TASK-CONDITIONED QUALITY LEARNING FOR HETEROGENEOUS AND UNRELIABLE MULTI-AGENT SYSTEMS

Anonymous authors
Paper under double-blind review

ABSTRACT

We study a heterogeneous multi-agent setting in which any open-world intelligent entity, including AI models, human-operated components, and hybrid systems, can be represented as a callable intelligence node. Such a setting expands the boundary of single-form intelligence, but it is difficult to orchestrate because the internal behavior and ability of nodes are usually not observable, making public descriptions an uncertain guide to actual execution quality. Deceptive profiles and collusive feedback can further distort task allocation. We propose HICS, a Heterogeneous Intelligence Coordination System that serves as an external routing and quality-control framework for such open node pools. HICS adopts a task encoder that learns the task geometry online from previously available feedback and maintains task-conditioned rolling quality expectations, complemented by a direct contextual estimate, to identify which nodes are likely to respond reliably under the current task condition. To facilitate learning, HICS supports separate answering and evaluation roles to obtain denser weak supervision, with reference feedback and audit anchors for calibration. We construct two complementary evaluation settings: a MedAgentBoard-based semi-synthetic medical benchmark for heterogeneous routing under partial outcome feedback, and controlled simulations that examine misleading profiles, hidden specialization, and collusive behavior. Comparisons and ablation studies assess routing quality, quality-model components, and the contribution of weak feedback. The results demonstrate the effectiveness and limits of task-conditioned reliability learning in the evaluated heterogeneous node pools.

1 INTRODUCTION

AI agents increasingly plan, call tools, reflect on intermediate results, and act over long horizons (Wooldridge & Jennings, 1995; Yao et al., 2023; Anthropic, n.d.; OpenAI, 2025). We envision that many future workflows are no longer single-model systems. Instead, they combine general-purpose LLMs and agents, human workers, and human–AI hybrid systems. These heterogeneous nodes are complementary in terms of intelligence. Machines offer scalable computation, retrieval, code generation, and repeatable tool use; human nodes contribute situated observation, tacit domain judgment, preference interpretation, and responsibility-bearing decisions where full automation is inappropriate. The central systems question is therefore not only how to improve one form of intelligence, but how to learn which nodes produce high-quality outcomes for each task condition and route tasks accordingly.

This requires more than assigning static roles or inserting a fixed fallback step. Open multi-agent systems must operate with nodes that are heterogeneous, partially observable, and sometimes strategic. Nodes may exert different effort levels, specialize in different task types, and present public descriptions that do not match their true reliability. The system must decide which nodes to invoke, how to evaluate outputs, and how to update routing state under noisy and sparse feedback. These are coordination and quality-control problems, not merely prompting or interface problems.

The central difficulty is that the quantity worth routing on is not directly observable. A node can look perfectly matched on its public profile yet answer perfunctorily or lack the underlying capability. This mismatch can arise from overconfidence or an incomplete description, insufficient effort despite

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.

adequate expertise, or deliberate misrepresentation. Conversely, a node with an unremarkable profile may be highly dependable in a narrow region. Static role labels, global reputation, and one-shot majority evaluation therefore leave the system underinformed. A coordination layer should learn a task-conditioned quality estimate from interaction history, even under scarce ground-truth signals.

We propose HICS, a Heterogeneous Intelligence Coordination System that operates as an external routing and quality-control layer. A task encoder first maps observable task content into a geometry in which nearby tasks should transfer evidence about node quality. Over this geometry, HICS maintains a rolling task-conditioned quality expectation for each node by aggregating its recent feedback. A direct contextual estimate summarizes task–quality relationships over a longer history and complements this local evidence. The combined estimate guides subsequent routing with exploration. Public profile fit remains an optional input to the framework; the evaluated core learns routing preferences from observed outcomes.

Because direct outcome labels are sparse, HICS also supports disjoint answering and evaluation roles to provide additional feedback. Answer nodes produce candidate responses; evaluation nodes provide weak quality signals whose reliability is learned separately from answering ability. Occasional reference feedback and labeled audit anchors provide checks on those judgments. Such anchors are intended as implicit honeypots when their identities can be concealed. Only feedback available after routing is used for later updates, so neither the quality state nor the task encoder can use the current outcome to influence its own route.

To our knowledge, this problem setting is underemphasized in recent LLM multi-agent work, which has focused more heavily on improving the interaction protocol among a mostly fixed set of participating agents. Our contributions are:

- We formulate heterogeneous and unreliable multi-agent coordination as task-conditioned quality learning, distinguishing public task fit from realized quality under hidden effort and ability, and highlighting the need to learn reliability across repeated interactions.
- We introduce HICS as a generic routing and quality-control layer that combines a learned task geometry, node-local rolling quality expectations, and a direct contextual estimate, with separate answering and evaluation states and audit anchors for weak feedback.
- We evaluate the mechanism both in MedAgentBoard-based heterogeneous medical routing and in controlled simulations, using complementary settings to assess routing quality under partial feedback and unreliable node behavior. Comparisons with baseline methods and component ablations examine the effectiveness of quality learning and the contribution of weak supervision.

2 Related Work

Much recent LLM-agent work studies how participating agents should collaborate to solve a given task instance. This includes prompting-based reasoning and acting (Yao et al., 2023; Wang et al., 2023), model-based judging and debate (Zheng et al., 2023; Du et al., 2024), conversation and role-based frameworks such as AutoGen and Mixture-of-Agents (Wu et al., 2024; Wang et al., 2025), and newer systems that improve planning, reflection, and long-context decomposition (Li et al., 2025; Bo et al., 2024; Zhang et al., 2024). These papers mostly optimize the internal collaboration protocol: how agents discuss, decompose, plan, or aggregate within a workflow. Agent-oriented planning also allocates subtasks, and MasRouter jointly selects collaboration modes, agent roles, and models (Li et al., 2025; Yue et al., 2025). HICS studies an external coordination layer that learns which heterogeneous nodes to invoke across repeated tasks when their public profiles can be misleading and their realized quality is only partially observable.

A closer line of work learns which language model to invoke for a query. MetaLLM uses bandit-based selection, RouterDC learns query–model representations, and CARROT estimates local performance and cost (Nguyen et al., 2024; Chen et al., 2024; Somerstep et al., 2025). PILOT combines a preference-trained prior with online adaptation, while Causal LLM Routing learns from selective observational logs (Panda et al., 2025; Tsiourvas et al., 2025). These methods establish task-dependent model selection as an important problem; HICS emphasizes execution reliability under misleading descriptions and the reliability of the feedback used to learn it. Informational and strategic frictions are

also studied directly: iAgents addresses collaboration under information asymmetry (Liu et al., 2024), and safety work investigates covert collusion through steganographic communication (Motwani et al., 2024). HICS focuses on the routing consequences of unreliable outputs and judgments, using separate executor and evaluator quality states and external checks to guide later allocations in controlled correlated-error and noisy-evaluator settings. Differences in required calibration data, workflow construction, and feedback acquisition also matter for baseline comparisons (Appendix C.1).

HICS is also connected to reputation systems (Resnick et al., 2000), contextual bandits (Li et al., 2010; Valko et al., 2013; Zhou et al., 2020), peer prediction and limited-ground-truth evaluation incentives (Miller et al., 2005; Gao et al., 2019). Human computation, crowdsourcing, and hybrid intelligence likewise treat capability as externally coordinatable (Kamar et al., 2012; Dellermann et al., 2019; Zeng et al., 2024); bandit-based task assignment and human–AI collaboration already address heterogeneous expertise with incomplete feedback (Zhang et al., 2015; Gao et al., 2021). Dawid–Skene estimation and reliable crowdsourcing infer quality from noisy participants (Dawid & Skene, 1979; Karger et al., 2011). More recently, Smoothie estimates sample-dependent model quality from unlabeled model outputs, and limited-feedback LLM routing studies selective verification under a feedback budget (Guha et al., 2024; Wei et al., 2026). HICS builds on these foundations by combining learned task geometry, node-local rolling expectations, and a direct contextual estimate with separate executor and evaluator reliability states. This combination supports repeated routing while keeping weak judgments and verified outcomes distinct in the learning process.

3 HICS Design

Task and objective. At round t, the coordinator receives task condition x_t and selects k answerers from a pool $\mathcal{N}$ with public descriptions u_i. It aggregates their responses and uses subsequently available feedback to improve cumulative answer quality under this answer-invocation budget. Routing has access only to public task information and previously available feedback.

HICS addresses description–execution mismatch by learning task-conditioned quality from interaction history. A learned task geometry identifies relevant past evidence, while a direct contextual predictor complements the local estimate. To learn under sparse supervision, HICS accepts weak feedback with explicit confidence weights and maintains evaluator reliability separately from answering ability. These components form a route–execute–aggregate–update loop (Figure 1); Appendix A specifies their experimental instantiation.

