theta)$ for closed-ended clarification questions, and updates the posterior for each hypothetical answer. It evaluates each question by

$$\mathrm{NetVOI}(q) = V_{\mathrm{post}}(b, q) - V(b) - c(q), \tag{42}$$

and asks the question with the largest positive NetVOI; otherwise, it commits to the skill with the highest expected utility. We use the same candidate set, communication cost $c(q) = 0.05$, and clarification budget $B = 2$ as VISTA. In this adaptation, the baseline retains a single latent intent and holistic utility $U(\theta, s_j)$, rather than our task-first joint (T, P) support. It also does not use the R_T/R_P utility decomposition, profile-side hard-violation constraints, or candidate-distinguishing dimensions for question construction.

VISTA-ER (Our variant). Following entropy-based query selection in active learning (Settles, 2009; Yu et al., 2010), VISTA-ER retains the candidate-conditioned joint support, belief, routing utilities, and question pool of VISTA, while replacing value-based clarification with entropy reduction. At each turn, it selects the question with the largest expected posterior entropy reduction and stops clarification using a development-tuned entropy threshold or when the clarification budget is exhausted.

VISTA (Ours). VISTA constructs a compact task-first joint belief over task–profile states from candidate-distinguishing dimensions. At each turn, EVPI provides a sufficient stopping test: if the maximum possible value of additional information does not exceed the minimum question cost, the router terminates clarification. Otherwise, it evaluates candidate task- and profile-side questions and selects the one with the highest positive NetVOI. The method routes when the EVPI stopping condition holds, the best NetVOI is non-positive, or the clarification budget is exhausted, selecting the skill with the highest expected routing utility under the current belief.

D.5 Evaluation Metrics

Let the test set contain E episodes. For episode e, let $\hat{s}_e$ denote the final selected skill, $s_e^\star$ the unique reference route, $\mathcal{A}_e$ the acceptable skill set, $\mathcal{S}_e$ the candidate set, and $z_e^\star$ the latent task–profile state.

For evaluation, we define the evaluator-side terminal utility as

$$U_e^{\mathrm{eval}}(s) = \mathbb{I}[s \in \mathcal{A}_e]\, U^\dagger(s, z_e^\star), \tag{43}$$

where $U^\dagger$ is the benchmark-side utility derived from isolated task–skill and profile–skill compatibility judgments together with hard-violation constraints. The router never observes U_e^{eval} and instead operates using its own estimate $\widehat{U}$.

D.5.1 Routing Quality

Preferred Route Accuracy (PRA, ↑). PRA measures the fraction of episodes in which the selected skill matches the unique reference route:

$$\mathrm{PRA} = \frac{1}{E}\sum_{e=1}^{E} \mathbb{I}[\hat{s}_e = s_e^\star]. \tag{44}$$

Acceptable Route Rate (ARR, ↑). ARR measures the fraction of episodes in which the selected skill belongs to the acceptable skill set:

$$\mathrm{ARR} = \frac{1}{E}\sum_{e=1}^{E} \mathbb{I}[\hat{s}_e \in \mathcal{A}_e]. \tag{45}$$

Because $s_e^\star \in \mathcal{A}_e$ by construction, $\mathrm{PRA} \le \mathrm{ARR}$.

Gold Utility (GU, ↑). GU measures the average evaluator-side utility of the final routing decision:

$$\mathrm{GU} = \frac{1}{E}\sum_{e=1}^{E} U_e^{\mathrm{eval}}(\hat{s}_e). \tag{46}$$

Unacceptable routes receive zero utility. GU evaluates terminal routing quality independently of clarification cost.

D.5.2 Interaction Quality

First-Action Accuracy (FAA, ↑). FAA measures whether the router's first action belongs to the benchmark-defined set of valid actions under the episode's observation and control condition (Appendix C.4.3). Actions are categorized as *route*, *ask task*, or *ask profile*. For joint-hidden episodes in which either clarification direction is valid, both *ask task* and *ask profile* are counted as correct. Formally,

$$\mathrm{FAA} = \frac{1}{E}\sum_{e=1}^{E} \mathbb{I}[a_{e,0} \in \mathcal{A}_e^{\mathrm{act}}], \tag{47}$$

where $a_{e,0}$ is the first action and $\mathcal{A}_e^{\mathrm{act}}$ is the corresponding set of valid first actions.