3.1 Task-conditioned quality expectations

Let $b = g(x)$ be a frozen representation of the observable task condition. HICS learns $Q_i(b)$ as a proxy for latent expected execution quality $R_i(x)$, reflecting both ability and effort. Node i retains its latest W available records $(b_s, \omega_{i,s}, c_{i,s})$ in $D_{i,W}$: task representation, quality signal, and learning weight, with $\omega_{i,s}, c_{i,s} \in [0, 1]$ (Section 3.4).

HICS first summarizes feedback from recent, relevant tasks, using a nonnegative learned similarity K_ϕ. A direct predictor $m_i(b) \in [0, 1]$ complements this local evidence by fitting task–quality relationships over a longer history. The resulting quality estimate is

$$\begin{aligned} \bar{\omega}_i(b) &= \frac{\beta Q_0 + \sum_{s \in D_{i,W}} c_{i,s} K_\phi(b, b_s) \omega_{i,s}}{\beta + \sum_{s \in D_{i,W}} c_{i,s} K_\phi(b, b_s)}, \\ Q_i(b) &= (1 - \eta)\bar{\omega}_i(b) + \eta m_i(b). \end{aligned} \tag{1}$$

Here Q_0 is the initial quality prior, $\beta > 0$ its strength, and $\eta \in [0, 1]$ the balance between the two estimates. With no relevant local evidence, $\bar{\omega}_i$ returns Q_0. Feedback confidence controls how much each observation contributes, separately from its relevance to the current task.

Local averaging transfers recent outcomes through learned task neighborhoods; direct prediction summarizes feature relationships over a longer history. We instantiate the latter with confidence-weighted ridge regression on all available feedback for each node. Both estimates use that node's own outcomes, while the shared geometry learns where evidence transfers across tasks. Setting $\eta = 0$ recovers local estimation; setting $\eta = 1$ retains only direct prediction.

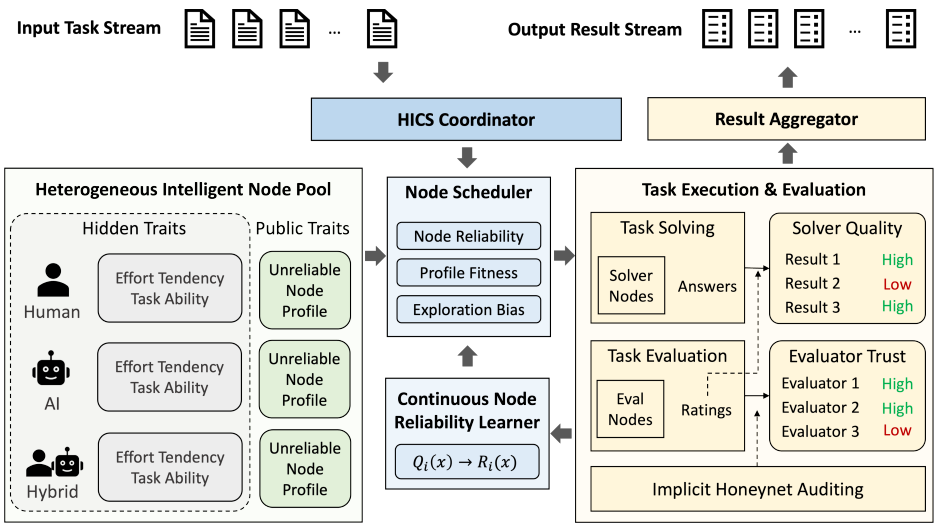

Figure 1: HICS framework. The arrow $Q_i(x) \to R_i(x)$ denotes learning a proxy for latent expected execution quality. Subsequent feedback updates executor quality and separate evaluator reliability. Profile fit is optional, and evaluator and audit mechanisms depend on the protocol. Human and hybrid nodes illustrate the intended scope; the experiments use controlled node pools.

3.2 Learning the Task Geometry

Semantic similarity need not imply similar execution quality. HICS therefore learns the similarity used in Eq. (1) from quality-prediction error. An encoder f_ϕ maps task representations into a shared space, with $K_\phi(b, b') = \kappa(f_\phi(b), f_\phi(b'))$ for a nonnegative similarity function κ. Our implementation uses a residual encoder and thresholded cosine similarity; their specific forms appear in Appendix A.

For each feedback event e, HICS saves the node's local history H_e at routing time. Let $\bar{\omega}_{\phi,i_e}(b_e; H_e)$ denote the local mean in Eq. (1), evaluated on that history with the current encoder. Given subsequent feedback ω_e and confidence c_e, the encoder minimizes

$$\mathcal{L}(\phi) = \frac{\sum_{e \in \mathcal{B}} c_e a_e \big(\bar{\omega}_{\phi,i_e}(b_e; H_e) - \omega_e\big)^2}{\sum_{e \in \mathcal{B}} c_e a_e}. \tag{2}$$

Here $\mathcal{B}$ contains eligible events with usable prior history, and a_e is an optional sampling weight specified in Appendix A. The prediction error trains similarity to transfer evidence useful for estimating node quality. The direct predictor is fitted separately. Current feedback never enters its own history snapshot, and model updates affect only subsequent decisions.

3.3 Quality-Aware Routing and Answer Aggregation

Routing score. HICS combines learned execution quality with public-profile fit and an evidence bonus:

$$\widehat{G}_i(x_t) = \lambda_Q Q_i(b_t) + \lambda_V v_i(x_t) + \lambda_U U_i(b_t). \tag{3}$$

Here $v_i(x_t)$ measures the match between the node's public description and the task. The bonus $U_i(b_t)$ favors nodes with sparse local evidence. The nonnegative coefficients sum to one and determine how strongly allocation relies on each signal. This separates description-based matching from quality learned through execution.

Selection and aggregation. The coordinator selects k answerers without replacement using ϵ-greedy sampling on $\widehat{G}_i$, with uniform tie-breaking. Random exploration gives nodes opportunities to accumulate evidence beyond those favored by their current scores. For discrete answers, HICS sums the quality-weighted votes supporting each answer and returns the answer with the greatest total weight. Simulation aggregation also uses the answerer's reported confidence; Appendix A gives the exact rule. With a single answerer, its response is returned directly. Optional evaluators are selected

outside the answer set using their separate reliability state. Their judgments inform future learning, while the current answer is committed independently of those judgments.

3.4 Weak quality signals and role-separated updates

Sparse outcome labels leave many executions without direct supervision. HICS represents each available feedback observation by a quality target ω and a learning weight c. This interface allows different feedback sources to update the same executor model while expressing how much they should be trusted.

Verified and weak feedback. A verified outcome supplies observed quality with unit weight. When verification is unavailable, peer agreement or an independent evaluator's score can supply a weak target. Peer agreement measures how often other selected answerers return the same response; it receives a lower weight because correlated errors can produce misleading consensus. An evaluator judges candidate answers, with each score weighted by the evaluator's previously estimated reliability. Verified feedback takes precedence when both sources are available for an execution. If no usable feedback exists, the model receives no update.

Separate evaluator reliability. Answering ability and evaluation competence need not coincide. HICS therefore maintains an evaluator state distinct from the executor's local memory and direct predictor. Verified candidate outcomes assess the evaluator's earlier judgments and update the weight assigned to later feedback. Labeled audit anchors provide an additional source of such checks when ordinary verification is scarce. They can serve as honeypots when their identities are concealed. In our instantiation, reliability balances correct acceptance and correct rejection, then determines the weight of the original evaluation score. Appendix A specifies this rule and the anchor protocol.

The common principle is to learn from additional judgments while retaining an external basis for assessing their reliability. The feedback source, verification budget, and weighting policy can vary with the setting. Our experiments distinguish the quality model from these supervision choices: the primary medical experiment uses verified outcomes, while the main simulation includes independent evaluators and anchors. A supplementary medical protocol examines peer agreement using cached answers. The corresponding ablations examine quality estimation, learned geometry, and the contribution of additional feedback.

4 Evaluation

We evaluate three questions: can execution history recover hidden specialization, can additional judgments improve allocation under unreliable behavior, and which components account for performance? Medical routing isolates specialization learning as verified feedback becomes scarce. Simulation adds misleading profiles, variable effort, correlated errors, and independent evaluation. Learning curves and ablations connect these outcomes to adaptation, quality estimation, and feedback use.

Comparison families. We compare nonlearning allocation, global reputation, nearest-neighbor prediction, contextual bandits, agreement, and truth estimation to distinguish execution evidence, task conditioning, and imperfect feedback. Each protocol includes compatible methods and a privileged Oracle using private node information. Appendix C specifies adaptations and the reported comparator subset.

4.1 Learning specialization in medical routing

Purpose and construction. We use MedQA, PubMedQA, PathVQA, and VQA-RAD from MedAgentBoard (Zhu et al., 2025) to test specialization learning across medical question types and modalities. Twelve prompted GPT-5.5 configurations share a general clinical public profile but differ in private specialization, making execution history informative when descriptions are not. A fixed private-profile fit threshold gates each cached response; below it, the environment supplies a controlled incorrect answer. Reference labels implement this semi-synthetic mismatch, while routing receives only permitted feedback.

Workload and feedback. Each dataset contributes 200 unique questions in two separately shuffled passes: 1600 events measuring adaptation to recurring work, not unseen-question generalization. Each event invokes one answerer; Full, Half, and Sparse GT reveal 100%, 50%, and 10% of outcomes. HICS uses trainable geometry, local and direct estimation, and exploration, with profile scoring disabled. The selected answer is returned directly; this protocol has no peer targets, evaluators, or extra audits. The single- and three-answer medical protocols use five algorithm seeds on fixed cache, task order, and GT masks, measuring algorithm variability rather than dataset or generation uncertainty (Appendix B).

Comparators and results. Table 1 compares single-answer routing. Across five seeds, HICS averages 75.46%, 71.53%, and 66.03% under Full, Half, and Sparse GT. Task-conditioned methods generally outperform global reputation, supporting specialization learning when profiles are uninformative. HICS has the highest aggregate nonprivileged mean among the methods in Table 1 under Full and Sparse GT; the Sparse margin is small, and NeuralUCB-Diag leads under Half GT, where HICS also varies more across seeds. Appendix D reports replay diagnostics; a separate calibration study favors offline-trained routers (Appendix D.2).

Table 1: Single-answer medical accuracy (%; five-seed mean ± sample SD). Each GT column covers all 1600 events with equal dataset weights. Mean across GT summarizes within-seed condition averages, not a new feedback setting. Bold marks the best reported nonprivileged mean; Oracle uses private profiles.

Method	Mean across GT	100% GT	50% GT	10% GT
Random	56.94 ± 0.74	56.94 ± 0.74	56.94 ± 0.74	56.94 ± 0.74
Visible-Fit	57.73 ± 0.63	57.73 ± 0.63	57.73 ± 0.63	57.73 ± 0.63
Global Reputation	61.94 ± 2.29	62.23 ± 3.07	61.88 ± 2.37	61.71 ± 3.77
Online KNN	66.85 ± 0.87	69.79 ± 0.64	67.91 ± 1.04	62.84 ± 2.09
LinUCB	69.92 ± 0.64	73.40 ± 1.20	70.63 ± 1.12	65.73 ± 1.36
KernelUCB	69.72 ± 0.85	73.93 ± 2.40	70.60 ± 1.62	64.63 ± 2.29
NeuralUCB-Diag	70.93 ± 0.61	74.80 ± 0.53	$\mathbf{72.19} \pm 0.61$	65.79 ± 1.26
NeuralLinearTS	63.11 ± 0.75	68.08 ± 0.99	63.23 ± 0.95	58.03 ± 0.83
HICS	$\mathbf{71.00} \pm 1.13$	$\mathbf{75.46} \pm 0.63$	71.53 ± 2.28	$\mathbf{66.03} \pm 1.15$
Oracle (privileged)	79.38 ± 0.00	79.38 ± 0.00	79.38 ± 0.00	79.38 ± 0.00

Agreement and truth-estimation comparison. Three-answer routing at 10% GT enables agreement and truth estimation using the same cache, 4800 answer invocations per method, and uniform majority vote. Table 2 compares these families with Online KNN and the strongest contextual bandit. HICS with peer feedback averages 75.18%, versus 72.15% without it and 71.80% for NeuralUCB-Diag, which it exceeds in all five seeds. The within-HICS gain averages 3.03 points, positive in four of five seeds at the same answer and GT budget. This protocol adds peer targets without independent medical evaluation.

4.2 Routing under heterogeneous and unreliable behavior

Purpose and construction. Simulation tests allocation beyond medical specialization, including misleading profiles, hidden effort, and coordinated wrong answers. Three frozen configurations each contain 48 nodes and 2400 tasks across six task families. Nodes vary in specialization and effort; 25% coordinate on a wrong answer while exposing attractive profiles. Methods share the task order, node pool, and potential task–node outcomes, selecting three answerers per task.

Supervision and aggregation. Direct GT covers approximately 8% of tasks; random audits reveal 25% of the remainder, giving about 31% verified exposure. HICS adds one independent evaluator outside the answer set per task and three balanced anchor-pair evaluations every 30 tasks. Ordinary verified outcomes and anchors calibrate evaluator reliability; weak judgments update executor quality only for future decisions. They never affect the current vote. All methods use 7200 answer invocations

Table 2: Three-answer medical accuracy (%; five-seed mean ± sample SD across all four datasets), evaluated only at 10% GT. Each run uses 4800 answer invocations and uniform majority vote. Bold marks the best nonprivileged mean among the displayed configurations.

Method	Overall (10% GT)
Random	60.99 ± 0.43
Global Reputation	66.50 ± 1.44
Agreement + Spot	65.73 ± 3.53
Online DS	67.91 ± 2.48
Online KNN	67.94 ± 1.26
NeuralUCB-Diag	71.80 ± 0.95
HICS without peer	72.15 ± 1.46
HICS with peer	$\mathbf{75.18} \pm 1.11$
Oracle (privileged)	79.23 ± 0.14

per configuration; HICS uses 2640 additional evaluation requests. This compares complete protocols with disclosed costs; the ablations isolate feedback use within HICS.

The common vote weights answers by predicted quality and reported confidence. Confidence equals the simulator's correctness probability after effort and difficulty adjustments, a strong calibration assumption. Main-protocol evaluators have above-chance accuracy independent of answer-role collusion. These choices define a controlled allocation test; they do not establish general answer inference or robustness to strategic evaluator collusion. Appendix B gives the generator and feedback schedules.

Comparators and results. Table 3 covers nonlearning, reputation, nearest-neighbor, agreement, truth-estimation, and bandit families. Agreement + Spot learns from peer consistency with verified corrections; Online DS adapts Dawid–Skene estimation to past overlapping responses. Verified-only baselines receive no independent judgments. We report full-stream and last-400-task accuracy as mean ± sample SD across three configurations.

Table 3: Simulation accuracy (%). Task-family columns show full-stream means; All 2400 and Last 400 show mean ± sample SD. Req.: requirement discovery. Bold marks the best reported nonprivileged method. Oracle uses private attributes; HICS uses 2640 extra evaluation requests per configuration.

Method	Code	Summary	Fact	Aesthetic	Field	Req.	All 2400	Last 400
Random	43.58	40.01	48.77	42.00	40.99	40.44	42.62 ± 2.13	44.00 ± 3.50
Visible-Fit	38.68	33.71	37.50	38.63	31.94	46.10	37.81 ± 2.27	39.92 ± 3.91
Global Reputation	53.62	50.38	67.92	55.09	55.92	63.64	57.82 ± 13.92	59.75 ± 16.39
Agreement + Spot	54.90	52.05	71.88	58.34	56.28	63.11	59.40 ± 17.52	59.58 ± 17.55
Online DS	57.03	65.23	60.35	55.92	64.38	56.48	59.93 ± 11.58	62.75 ± 12.67
Online KNN	57.99	52.92	66.73	65.43	53.21	63.76	60.03 ± 7.30	65.08 ± 6.35
UCB	52.10	45.24	53.74	44.88	47.38	46.55	48.26 ± 7.25	51.67 ± 13.54
LinUCB	61.81	60.75	70.58	60.27	62.52	68.31	64.04 ± 4.93	73.17 ± 4.30
KernelUCB	60.18	58.44	71.98	58.57	63.95	63.07	62.74 ± 7.08	69.58 ± 5.30
NeuralUCB-Diag	53.48	49.92	60.71	49.30	51.85	55.27	53.46 ± 4.14	58.50 ± 5.06
NeuralLinearTS	43.25	41.09	50.93	42.80	45.57	45.78	44.88 ± 2.25	49.33 ± 1.70
HICS	**69.26**	**74.29**	**76.21**	**70.29**	**78.30**	**70.76**	$\mathbf{73.17} \pm 1.28$	$\mathbf{80.92} \pm 0.38$
Oracle	84.75	85.01	86.29	84.05	89.09	85.82	85.83 ± 2.10	85.25 ± 2.88

HICS reaches 73.17% full-stream and 80.92% final-window accuracy, exceeding the strongest reported nonprivileged comparator in Table 3, LinUCB, by 9.13 and 7.75 points. This margin combines quality learning with additional supervision. Visible-Fit finishes below Random, whereas the privileged Oracle reaches 85.25%, illustrating the gap between public descriptions and private execution attributes. With the same verified-feedback and answer budgets as LinUCB, HICS without evaluators reaches 66.82% and 75.67%, compared with 64.04% and 73.17%. Appendix E compares all three HICS feedback regimes.

4.3 Adaptation over repeated interactions

Figure 2 compares learning trajectories across method families using all three simulation configurations and all six task families. HICS improves with interaction and separates from the strongest contextual comparator later in the stream. Reputation, agreement, and truth-estimation controls show adaptation without HICS's task-conditioned quality model. These are interaction-based learning curves; they do not measure computational speed or equal total invocation cost.