Conditional First-Action Accuracy. Task-Q Acc. and Profile-Q Acc. measure first-action accuracy on episodes whose reference action is *ask task* and *ask profile*, respectively. These subsets include both single-axis cases and joint-track episodes in which only the corresponding side is hidden. Joint Flexible-Ask Acc. is computed on joint-hidden episodes for which either clarification direction is valid. Route/No-Ask Acc. measures the fraction of route-reference episodes on which the router immediately routes without clarification. Each metric uses its corresponding reference-action subset as the denominator, whereas FAA aggregates over all episodes.

First-Question Direction. Joint First-Q T/P reports the task/profile distribution of the first clarification question among joint-track episodes in which the router asks. It characterizes question-selection preference rather than correctness.

Average Questions (Avg. Q, ↓). Let Q_e denote the number of clarification questions asked in episode e. The average interaction burden is

$$\text{Avg.Q} = \frac{1}{E}\sum_{e=1}^{E} Q_e. \tag{48}$$

Avg. Q is interpreted jointly with routing quality, since fewer questions are desirable only when terminal routing quality is preserved.

Unnecessary-Ask Rate (UAR, ↓). Let $\mathcal{E}_{\text{ctrl,route}}$ denote control episodes whose valid first action is immediate routing. UAR measures the fraction of these episodes on which the router asks at least one clarification question:

$$\text{UAR} = \frac{1}{|\mathcal{E}_{\text{ctrl,route}}|}\sum_{e\in\mathcal{E}_{\text{ctrl,route}}} \mathbb{I}[Q_e > 0]. \tag{49}$$

Task-sufficient controls that require task clarification are excluded, so necessary clarification is not counted as unnecessary asking. UAR is reported as a percentage.

D.5.3 Cost-Aware Performance

Net Reward (NR, ↑). NR combines terminal routing utility with communication cost. Let

$$C_e = \sum_{i=0}^{Q_e-1} c(q_{e,i}) \tag{50}$$

denote the clarification cost incurred in episode e. Then

$$\text{NR} = \frac{1}{E}\sum_{e=1}^{E}\left[U_e^{\text{eval}}(\hat{s}_e) - C_e\right]. \tag{51}$$

In the main experiments, all questions have the same cost $c(q) = c = 0.05$, so

$$\text{NR} = \text{GU} - c\,\text{Avg.Q}. \tag{52}$$

NR is the primary cost-aware metric because it directly mirrors the clarify-or-route objective by balancing terminal routing utility against interaction cost.

Tokens per Episode (Token/ep., ↓). Token/ep. reports the mean number of input, output, and provider-reported reasoning tokens consumed per episode, in thousands of tokens. It measures inference overhead and is not included in $U^{\dagger}$, GU, NR, or the NetVOI criterion used by VISTA.

E Additional Experimental Results

E.1 Clarification Behavior

Tables 12 and 13 report clarification direction, first-action accuracy, and interaction frequency under both base LLMs. The conditional accuracy measures, first-question distribution, and UAR are defined in Appendix D.5.

E.2 Factorial Analysis of State Representation and Clarification Criterion

We conduct a 3×2 factorial analysis to disentangle two components of VISTA: the representation of routing uncertainty and the criterion used to value clarification. The analysis compares holistic, independent task/profile, and joint task–profile representations under expected entropy reduction and value of information (VoI).

E.2.1 Experimental Design

Holistic Representation. The unresolved user need is represented by a single latent variable $\theta \in \Theta$, without explicitly separating task and profile factors. Holistic-ER selects questions by expected entropy reduction, while Generic VoI values them by their expected downstream routing utility.