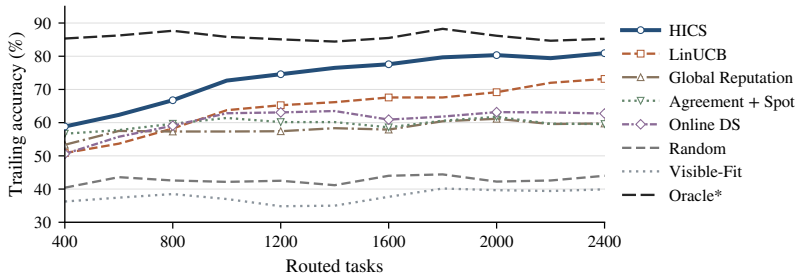

Figure 2: Simulation adaptation. Mean trailing-400-task accuracy over three configurations, sampled every 200 tasks without smoothing or omitted runs. The vertical axis starts at 30%. HICS includes its extra evaluation budget; Oracle* is privileged. Overlapping windows are not independent replicates. Individual trajectories appear in Appendix E.1.

4.4 Which components contribute?

Table 4 isolates components in simulation, where independent evaluation is active. Paired losses use complete main-table HICS as reference: positive values indicate degradation, negative values improvement.

Table 4: Simulation ablation losses (percentage points): Full HICS accuracy minus ablated accuracy, averaged over three paired configurations. Disabling evaluator weak updates retains all evaluation requests.

Ablation	All tasks	Last 400
w/o geometry updates	+0.2	+0.1
w/o direct branch	+3.3	+7.8
Direct branch only	+0.5	+0.5
w/o evaluator weak updates	+6.3	+5.3

Quality estimation and geometry. In Table 4, removing the direct branch costs 7.8 points in final-window simulation accuracy; retaining only direct prediction loses 0.5 points. Freezing geometry changes mean accuracy by at most 0.2 points.

Additional feedback. Disabling evaluator weak updates costs 6.3 full-stream and 5.3 final-window points while retaining the same request budget. The additional judgments therefore improve learning within this protocol. Fixed weighting and omitting anchor calibration yield slightly higher final-window means, so the evidence does not establish that every calibration component is necessary. Appendix E.2 reports the full evaluator controls, including these negative ablations and their update-mass differences.

Sparse feedback with collusive evaluators. An exploratory stress study extends the same three simulation configurations to scarce verification and dishonest judges (Appendix E.3). With approximately 8% verified tasks and 25% collusive judges, reliability-aware selection exceeds uniform selection by 20.75 full-stream and 21.83 final-window points (Table 5). Both use fixed weak weight

0.55 and identical answer, evaluation, and weak-update budgets. This supports learning whom to ask for feedback under this attack; anchor and adaptive-weight benefits remain setting-dependent.

Table 5: Evaluator selection under sparse verification and collusion. Accuracy (%; mean ± sample SD over three configurations). Both controls retain anchors and use 7200 answer invocations and 2640 evaluation requests.

Evaluator selection (fixed weight)	All 2400	Last 400
Reliability-aware	71.96 ± 2.92	80.17 ± 3.89
Uniform	51.21 ± 4.95	58.33 ± 5.58

5 Limitations and Discussion

Coordination as learning about execution. HICS treats reliability as conditional on the task and observed behavior. This perspective is relevant to future communities of AI agents, human contributors, and hybrid systems, whose descriptions cannot fully specify their performance. Such communities require coordination across complementary capabilities while preserving the distinction between producing an answer and judging it. Our contribution is an operational framework for studying these linked decisions, with preliminary evidence from controlled settings.

What the evidence establishes. The experiments support task-conditioned quality learning and the use of additional judgments, but give unequal support to individual components. Direct contextual prediction and evaluator selection under the tested collusive attack contribute substantially; learned geometry and anchor calibration have smaller or setting-dependent effects. Fixed task representations and verified evaluator checks may already capture much of the structure in these settings. The simulation advantage includes extra evaluation resources; verified-only and fixed-weight controls separate their effects. Historical calibration also outperforms the evaluated cold-start configuration (Appendix D.2).

From controlled pools to mixed communities. Our settings isolate description–execution mismatch and feedback access rather than reproduce a functioning human–AI community. Medical nodes share one backend and use induced specialization on a repeated workload; simulation specifies effort, errors, and evaluation competence. The label encoding and idealized confidence further restrict the simulation's interpretation (Appendix B). Neither setting tests actual human participation, transfer to new domains, or adaptive evaluator collusion. These remain substantive validation needs, not properties established by the present results.

Real communities also change as participants learn, disengage, or alter their incentives. Rolling evidence alone does not resolve drift in rarely selected nodes, and reliability estimates cannot substitute for accountable human oversight. A next step is evaluation in evolving mixed pools with natural judgments, verification costs, and delayed outcomes. This would test when task-dependent reliability improves collaboration and how much supervision it requires.

6 Conclusion

We introduced HICS, an external coordination framework that learns task-conditioned execution quality when public descriptions and realized performance can diverge. It combines local historical evidence, learned task geometry, and direct contextual prediction, with separate evaluator reliability for additional weak supervision. Medical routing and controlled simulation provide complementary evidence for quality learning and feedback use, while identifying components whose benefits depend on the setting. HICS offers a starting point for studying how heterogeneous communities can allocate work through accumulated evidence about performance. Establishing this principle in real human–AI communities requires broader behavioral and resource-aware validation.

Reproducibility Statement

Appendix A specifies model updates, hyperparameters, aggregation, and feedback timing. Appendix B describes the frozen inputs and supervision protocols; Appendix C defines baseline adaptations and training-only selection. Additional medical and simulation analyses appear in Appendices D and E. Reported tables and vector figures are computed from recorded decisions and reference labels using retained analysis scripts. Exact medical replay additionally requires the fixed response cache.

Ethics Statement

The medical experiments use retrospective benchmark questions with induced node heterogeneity. They do not involve clinical deployment or establish suitability for patient care. Human participation and effort are not measured. In a deployed community, task-conditioned reliability estimates should inform allocation without becoming unqualified judgments about individuals. Their use would require appropriate consent, privacy protections, human oversight, and ways to challenge incorrect assessments.

AI Use Statement

Generative AI assisted with experimental scripts and implementation, manuscript drafting, and language editing, as well as experimental-protocol review, result interpretation, and code and literature inspection. All content was manually reviewed by the authors, who take responsibility for the final text, claims, code, and results, including all AI-assisted content.

References

Anthropic. Claude code. Product page, n.d. URL https://claude.com/product/claude-code. Accessed: 2026-09-26.

Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2–3):235–256, 2002. doi: 10.1023/A:1013689704352.

Xiaohe Bo, Zeyu Zhang, Quanyu Dai, Xueyang Feng, Lei Wang, Rui Li, Xu Chen, and Ji-Rong Wen. Reflective multi-agent collaboration based on large language models. In *Advances in Neural Information Processing Systems*, volume 37, pp. 138595–138631, 2024. doi: 10.52202/079017-4397. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/fa54b0edce5eef0bb07654e8ee800cb4-Abstract-Conference.html.

Shuhao Chen, Weisen Jiang, Baijiong Lin, James T. Kwok, and Yu Zhang. RouterDC: Query-Based Router by Dual Contrastive Learning for Assembling Large Language Models. In *Advances in Neural Information Processing Systems*, volume 37, pp. 66305–66328, 2024. doi: 10.52202/079017-2120. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/7a641b8ec86162fc875fb9f6456a542f-Abstract-Conference.html.

A. P. Dawid and A. M. Skene. Maximum likelihood estimation of observer error-rates using the EM algorithm. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 28(1):20–28, 1979. doi: 10.2307/2346806.

Dominik Dellermann, Philipp Ebel, Matthias Söllner, and Jan Marco Leimeister. Hybrid intelligence. *Business & Information Systems Engineering*, 61(5):637–643, 2019. doi: 10.1007/s12599-019-00595-2.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. Improving factuality and reasoning in language models through multiagent debate. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 11733–11763, 2024. URL https://proceedings.mlr.press/v235/du24e.html.

Ruijiang Gao, Maytal Saar-Tsechansky, Maria De-Arteaga, Ligong Han, Min Kyung Lee, and Matthew Lease. Human-AI Collaboration with Bandit Feedback. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*, pp. 1722–1728, 2021. doi: 10.24963/ijcai.2021/237. URL https://www.ijcai.org/proceedings/2021/237.

Xi Alice Gao, James R. Wright, and Kevin Leyton-Brown. Incentivizing evaluation with peer prediction and limited access to ground truth. *Artificial Intelligence*, 275:618–638, 2019. doi: 10.1016/j.artint.2019.03.004.

Neel Guha, Mayee F. Chen, Trevor Chow, Ishan S. Khare, and Christopher Ré. Smoothie: Label Free Language Model Routing. In *Advances in Neural Information Processing Systems*, volume 37, pp. 127645–127672, 2024. doi: 10.52202/079017-4053. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/e6b57a990462df5afa58d64ce2709db9-Abstract-Conference.html.

Ece Kamar, Severin Hacker, and Eric Horvitz. Combining human and machine intelligence in large-scale crowdsourcing. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems*, pp. 467–474, 2012. URL https://www.ifaamas.org/Proceedings/aamas2012/papers/3C_3.pdf.

David R. Karger, Sewoong Oh, and Devavrat Shah. Iterative learning for reliable crowdsourcing systems. In *Advances in Neural Information Processing Systems*, volume 24, pp. 1953–1961, 2011.

Ao Li, Yuexiang Xie, Songze Li, Fugee Tsung, Bolin Ding, and Yaliang Li. Agent-oriented planning in multi-agent systems. In *Proceedings of the Thirteenth International Conference on Learning Representations*, 2025.

Lihong Li, Wei Chu, John Langford, and Robert E. Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th International Conference on World Wide Web*, pp. 661–670, 2010. doi: 10.1145/1772690.1772758. URL https://www.schapire.net/papers/www10.pdf.

Wei Liu, Chenxi Wang, Yifei Wang, Zihao Xie, Rennai Qiu, Yufan Dang, Zhuoyun Du, Weize Chen, Cheng Yang, and Chen Qian. Autonomous agents for collaborative task under information asymmetry. In *Advances in Neural Information Processing Systems*, volume 37, pp. 2734–2765, 2024. doi: 10.52202/079017-0090. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/0534abc9e6db91683d82186ef0d68202-Abstract-Conference.html.

Nolan Miller, Paul Resnick, and Richard Zeckhauser. Eliciting informative feedback: The peer-prediction method. *Management Science*, 51(9):1359–1373, 2005. doi: 10.1287/mnsc.1050.0379.

Sumeet Ramesh Motwani, Mikhail Baranchuk, Martin Strohmeier, Vijay Bolina, Philip H. S. Torr, Lewis Hammond, and Christian Schroeder de Witt. Secret collusion among AI agents: Multi-agent deception via steganography. In *Advances in Neural Information Processing Systems*, volume 37, pp. 73439–73486, 2024. doi: 10.52202/079017-2336.

Quang H. Nguyen, Thinh Dao, Duy C. Hoang, Juliette Decugis, Saurav Manchanda, Nitesh V. Chawla, and Khoa D. Doan. MetaLLM: A High-performant and Cost-efficient Dynamic Framework for Wrapping LLMs. arXiv preprint arXiv:2407.10834, 2024. URL https://arxiv.org/abs/2407.10834.

OpenAI. Introducing codex. Product announcement, 2025. URL https://openai.com/index/introducing-codex/. Published May 16, 2025; accessed: 2026-09-26.

Pranoy Panda, Raghav Magazine, Chaitanya Devaguptapu, Sho Takemori, and Vishal Sharma. Adaptive LLM Routing under Budget Constraints. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 23934–23949, 2025. doi: 10.18653/v1/2025.findings-emnlp.1301. URL https://aclanthology.org/2025.findings-emnlp.1301/.

Paul Resnick, Ko Kuwabara, Richard Zeckhauser, and Eric Friedman. Reputation systems. *Communications of the ACM*, 43(12):45–48, 2000. doi: 10.1145/355112.355122.

Carlos Riquelme, George Tucker, and Jasper Snoek. Deep Bayesian Bandits Showdown: An Empirical Comparison of Bayesian Deep Networks for Thompson Sampling. In *Proceedings of the Sixth International Conference on Learning Representations*, 2018. URL https://arxiv.org/abs/1802.09127.

Seamus Somerstep, Felipe Maia Polo, Allysson Flavio Melo de Oliveira, Prattyush Mangal, Mírian Silva, Onkar Bhardwaj, Mikhail Yurochkin, and Subha Maity. CARROT: A Cost Aware Rate Optimal Router. arXiv preprint arXiv:2502.03261, 2025. URL https://arxiv.org/abs/2502.03261.

Asterios Tsiourvas, Wei Sun, and Georgia Perakis. Causal LLM Routing: End-to-End Regret Minimization from Observational Data. In *Advances in Neural Information Processing Systems*, volume 38, pp. 37289–37314, 2025. doi: 10.52202/085713-1252. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/357774d53e5ee21c5f08ba779e3b5dd9-Abstract-Conference.html.

Michal Valko, Nathan Korda, Rémi Munos, Ilias Flaounas, and Nello Cristianini. Finite-Time Analysis of Kernelised Contextual Bandits. In *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence*, pp. 654–663, 2013. URL https://auai.org/uai2013/prints/papers/161.pdf.

Junlin Wang, Jue Wang, Ben Athiwaratkun, Ce Zhang, and James Zou. Mixture-of-agents enhances large language model capabilities. In *Proceedings of the Thirteenth International Conference on Learning Representations*, 2025.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *Proceedings of the Eleventh International Conference on Learning Representations*, 2023.

Wang Wei, Soumyabrata Pal, Koyel Mukherjee, Franck Dernoncourt, Ryan A. Rossi, Branislav Kveton, and Hoda Eldardiry. Online Learning with LLM Experts from Limited Feedback. arXiv preprint arXiv:2609.05820, 2026. URL https://arxiv.org/abs/2609.05820.

Michael Wooldridge and Nicholas R. Jennings. Intelligent agents: Theory and practice. *The Knowledge Engineering Review*, 10(2):115–152, 1995. doi: 10.1017/S0269888900008122.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, Ahmed Awadallah, Ryen W. White, Doug Burger, and Chi Wang. AutoGen: Enabling next-gen LLM applications via multi-agent conversations. In *Proceedings of the First Conference on Language Modeling*, 2024. URL https://www.microsoft.com/en-us/research/wp-content/uploads/2023/08/LLM_agent.pdf.

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

Yanwei Yue, Guibin Zhang, Boyang Liu, Guancheng Wan, Kun Wang, Dawei Cheng, and Yiyan Qi. MasRouter: Learning to Route LLMs for Multi-Agent Systems. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15549–15572, 2025. doi: 10.18653/v1/2025.acl-long.757. URL https://aclanthology.org/2025.acl-long.757/.

Xia Zeng, David La Barbera, Kevin Roitero, Arkaitz Zubiaga, and Stefano Mizzaro. Combining large language models and crowdsourcing for hybrid human–AI misinformation detection. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 2332–2336, 2024. doi: 10.1145/3626772.3657965.

Hao Zhang, Yao Ma, and Masashi Sugiyama. Bandit-Based Task Assignment for Heterogeneous Crowdsourcing. *Neural Computation*, 27(11):2447–2475, 2015. doi: 10.1162/NECO_a_00782. URL https://www.ms.k.u-tokyo.ac.jp/sugi/2015/BBTA.pdf.

Yusen Zhang, Ruoxi Sun, Yanfei Chen, Tomas Pfister, Rui Zhang, and Sercan Ö. Arık. Chain of agents: Large language models collaborating on long-context tasks. In *Advances in Neural Information Processing Systems*, volume 37, pp. 132208–132237, 2024. doi: 10.52202/079017-4202.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging LLM-as-a-judge with MT-Bench and chatbot arena. In *Advances in Neural Information Processing Systems*, volume 36, pp. 46595–46623, 2023.

Dongruo Zhou, Lihong Li, and Quanquan Gu. Neural Contextual Bandits with UCB-based Exploration. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 11492–11502, 2020. URL https://proceedings.mlr.press/v119/zhou20a.html.

Yinghao Zhu, Ziyi He, Haoran Hu, Xiaochen Zheng, Xichen Zhang, Zixiang Wang, Junyi Gao, Liantao Ma, and Lequan Yu. MedAgentBoard: Benchmarking multi-agent collaboration with conventional methods for diverse medical tasks. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-4851. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/d59aa09699530c00d4b875a883876641-Abstract-Datasets_and_Benchmarks_Track.html.

A Implementation Details

A.1 Quality estimation and task geometry

Target and direct prediction. HICS estimates expected execution quality $R_i(x) = \mathbb{E}[Y_{i,t} \mid x_t = x, i \text{ executes}]$, with $Y_{i,t} \in [0,1]$, combining the effects of hidden ability and effort. In Figure 1, $Q_i(x)$ abbreviates $Q_i(g(x))$; the arrow denotes estimation, not guaranteed convergence. Weak targets are proxies and do not make Q_i a calibrated correctness probability. Each positive-weight observation updates the latest-W local memory and the direct model; zero-weight observations are skipped.

For $\widetilde{b} = [b; 1]$, the direct model maintains confidence-weighted ridge statistics over all previously accepted observations for node i:

$$A_i = \lambda I + \sum_s c_{i,s} \widetilde{b}_s \widetilde{b}_s^\top, \qquad h_i = \sum_s c_{i,s}(\omega_{i,s} - Q_0)\widetilde{b}_s, \qquad \theta_i = A_i^{-1} h_i. \tag{4}$$

Its prediction is $m_i(b) = \mathrm{clip}_{[0,1]}(Q_0 + \widetilde{b}^\top \theta_i)$, including a regularized intercept and no UCB bonus. It uses frozen task vectors and all accepted history, whereas local estimation uses a rolling memory. The reported mixture has $\eta = 0.5$; local-only and direct-only ablations set $\eta = 0$ and $\eta = 1$.