Table 12: Clarification decision accuracy on the full test split across different base LLMs. All values are reported in percentages (%). Higher is better for all metrics. For adaptive methods, the best results are shown in **bold** and the second-best results are underlined.

Method	Task-Q Acc.↑	Profile-Q Acc.↑	Route/No-Ask Acc.↑	Joint Flexible-Ask Acc.↑	FAA↑
Base LLM: DeepSeek-v4-Flash					
Fixed-Round Clarification	61.48 ± 1.19	85.40 ± 1.04	0.00 ± 0.00	100.00 ± 0.00	46.57 ± 1.02
Adaptive LLM Prompting	47.14 ± 1.18	51.29 ± 1.26	54.17 ± 1.17	48.56 ± 1.42	50.85 ± 1.04
Random-Ask@Matched	35.19 ± 1.34	38.23 ± 1.41	53.14 ± 1.29	48.27 ± 1.55	44.73 ± 1.17
Confidence Thresholding	51.19 ± 1.06	56.29 ± 1.14	55.17 ± 1.08	50.00 ± 1.29	53.48 ± 0.95
Intent-Sim Clarification	52.14 ± 1.04	57.26 ± 1.12	56.17 ± 1.05	52.21 ± 1.25	54.63 ± 0.93
SAGE	54.14 ± 0.98	55.24 ± 1.06	58.14 ± 0.97	56.63 ± 1.18	56.28 ± 0.89
Generic VoI Clarification	55.19 ± 0.96	58.31 ± 1.03	58.17 ± 0.94	55.87 ± 1.14	57.01 ± 0.87
VISTA-ER (Ours)	**61.29 ± 1.21**	**67.42 ± 1.33**	51.86 ± 1.26	**68.17 ± 1.45**	59.56 ± 1.07
VISTA (Ours)	60.19 ± 0.86	64.19 ± 0.97	**65.83 ± 0.79**	57.12 ± 1.04	**62.68 ± 0.76**
Base LLM: Mimo-v2.5					
Fixed-Round Clarification	55.19 ± 1.34	78.23 ± 1.17	0.00 ± 0.00	100.00 ± 0.00	43.53 ± 1.12
Adaptive LLM Prompting	42.14 ± 1.31	47.26 ± 1.39	51.14 ± 1.32	47.21 ± 1.56	47.32 ± 1.14
Random-Ask@Matched	32.14 ± 1.46	36.29 ± 1.53	50.17 ± 1.41	48.46 ± 1.68	42.36 ± 1.23
Confidence Thresholding	46.19 ± 1.21	51.21 ± 1.29	52.14 ± 1.19	51.44 ± 1.44	50.16 ± 1.04
Intent-Sim Clarification	47.14 ± 1.18	52.26 ± 1.26	52.17 ± 1.17	52.88 ± 1.39	50.84 ± 1.02
SAGE	50.14 ± 1.11	55.24 ± 1.18	54.14 ± 1.09	50.19 ± 1.31	52.61 ± 0.98
Generic VoI Clarification	51.24 ± 1.08	56.13 ± 1.16	55.28 ± 1.06	54.62 ± 1.28	54.16 ± 0.96
VISTA-ER (Ours)	**57.43 ± 1.34**	**63.31 ± 1.48**	50.97 ± 1.39	**64.23 ± 1.63**	56.83 ± 1.15
VISTA (Ours)	**57.43 ± 0.97**	61.77 ± 1.08	**62.14 ± 0.91**	51.15 ± 1.16	**59.15 ± 0.84**

Table 13: Clarification behavior and interaction burden on the full test split across different base LLMs. Joint First-Q T/P reports the task/profile distribution of the first clarification question. Control UAR is computed only over control episodes whose reference action is immediate routing. Lower is better for Control UAR and Avg. Q.