Similarity and training. The residual encoder is $f_\phi(b) = \mathrm{norm}(b + \tanh(s_\phi)\Delta_\phi(b))$, where norm is ℓ_2 normalization and Δ_ϕ comprises LayerNorm, two tanh hidden layers, and a linear output map. Its nonnegative local weighting function is

$$K_\phi(b, b') = \max\left(0, \frac{\langle f_\phi(b), f_\phi(b')\rangle + 1}{2} - \delta\right). \tag{5}$$

No positive-semidefiniteness assumption is required. Training and inference use the same hard threshold, with the usual subgradient at its boundary. Geometry training recomputes similarities on saved routing-time histories; eligible events have nonempty history and sufficient feedback confidence. Simulation and three-answer medical routing use $a_e = 1$ in Eq. (2), since slot-conditional selection probabilities are not set-inclusion probabilities. Single-answer medical routing uses $a_e = \min(5, 1/\pi_e)$ with the actual action probability; clipping does not establish unbiased estimation. No ranking or pairwise objective is added.

Implementation settings. We use $Q_0 = 0.5$, $\beta = 3$, $W = 160$, and ridge $\lambda = 0.1$. The encoder takes 24-dimensional simulation or 256-dimensional medical vectors, with two width-128 hidden layers and $\delta = 0.35$. Xavier weights, zero biases, and $s_\phi = 0.1$ initialize it. AdamW uses learning rate 10^{-3}, weight decay 10^{-4}, and gradient-norm clipping at 1. Training begins after 96 observations with batches of 24 and feedback confidence at least 0.2.

The frozen-geometry ablation keeps the same initialized residual map, close to but not identical to the normalized base representation. Stored task vectors are not requantized or renormalized by the quality learner. Encoder initialization does not consume routing RNG draws.

A.2 Routing and aggregation

Public fit is $v_i(x) = (1 + \cos(g(x), u_i^{\mathrm{emb}}))/2$. Local support $n_i(b) = \sum_{s \in D_{i,W}} c_{i,s} K_\phi(b, b_s)$ gives the optional bonus $U_i(b) = (1 + n_i(b))^{-1/2}$. The reported experiments use ϵ-greedy selection on learned quality, with $\epsilon = 0.05$. Each slot samples uniformly from remaining nodes with probability ϵ, otherwise uniformly among remaining score maximizers; all scores are fixed before feedback. There is no top-M restriction or forced-trial schedule. For $k = 1$, π_e sums the uniform-exploration probability and, for a maximizer, its share of the greedy probability.

For answer set A_t, completed answer $y_{i,t}$, and clipped reported confidence $r_{i,t}$, simulation uses $\kappa_{i,t} = \max(0.05, Q_i(b_t)) \max(0.05, r_{i,t})$ and

$$\widehat{y}_t = \arg\max_y \sum_{i \in A_t} \mathbf{1}[y_{i,t} = y]\kappa_{i,t}. \tag{6}$$

Failed executions do not vote; ties use fixed answer-string ordering. Quality weights precede feedback, and current evaluator scores never enter this vote. Single-answer medical routing returns the selected

response; three-answer medical routing instead uses uniform majority vote with task-keyed random ties, excluding quality and reported-confidence weights.

A.3 Feedback, evaluator reliability, and update order

Executor feedback. Verified correctness has unit weight and supersedes weak feedback. Peer feedback is the fraction of other completed selected answers agreeing with a node's response, with weight 0.2 (zero agreement if no other answer completes). The main independent-evaluator simulation disables peer targets and uses the evaluator's score instead. With no enabled source, unverified successful execution receives no update; an observed execution failure receives target zero and unit weight.

Evaluator state. Scores are classified at 0.5. Let n_j^+, n_j^- count verified positive and negative candidates and $\mathrm{TP}_j, \mathrm{TN}_j$ correct acceptances and rejections. The complete protocol computes reliability and weak-update weight as

$$\begin{aligned} \rho_j &= \frac{1}{2}\left(\frac{\mathrm{TP}_j + \nu r_0}{n_j^+ + \nu} + \frac{\mathrm{TN}_j + \nu r_0}{n_j^- + \nu}\right), \\ c_{ji,t} &= \mathrm{clip}_{[0,c_{\max}]}\left(c_{\mathrm{base}}\frac{\rho_j(t^-) - 0.5}{r_0 - 0.5}\right), \end{aligned} \tag{7}$$

with $r_0 = 0.70$, $\nu = 4$, $c_{\mathrm{base}} = 0.55$, and $c_{\max} = 0.9$. This weights the original score rather than recalibrating its value. Evaluators are selected by ϵ-greedy sampling on past reliability with $\epsilon = 0.1$ and uniform ties, outside the current answer set. Every 30 tasks, three eligible evaluators are selected without replacement, each judging a balanced pair of labeled anchors. Anchor labels update evaluator reliability only; ordinary verified outcomes also calibrate evaluators while supervising executor quality. Evaluator state is separate from both executor models, and each evaluator–evidence pair is counted once. Anchor identities are synthetic fixtures, without demonstrated concealment from strategic judges.

Update order. Each round selects all answerers, executes them, obtains optional judgments, and commits the aggregate answer. Verified or enabled weak feedback then updates executor models, using evaluator reliability available before that round. New verified judgments and scheduled anchors update evaluator state for later rounds. Geometry training uses saved routing-time histories; a final pending-batch flush cannot alter committed answers. Appendix E.2 defines the fixed-weight, no-anchor, and other controls.

B Experimental Protocols and Statistics

B.1 Simulation environment and supervision

Three frozen configurations (generation seeds 13, 29, 47) each contain 48 nodes, 2400 tasks, and supplied 24-dimensional task vectors. Task order, node definitions, audit RNG, and potential task–node outcome draws are shared across methods. Typical specialists have high-effort ability in $[0.72, 0.84]$ within their specialty and $[0.18, 0.38]$ outside it; a small minority are generalists. Twelve nodes coordinate on a wrong answer. Private state is used for execution and diagnostics, not ordinary routing.

The generator samples high effort using a node's task-specific effort probability. Before difficulty adjustment, correctness probability is its ability under high effort and 0.05 otherwise. It subtracts $0.20(d - 0.5)$ for task difficulty d, then clips to $[0.02, 0.98]$. Both the correctness draw and returned confidence use this probability. Thus confidence is idealized, and the ability–effort product used by Oracle is not the complete observable correctness probability. All tasks use a common correct-answer symbol, with a distinct colluder-error symbol and three other error symbols; methods aggregate selected responses, and the concentrated label prior limits truth-discovery comparisons.

All methods share an ordinary verification schedule covering approximately 31% of tasks; Section 4.2 specifies this schedule and the invocation budgets. Main HICS obtains independent judgments for

weak learning; Agreement + Spot uses peer targets, and Online DS uses response overlap. Verified-only baselines skip missing labels. Each ordinary evaluation request scores all three selected answers; anchor labels supervise evaluators only. We report full-stream accuracy, including cold start, and accuracy over the final 400 tasks.

B.2 Medical environment and protocols

Each of four datasets contributes 200 unique questions. Primary single-answer and supplementary three-answer routing present these 800 questions in two separately shuffled passes. Twelve prompted GPT-5.5 configurations share identical public descriptions and vectors, while private profiles induce specialization. Visible-Fit therefore breaks public-fit ties; its difference from Random reflects the RNG path. Oracle ranks private fit, not the best realized candidate answer.

The gate compares raw cosine similarity between the frozen task and private-profile vectors with a node-specific fixed threshold (0.16–0.31 across nodes). Below threshold, the environment uses the reference label to construct an incorrect answer; otherwise it returns the fixed task–node cached response. This semi-synthetic error mechanism is not natural clinical error. Gate-generated wrong options are keyed by event ID and can differ between passes, but their correctness stays zero. Policies receive correctness only under the GT mask. The supplied 256-dimensional vectors are recorded as task-description-text representations, not image embeddings; exact replay uses these frozen vectors, the node mapping, gate configuration, question manifest, and cache. Full/Half/Sparse masks reveal 1600/800/160 question outcomes. The reported new plain-baseline runs have no terminal failures or unresolved cache requests.

The three-answer protocol uses the same stream and 10% GT mask, selecting all answerers before observing responses. Uniform majority vote avoids relying on the gate's fixed incorrect-response confidence of 0.2 or confidence missing from some cached responses. Each method makes 4800 answer invocations; 160 revealed question labels yield 480 verified candidate outcomes. HICS peer and Agreement + Spot use 4320 additional agreement targets with weight 0.2; Online DS treats these unverified responses as latent observations. No independent medical judgments or anchors are collected. Appendix D.2 separately defines single-pass calibration/test partitions of this same cache.

B.3 Statistical units and reproducibility

Scores are computed from recorded test-time decisions against frozen labels. Simulation reports means and sample SDs across three configurations. Medical comparisons use five algorithm seeds on fixed task order, cache, nodes, hyperparameters, and GT masks, with a fresh learner in each run. They measure algorithm variability, not uncertainty over datasets, response generation, or feedback masks. Medical accuracy equally weights the four datasets; Mean across GT averages conditions within each seed before computing the five-seed mean and sample SD. Repeated questions, passes, and GT conditions are not independent replicates.

The calibration study reports SD over five overlapping group-disjoint train/test partitions; offline routers are fitted before testing. The second-pass diagnostic uses only the original seed. All SDs use denominator $n - 1$; paired comparisons align configurations or seeds, and no statistical significance is asserted. Simulation curves use trailing windows of 400 tasks sampled every 200 tasks.

C Baseline Implementations and Comparison Scope

All online methods select answerers before feedback and skip missing verified labels unless a declared weak source is enabled. Simulation uses the common vote interface with each method's pre-feedback quality prediction and reported answer confidence. Medical three-answer routing uses uniform voting. The following adaptations do not inherit original single-action regret or incentive guarantees.

Nonlearning and reputation controls. Visible-Fit ranks public-profile cosine similarity with uniform ties; profiles vary in simulation but coincide in medicine. Simulation Oracle ranks hidden ability times high-effort probability; medical Oracle ranks private similarity. These privileged references are not realized-answer upper bounds. Their simulation votes use the selection score and confidence with floors of 0.05. Global Reputation maintains Beta$(1, 1)$ success/failure counts from

unit-weight verified outcomes, uses the posterior mean, and routes with $\epsilon = 0.05$ and uniform ties, without an optimism bonus (Resnick et al., 2000).

Agreement and truth estimation. Agreement + Spot adds peer-agreement targets of weight 0.2 to the same global Beta estimator, with verified labels taking precedence. It shares the HICS peer protocol's targets but has no task geometry, contextual quality model, rolling memory, or evaluator state. It is an agreement control, not a formal peer-prediction payment or incentive-compatibility test (Miller et al., 2005; Gao et al., 2019). Main simulation HICS instead receives independent judgments.

Online DS adapts Dawid–Skene estimation to completed past tasks (Dawid & Skene, 1979). It stores the three selected responses; revealed truth anchors the task posterior, otherwise truth remains latent. Every 50 tasks, up to 20 EM iterations (stopping at parameter change below 10^{-6}) fit class priors and worker confusion matrices, with unit pseudocounts and initial diagonal probability 0.6. Simulation uses five symbols and 5×5 matrices; medicine fits separate AB, ABC, and ABCDE models. Expected diagonal accuracy under the learned prior drives $\epsilon = 0.05$ routing. Current responses never select their own answerers. This online adaptation uses richer response observations than verified-only methods, without extra invocations; it is not a reproduction of the original offline estimator or a task-contextual router.

Nearest-neighbor prediction. Online KNN averages each node's 40 nearest previously verified outcomes (all available if fewer) using cosine distance on shared fixed task vectors. Empty histories predict 0.5; equal-distance neighbors retain chronological order. Routing uses $\epsilon = 0.05$ and uniform ties, without learned geometry, a direct mixture, or weak targets. The rule is fixed across all online protocols and does not reproduce CARROT's offline calibration procedure.

Bandits. Disjoint LinUCB uses raw vectors without an intercept, zero prior mean, raw binary rewards, ridge 0.1, and exploration coefficient 0.1 (Li et al., 2010). KernelUCB uses a node-block RBF kernel (length scale 1), ridge 0.1, coefficient 0.1, and variance $(1 - k^\top (K + \lambda I)^{-1} k)/\lambda$ (Valko et al., 2013). Both keep full revealed histories; clipping means to $[0, 1]$ affects vote weights only, not ranking. Context-free UCB uses UCB1 empirical means and bonus $\sqrt{2 \log(\max(2, N))/n_i}$, with revealed-feedback counts N, n_i; unseen nodes are prioritized with random ties (Auer et al., 2002). It has no profile term, prior smoothing, or history window. Simulation selects a simultaneous top-three batch before learning.

Neural adaptations. NeuralUCB-Diag uses a width-100 two-layer network, disjoint block context–node map, raw gradient features, ridge 0.001, and exploration $\nu = 1$. SGD uses learning rate 0.01, weight decay equal to ridge, and at most 1000 steps per fit. Fits occur on each revealed action through 2000, then every 100; all samples are retained. Selection-time gradients update uncertainty only for revealed outcomes. Sample retention between scheduled fits corrects the source example's retention behavior (Zhou et al., 2020).

NeuralLinearTS uses one width-50 ReLU layer, normal–inverse-gamma prior $a_0 = b_0 = 6$, ridge 0.25, and two initial pulls per node. Source-example RMSProp performs 100 minibatch steps of size 512 every 50 revealed observations with global learning-rate decay. Representation changes refresh every arm posterior; otherwise only the selected arm is refreshed, correcting a source slicing omission (Riquelme et al., 2018). Both neural methods are CPU/formula and sparse-feedback adaptations, not bitwise CUDA/TensorFlow reproductions.

Offline controls and training-only selection. The separate calibration protocol (Appendix D.2) uses the best calibrated node, adapted CARROT-KNN, per-node L2 logistic regression, and multi-output random forest. Five-fold training-only CV preserves image/text groups. CARROT-KNN searches $K \in \{2, 4, 8, 16, 32, 40, 64, 128\}$ with cosine distance and uniform averaging, selecting mean multi-output validation R^2 as in the public regression implementation (Somerstep et al., 2025). Logistic regression selects $C \in \{0.01, 0.1, 1, 10\}$ by Brier loss, using constant predictors for single-class training subsets. The 100-tree forest selects depth in $\{10, 20, \text{unlimited}\}$ and minimum leaf size in $\{1, 5\}$ by Brier loss. Parameter ties choose the first declared candidate; routing ties are random. The generic supervised controls are not named-paper reproductions. CARROT-KNN uses shared features and an accuracy-only, equal-invocation-cost objective, not its original embedding or cost benchmark.

Comparison scope. Online baselines use the disclosed hyperparameters across feedback conditions; offline controls use the training-only selection above. Rankings and boldface refer to the displayed methods. Complete tested-run results are retained in the accompanying analysis. No reported baseline adds HICS's quality mixture or independent evaluator feedback.

C.1 Relation to other routing protocols

Related routing protocols differ in their supervision resources and action spaces. Calibration-based methods use historical outcomes or preference information (Chen et al., 2024; Somerstep et al., 2025; Panda et al., 2025; Tsiourvas et al., 2025); Appendix D.2 examines this regime separately. Our online comparisons start without a calibration matrix and use fixed verification schedules. Other methods optimize which outcomes to verify (Wei et al., 2026), select collaboration modes and roles (Yue et al., 2025), or address human–AI allocation (Gao et al., 2021). These choices require distinct evaluation protocols and resource accounting.

D Additional Medical Evidence

D.1 Retaining information across repeated work

The second pass repeats questions and cached responses. An original-seed (13) diagnostic intervenes at the pass boundary (Table 6). This intervention was not repeated over the five-seed matrix.

Table 6: Second-pass accuracy (%; seed 13). All branches share the first 800 decisions. Continue retains and updates quality state; Freeze retains it without further learning; Reset reinitializes it but keeps routing time, counts, and RNG.

GT	First pass	Continue	Freeze	Reset
Full	70.75	79.38	79.25	71.75
Half	69.50	77.38	77.12	67.12
Sparse	62.75	70.12	69.88	64.25

Continue exceeds Reset by 7.625/10.250/5.875 points under Full/Half/Sparse GT, but exceeds Freeze by only 0.125/0.250/0.250. Acquired state helps on recurring work, with little additional second-pass learning. This does not isolate recurrence from cache reuse or measure held-out-question generalization.

D.2 Historical calibration and cold-start routing

Protocol. Five paired splits assign 200 existing questions to calibration and 600 to a single-pass test stream (50/150 per dataset). Source-question IDs are disjoint; shared image hashes or identical text-only questions stay together. Original first-occurrence event IDs preserve the gate and cache. Each test block of 200 balances the datasets. This post hoc repartition uses an already studied cache, not a pristine external generalization test; overlapping splits are not independent datasets.

Each offline router receives the full 200×12 training correctness matrix, uses the training-only selection in Appendix C, and remains fixed during testing. HICS keeps its main configuration without retuning, but starts with empty quality histories and no calibration matrix, peer targets, or independent judges. After committing each test answer, it receives 600 or exactly 60 selected-node labels, with the sparse mask balanced across datasets and blocks. All methods make 600 test answer invocations. This is not an equal-information-budget comparison. With only 60 labels, HICS does not reach its 96-observation encoder warmup: local and direct estimates update, but geometry takes no training steps. All six configurations and five splits were specified before execution and are included.

Results and sensitivity. The calibrated contextual routers outperform cold-start HICS in both windows. Logistic regression reaches 80.63% full-stream accuracy versus 76.80% for CARROT-KNN and 69.83% for full-feedback HICS; sparse HICS also trails the calibrated fixed-node control. This illustrates a limitation of the evaluated cold-start configuration when complete historical outcomes

Table 7: Historical calibration and cold-start routing (accuracy, %; mean ± sample SD over five paired splits). Offline outcomes are training question–node records; online outcomes are revealed selected-node test results. All 600 includes cold start; Last 200 is the predeclared adaptation window.