Method	Joint First-Q T/P (%)	Control UAR↓ (%)	Avg. Q↓
Base LLM: DeepSeek-v4-Flash			
Fixed-Round Clarification	58.36 ± 1.71 / 41.64 ± 1.71	100.00 ± 0.00	2.00 ± 0.00
Adaptive LLM Prompting	60.43 ± 1.82 / 39.57 ± 1.82	43.19 ± 1.39	0.64 ± 0.05
Random-Ask@Matched	49.18 ± 2.26 / 50.82 ± 2.26	49.72 ± 1.62	0.54 ± 0.05
Confidence Thresholding	61.28 ± 1.61 / 38.72 ± 1.61	36.81 ± 1.27	0.71 ± 0.06
Intent-Sim Clarification	58.73 ± 1.58 / 41.27 ± 1.58	31.25 ± 1.18	0.78 ± 0.06
SAGE	62.16 ± 1.49 / 37.84 ± 1.49	25.49 ± 1.06	0.85 ± 0.07
Generic VoI Clarification	61.47 ± 1.44 / 38.53 ± 1.44	22.43 ± 1.02	0.81 ± 0.07
VISTA-ER (Ours)	56.37 ± 1.93 / 43.63 ± 1.93	43.40 ± 1.37	0.99 ± 0.08
VISTA (Ours)	63.24 ± 1.43 / 36.76 ± 1.43	**16.60 ± 0.92**	0.54 ± 0.04
Base LLM: Mimo-v2.5			
Fixed-Round Clarification	59.41 ± 1.92 / 40.59 ± 1.92	100.00 ± 0.00	2.00 ± 0.00
Adaptive LLM Prompting	61.73 ± 2.11 / 38.27 ± 2.11	47.36 ± 1.52	0.68 ± 0.06
Random-Ask@Matched	48.47 ± 2.43 / 51.53 ± 2.43	52.85 ± 1.74	0.57 ± 0.05
Confidence Thresholding	62.58 ± 1.88 / 37.42 ± 1.88	40.63 ± 1.39	0.75 ± 0.07
Intent-Sim Clarification	59.84 ± 1.81 / 40.16 ± 1.81	35.76 ± 1.31	0.82 ± 0.07
SAGE	63.42 ± 1.72 / 36.58 ± 1.72	29.44 ± 1.18	0.89 ± 0.08
Generic VoI Clarification	62.74 ± 1.67 / 37.26 ± 1.67	27.85 ± 1.14	0.94 ± 0.08
VISTA-ER (Ours)	57.84 ± 2.17 / 42.16 ± 2.17	45.83 ± 1.56	1.03 ± 0.09
VISTA (Ours)	64.47 ± 1.67 / 35.53 ± 1.67	**20.35 ± 1.04**	0.57 ± 0.05

Independent Task/Profile Representation. Task and profile uncertainty are represented by separate beliefs, $b_T(t)$ and $b_P(p)$ with the joint belief approximated as $b(t,p) = b_T(t)b_P(p)$. This representation distinguishes task-side and profile-side uncertainty but does not preserve their dependence. Independent-ER uses expected entropy reduction, whereas Independent-VoI uses expected downstream routing value.

Joint Task–Profile Representation. The joint variants maintain an explicit belief $b(t,p)$ over candidate-conditioned task–profile states. VISTA-ER and VISTA share this state representation but differ in their clarification criterion: VISTA-ER uses expected entropy reduction, whereas VISTA uses cost-aware decision value.

Controlled Factors. All six variants use the same test episodes, candidate sets, base LLM, clarification budget $B = 2$, per-question communication cost $c = 0.05$, user-simulation protocol, and evaluation procedure. Thus, the factorial design isolates the effects of state representation and clarification criterion while holding the remaining conditions fixed.

E.2.2 Full Factorial Results

Table 14: **Factorial analysis of state representation and clarification criterion** with DeepSeek-v4-Flash. FAA, Route/No-Ask, and UAR are reported in percentages. Higher is better for NR, FAA, Route/No-Ask, and GU; lower is better for UAR and Avg. Q. Best results are shown in bold and second-best results are underlined.

State	Criterion	Method	NR↑	FAA↑	Route/No-Ask↑	UAR↓	Avg. Q↓	GU↑
Holistic	Entropy	Holistic-ER	0.4455	54.38	49.28	47.85	1.0268	0.4968
Holistic	VoI	Generic VoI	0.4502	57.01	58.17	22.43	0.8127	0.4908
Independent T/P	Entropy	Independent-ER	0.4554	57.18	52.48	40.83	0.9986	0.5053
Independent T/P	VoI	Independent-VoI	0.4609	58.23	<u>61.28</u>	<u>20.76</u>	<u>0.5696</u>	0.4894
Joint T/P	Entropy	VISTA-ER	<u>0.4649</u>	<u>59.56</u>	51.86	43.40	0.9864	**0.5142**
Joint T/P	VoI	VISTA	**0.4813**	**62.68**	**65.83**	**16.60**	**0.5372**	<u>0.5082</u>

Effect of State Representation. Table 14 shows that more structured state representations consistently improve NR. Under entropy reduction, NR increases from 0.4455 for the holistic state to 0.4554 for independent T/P modeling and 0.4649 for joint T/P modeling. Under VoI, the corresponding values are 0.4502, 0.4609, and 0.4813. Holding the VoI criterion fixed, joint modeling therefore improves NR by 0.0311 over the holistic representation and by 0.0204 over independent T/P modeling, supporting the importance of preserving task–profile dependence in the routing belief.

Effect of Clarification Criterion. VoI improves NR over entropy reduction for all three state representations, with the largest gain under the joint representation (0.4813 vs. 0.4649). For the joint state, VoI also increases Route/No-Ask accuracy from 51.86% to 65.83%, reduces UAR from 43.40% to 16.60%, and lowers Avg. Q from 0.9864 to 0.5372. Although VISTA-ER attains slightly higher GU, VISTA achieves higher NR and FAA with substantially less interaction, indicating that VoI primarily improves selective clarification and stopping.

E.2.3 Descriptive Main Effects

To summarize the two factorial dimensions, we average each state representation over the two clarification criteria and each criterion over the three state representations. These values are descriptive averages used to summarize the two factorial dimensions.

State Representation. As summarized in Table 15, averaging over clarification criteria, the joint representation achieves the highest NR (0.4731), FAA (61.12%), and Route/No-Ask accuracy (58.84%).

Clarification Criterion. Averaging over state representations, VoI increases NR from 0.4553 to 0.4641 and FAA from 57.04% to 59.30%, while improving Route/No-Ask accuracy by 10.55 points

Table 15: **Descriptive averages for the two factorial dimensions.** State-representation averages are taken over Entropy and VoI, while criterion averages are taken over Holistic, Independent T/P, and Joint T/P representations.

Setting	NR↑	FAA↑	Route/No-Ask↑	UAR↓	Avg. Q↓	GU↑
State representation, averaged over criterion						
Holistic	0.4479	55.69	53.72	35.14	0.9198	0.4938
Independent T/P	0.4582	57.71	56.88	30.80	0.7841	0.4974
Joint T/P	**0.4731**	**61.12**	**58.84**	**30.00**	**0.7618**	**0.5112**
Clarification criterion, averaged over representation						
Entropy	0.4553	57.04	51.21	44.03	1.0039	**0.5054**
VoI	**0.4641**	**59.30**	**61.76**	**19.93**	**0.6398**	0.4961

and reducing UAR by 24.10 points. Entropy reduction yields slightly higher mean GU, reflecting its tendency to acquire more information even when that information has limited decision value.

E.2.4 Within-Representation Effect of VoI

Table 16 reports the change obtained by replacing entropy reduction with VoI while holding the state representation fixed.

Table 16: **Effect of replacing entropy reduction with VoI within each state representation.** Positive values indicate increases under VoI; negative values for UAR and Avg. Q indicate fewer unnecessary clarifications and fewer questions.