Method	Offline outcomes	Online outcomes	All 600	Last 200
Best calibrated node	2400	0	65.37 ± 0.55	66.00 ± 3.18
CARROT-KNN (adapted)	2400	0	76.80 ± 1.57	77.00 ± 3.98
Logistic regression	2400	0	80.63 ± 1.47	81.60 ± 2.82
Random forest	2400	0	75.43 ± 1.62	75.90 ± 3.90
HICS (10% feedback)	0	60	61.70 ± 3.14	63.50 ± 5.10
HICS (100% feedback)	0	600	69.83 ± 2.18	71.80 ± 3.29

are available. Unequal supervision and a different workload prevent attributing the gap solely to architecture or extending it to the repeated-stream comparison.

CARROT implementation sensitivity. The public KNN functions reproduce the selected neighbor counts and test predictions under the same grouped folds; this checks the shared-feature adaptation, not the original embedding or cost benchmark. Post hoc Brier-based selection matches R^2, while fixed $K = 40$ raises full-stream accuracy from 76.80% to 77.90%; upstream first-argmax ties give 76.50% and 77.67%, respectively. All six sensitivity configurations are retained in the supplementary statistics without replacing the primary results; logistic regression remains ahead. The normalized-feature gate has a linear boundary that may favor global prediction, but cached answer errors also matter and this explanation was not causally isolated.

E Additional Simulation Evidence

E.1 Learning dynamics by configuration

Figure 3 separates the three configurations underlying Figure 2, with identical methods and windows. Final-window HICS gains over LinUCB range from 4.25 to 12.75 points; overlapping windows are not independent replicates.

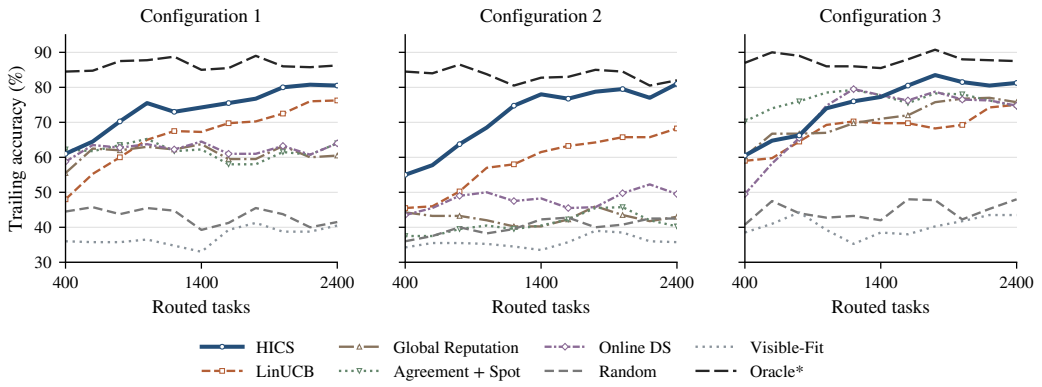

Figure 3: Simulation trajectories by configuration: trailing-400-task accuracy every 200 tasks, without smoothing. All panels use 30–95%, including every observation. Oracle* is privileged.

E.2 Feedback sources and evaluator controls

Sources and budgets. Table 8 compares verified-only, peer, and independent-evaluator HICS. Evaluation replaces peer targets and adds requests, so these are not equal-total-budget algorithms. Relative to peer feedback, evaluation gains 7.83/5.75/0.46 full-stream points and +3.75/+1.75/−1.75 final-window points across configurations.

Table 8: HICS feedback configurations (accuracy, %; mean ± sample SD). Each uses 7200 answer invocations and the same ordinary GT/audit schedule; independent evaluation adds 2640 requests.

HICS configuration	All 2400	Last 400	Extra eval. calls
Verified feedback only	66.82 ± 2.68	75.67 ± 3.55	0
Peer feedback	68.49 ± 4.67	79.67 ± 3.15	0
Full evaluator (main)	73.17 ± 1.28	80.92 ± 0.38	2640

Controls. B0 disables evaluators; B1 collects judgments without executor weak updates; B2 fixes weak weight at 0.55; B3 ignores anchors but retains ordinary verified checks; B4 samples evaluators uniformly; B5 is complete HICS. B0–B5 were specified before this panel ran. B1–B5 share 2640 requests; B0 uses none. Answer budgets, ordinary GT schedules, aggregation, and quality models are unchanged. Subsequent exploratory B6 (uniform selection, fixed weight) and B7 (no anchors, fixed weight) match B2's request budget and total weak-update weight.

Table 9: Evaluator controls: paired performance loss (percentage points) across three configurations. Positive means degradation. Panel (a) uses B5 (73.17% all, 80.92% last 400); panel (b) uses B2 (73.19% all, 81.17% last 400) to match weak-update mass. B0 has no evaluation requests.

Configuration	Loss: all	Loss: last 400
(a) Reference: B5, complete protocol		
B0: No evaluator	+6.3	+5.3
B1: Calls, weak updates off	+6.3	+5.3
B2: Fixed weak weight	0.0	-0.3
B3: No anchor calibration	-2.3	-0.3
B4: Uniform evaluator selection	+2.6	+2.3
B5: Full evaluator protocol (reference)	0.0	0.0
(b) Reference: B2, fixed weight (post hoc controls)		
B6: Uniform + fixed weight	+2.6	+1.3
B7: No anchors + fixed weight	-0.8	+0.2

B0/B1 have identical decisions, verifying that unused judgments do not affect routing. B5 beats B1 in every final window, but fixed weighting and omitting anchors have slightly higher final-window means; no anchors also improves full-stream accuracy. Additional judgments help without establishing every calibration component's necessity.

Supervision strength. Evaluator correctness probabilities range from 0.52 to 0.92, independent of answer ability and answer-role collusion; ordinary and anchor judgments use the same noisy mechanism, returning scores 0.18 or 0.82. B5 weak-label binary accuracy is 89.55/89.66/84.00%, and mean weight is about 0.86 versus peer weight 0.20. B2's total weak-update weight is about 8204 across configurations versus 12813 for B5: adaptive weighting changes learning exposure. B3 still learns from ordinary verified checks. B2 exceeds B6 by 1.00/1.25/1.75 final-window points; its anchor increment over B7 is small and not uniformly positive. These post hoc controls reuse the configurations, without independent confirmation; hidden-label diagnostics never enter routing.

E.3 Sparse verification and collusive evaluators

Scope and intervention. This study is exploratory rather than independent confirmation: scenarios were chosen after earlier results, then fixed before the matrix ran. All three original configurations, quality settings, answer generation, and aggregation are retained. Sparse verification disables ordinary random audits, reducing approximately 31% combined GT/audit exposure to 8% direct GT. The three scenarios are sparse/honest, standard/collusive, and sparse/collusive.

In collusive scenarios, the 25% answer colluders also judge dishonestly, assigning 0.82 to Z and 0.18 to other answers. Honest judges retain independent accuracy in $[0.52, 0.92]$. Judges remain outside the current answer set. All stress scenarios use balanced correct/incorrect anchors, with incorrect symbols cycling B/C/D/Z rather than exclusively targeting Z. Adaptive attacks and anchor recognition are untested.

Matched comparisons. B0/B2/B3/B5/B7 retain their definitions. All use 7200 answers; all except B0 add 2640 evaluation requests. B2/B7 share fixed weight 0.55 and total update mass; B3 retains ordinary verified calibration. Sparse/collusive B6 additionally matches B2's answer, evaluation, and weak-update budgets (Table 5). All 57 matrix runs are retained, plus one separate replay of an original complete-protocol trajectory.

Table 10: Exploratory stress controls. Full HICS gives accuracy (%; mean $\pm$ sample SD); other rows give paired loss relative to it, with negative values indicating improvement. 8/H, 31/C, 8/C denote approximate verification rate and honest/collusive judges. Each cell uses all three configurations. B0 changes evaluation budget.

Reference / ablation	8/H: all	8/H: last	31/C: all	31/C: last	8/C: all	8/C: last
Full HICS (accuracy)	71.62 ± 4.92	81.00 ± 5.89	74.42 ± 2.40	80.83 ± 1.38	71.61 ± 5.00	79.17 ± 3.56
No evaluator	+15.1	+19.2	+7.6	+5.2	+15.1	+17.3
Fixed weak weight	+1.0	+0.4	+1.5	+1.3	-0.3	-1.0
No anchor calibration	-0.5	+0.7	+2.3	+0.4	-1.3	-0.8
No anchors + fixed weight	+1.0	+3.5	+1.8	+1.2	+1.4	+0.4

Under sparse/collusive feedback, reliability-aware selection reduces malicious evaluator assignments from 24.2% to 2.6% and raises binary weak-label accuracy from 63.01% to 88.11%. These hidden-outcome diagnostics explain the matched B2–B6 result under this attack, without entering routing. Fixed-weight anchor gains (B2 versus B7) in the last window are +3.08/$-$0.08/+1.42 points across scenarios. In sparse/collusive feedback, fixed weighting and omitting anchor calibration both outperform Full HICS. Ordinary verified checks already calibrate evaluators, while adaptive weighting changes exposure; not every calibration component is uniformly necessary.