State	ΔNR	ΔFAA	ΔRoute/No-Ask	ΔUAR	ΔAvg. Q	ΔGU
Holistic	+0.0047	+2.62	+8.90	-25.42	-0.2141	-0.0060
Independent T/P	+0.0055	+1.04	+8.79	-20.07	-0.4290	-0.0159
Joint T/P	**+0.0164**	+3.12	**+13.97**	**-26.81**	**-0.4492**	-0.0060

Complementarity of State Structure and VoI. VoI improves NR for every state representation, with the largest gain under the joint state ($+0.0164$). The joint setting also yields the largest improvement in Route/No-Ask accuracy and the largest reductions in UAR and Avg. Q. These results indicate that structured task–profile modeling and decision-aware information valuation are complementary: the joint state identifies which unresolved factors can affect the preferred route, while VoI determines whether resolving them is worth the interaction cost.

E.3 Sensitivity to Communication Cost

Utility–Interaction Trade-off. Figure 3(a) shows that all methods reduce clarification as the per-question cost increases. VISTA maintains a favorable utility–interaction trade-off across the sweep, preserving high gold utility while using fewer questions. VISTA-ER generally requires more interaction to achieve similar utility, whereas Generic VoI and SAGE operate at lower utility levels over most of the range.

Robustness to Communication Cost. Figure 3(b) reports Net Reward across communication costs. VISTA achieves the highest NR throughout the evaluated range and degrades more gradually as clarification becomes more expensive. The advantage is especially clear at larger costs, where methods that ask more aggressively incur greater penalties. The main setting $c = 0.05$ lies within this stable regime, indicating that VISTA is not sensitive to a single choice of communication cost.

F Qualitative Case Studies

We present two representative cases to illustrate how VISTA distinguishes between uncertainty that is relevant to the routing decision and uncertainty that can be safely ignored. In the first case, different

Table 17: Evolution of the joint belief in the compound-retrieval case. b_0 is the initial belief; b_1 follows the task-side response selecting ZINC; and b_2 follows the profile-side response selecting `requests and pandas`. State descriptions are abbreviated for readability.

State	Task interpretation	Profile condition	b_0	b_1	b_2
z_{obs}	Broad request without a specific database assumption	Observed Python/package conditions	.1500	.3659	.2027
z_1	ChEMBL target search with inhibitors and activity measurements	Python 3 and `chembl_webresource_client`	.3000	.0732	.0101
z_2	ChEMBL similarity/substructure search with property filtering	Same as z_1	.2000	.0488	.0068
z_3	PubChem property, bioactivity, and similarity retrieval	`PubChemPy`, `requests`	.1500	.0915	.0127
z_4	PubChem substructure search for related compounds	Same as z_3	.0500	.0305	.0042
z_5	ZINC purchasable analogues with structures and supplier metadata	`requests`, `pandas`	.0200	.1707	.3784
z_6	ZINC property/Tanimoto filtering with SMILES/SDF retrieval	Same as z_5	.0200	.1707	.3784
z_7	ChEMBL bioactivity retrieval with tabular analysis	ChEMBL client, optional `pandas`	.0500	.0122	.0017
z_8	PubChem property/similarity retrieval with local caching	PubChem libraries and cache	.0600	.0366	.0051

plausible task–profile states favor different skills, such that clarification can change the preferred route. In the second case, the belief remains non-degenerate, but all plausible states agree on the same preferred skill, making further clarification unnecessary. Together, these cases illustrate the decision-aware clarification and stopping behavior of VISTA.

F.1 Case I: Clarification Changes the Preferred Route

Case Setting. Consider a user requesting BRAF V600E kinase inhibitors, structurally related compounds, activity measurements, molecular structures, and source annotations through Python. The visible profile permits the use of third-party Python packages. The candidate set contains three skills centered on ChEMBL, ZINC, and PubChem.

The request initially admits several plausible interpretations. A bioactivity-oriented interpretation naturally favors ChEMBL, whereas a request for purchasable compounds, supplier information, catalog identifiers, and downloadable molecular structures is better supported by ZINC. PubChem is also plausible for general property and similarity retrieval. The initial observation therefore does not uniquely determine the preferred route.

Belief over Joint Routing States. Let C, Z, and P denote the ChEMBL-, ZINC-, and PubChem-centered skills, respectively. Table 17 summarizes the joint task–profile support and its evolution over the two clarification turns. Initially, most probability mass is assigned to ChEMBL-oriented interpretations, whereas the two ZINC-oriented states receive little support. Consequently, immediate routing under the initial belief favors ChEMBL.

The first clarification resolves the task-side ambiguity concerning the primary compound database. After the user selects ZINC for accessing purchasable compound libraries, probability mass shifts substantially toward ZINC-oriented interpretations. The second clarification then resolves a profile-side execution condition by identifying the preferred Python access mechanism. This response further supports states compatible with the ZINC REST workflow, after which ZINC becomes the preferred route.

Detailed Clarification Process. Table 18 presents the full clarify-or-route sequence, including the selected question, candidate answers, user response, and associated decision values.

At the initial turn, ChEMBL has the highest expected routing utility. However, the EVPI exceeds the communication cost, so the upper-bound test does not rule out asking. The selected question has EVSI 0.0741 and cost 0.05, giving positive NetVOI 0.0241. It targets the primary database, a distinction that directly separates the functional roles of ChEMBL, PubChem, and ZINC.

The user selects ZINC because the desired compounds should come from purchasable compound libraries. This response substantially increases the posterior mass of ZINC-oriented states, but ChEMBL remains the current maximizing route and further decision-relevant uncertainty remains. The next selected question therefore switches to the profile side and asks how the database should be

Table 18: Detailed clarify-or-route process for the compound-retrieval case. Each clarification turn reports the selected question, available response options, user response, and corresponding decision values. Route values are ordered as ChEMBL/ZINC/PubChem.

Turn	Side	Clarification interaction	Decision values
0	Task	**Question:** Which primary database should be used to search for BRAF V600E kinase inhibitors? **Options:** ChEMBL / PubChem / ZINC / Unsure **User response:** ZINC, for accessing purchasable compound libraries.	**Route** $(C/Z/P)$**:** .8385/.1175/.7523 **EVPI:** .1068 **EVSI:** .0741 **NetVOI:** .0241
1	Profile	**Question:** Which Python package is preferred for querying the selected database? **Options:** `chembl_webresource_client` / `PubChemPy` / `requests and pandas` / Unsure **User response:** `requests and pandas` (for ZINC REST API).	**Route** $(C/Z/P)$**:** .7346/.5000/.5643 **EVPI:** .1671 **EVSI:** .0732 **NetVOI:** .0232
2	Route	**Decision:** No further clarification is performed. ZINC has the highest expected routing utility, and the clarification budget has been exhausted. The router therefore selects ZINC.	**Route** $(C/Z/P)$**:** .6389/.7736/.2112 **EVPI:** .1050

accessed from Python. The response selects `requests and pandas` for the ZINC REST API, providing additional evidence that the ZINC-centered execution path is compatible with the user.

After incorporating the second response, ZINC obtains the highest expected routing utility. Although the remaining EVPI is still positive, the clarification budget has been exhausted, and the router therefore commits to ZINC.

Why the Clarifications Matter. The two questions elicit database and access-package preferences that affect the estimated routing values. The first question concerns the intended data source and distinguishes a ChEMBL-oriented bioactivity task from a ZINC-oriented purchasable-compound search. Its response changes the task-side evidence substantially in favor of ZINC.

The second question does not revisit the task. Instead, it tests whether the execution conditions associated with the emerging route are compatible with the user's profile. Selecting `requests and pandas` supports the ZINC REST workflow and further increases the posterior expected value of routing to ZINC. The resulting belief evolution can be summarized as

$$b_0 \xrightarrow[\text{ZINC}]{q_0^{\text{Task}}} b_1 \xrightarrow[\texttt{requests and pandas}]{q_1^{\text{Profile}}} b_2 \to \text{Route}(\text{ZINC}). \tag{53}$$

This sequence illustrates how task and profile clarification are evaluated under the evolving joint belief. The first answer shifts belief toward ZINC, while ChEMBL remains the current maximizing route. The second answer provides additional evidence about execution compatibility, making ZINC the preferred route. The router then terminates because the clarification budget is exhausted.

F.2 Case II: Residual Uncertainty Does Not Require Clarification

Case Setting. Consider a request for an operational blueprint for an online learning community, including its organizational structure, participant roles, community guidelines, a four-week engagement calendar, and a moderation workflow. The candidate skills are `learning-community-builder`, `project-brainstorming`, and `creative-thought-partner`.

Unlike the previous case, the remaining uncertainty concerns only alternative variants of the same underlying community-design task. For example, the community could be organized as a cohort-based professional-learning group or as a broader online-course community. These interpretations differ semantically, but they do not change which candidate skill is most appropriate.

Decision-Invariant Belief. Table 19 shows that the initial belief is not a point mass. Nevertheless, every state with nonzero probability favors `learning-community-builder`. The uncertainty therefore concerns the precise interpretation of the task, but not the maximizing routing action.

Table 19: Initial belief for the learning-community design case. Multiple interpretations retain nonzero probability, but all induce the same preferred skill.

State	Task interpretation	b_0	Preferred skill
z_1	Complete operational design of an online learning community	.70	Community builder
z_2	Cohort-based professional-learning community	.15	Community builder
z_7	Global online-course community	.15	Community builder
Other	Project brainstorming or open-ended creative dialogue	.00	—

The corresponding expected route values are 1.0000, 0.3000, and 0.1000 for `learning-community-builder`, `project-brainstorming`, and `creative-thought-partner`, respectively. Although the belief remains non-degenerate, perfect information cannot improve the routing decision because every plausible state selects the same skill. Consequently,

$$\mathrm{EVPI}(b_0) = \mathbb{E}_{z\sim b_0}\left[\max_{s_j\in\mathcal{S}} U(s_j, z)\right] - V_{\mathrm{route}}(b_0) = 0 \le c. \tag{54}$$

The EVPI gate therefore terminates the interaction before any question-specific EVSI computation. No question-specific likelihood or EVSI evaluation is needed, and the router directly selects `learning-community-builder`.

Why No Clarification Is Needed. This case highlights the distinction between *belief uncertainty* and *decision-relevant uncertainty*. The router remains uncertain about the precise subtype of the requested learning community, but resolving this uncertainty cannot change the preferred skill.

In particular, an entropy-based criterion could still regard the posterior as uncertain because probability mass is distributed across several interpretations. For routing, however, these distinctions are immaterial: all plausible states induce the same maximizing action. Additional clarification would therefore make the belief more concentrated without improving the downstream routing decision. The EVPI gate captures this invariance directly and allows the router to stop before incurring unnecessary interaction.

F.3 Comparison and Implications

Table 20 summarizes the contrast between the two cases. The critical difference is not whether uncertainty remains, but whether resolving that uncertainty can alter the preferred skill. In Case I, the router asks two positive-NetVOI questions and terminates at the clarification budget; in Case II, the EVPI gate selects immediate routing before any question-specific EVSI evaluation.

Table 20: Comparison of the two qualitative cases. Clarification is selected when an available question has positive estimated net routing value.

Property	Compound retrieval	Learning-community design
Remaining uncertainty	Database choice and execution conditions	Community-design subtype
Can uncertainty change the preferred skill?	Yes	No
Initial EVPI	.1068	.0000
First action	Ask task-side database question	Route immediately
First response	ZINC for purchasable compound libraries	—
Second action	Ask profile-side Python-package question	—
Second response	`requests and pandas` for the ZINC REST API	—
Interaction policy	Ask Task → Ask Profile → Route	Route
Number of questions	2	0
Final route	ZINC	Community builder

Together, the two cases illustrate the central decision principle of VISTA. Clarification is warranted when unresolved task or profile information can alter the preferred skill and an available question has positive decision value. Conversely, residual uncertainty alone is insufficient to justify further interaction. When all plausible states induce the same maximizing skill, additional information may reduce uncertainty without changing the routing decision, and immediate routing is therefore preferred.