

From Reasoning Allocation to Behavioural Specialisation: Boundary Results in Multi-Robot Systems

Pouya Mansournia

p.mansournia@gmail.com

Research Article

Keywords: multi-robot systems, value of computation, outcome-aware routing, cognitive delegation, behavioral specialization, emergent collective cognition

Posted Date: August 25th, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-10773364/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: No competing interests reported.

From Reasoning Allocation to Behavioral Specialization: Boundary Results in Multi-Robot Systems

Pouya Mansournia^{1*}

^{1*}Independent Researcher.

Corresponding author(s). E-mail(s): p.mansournia@gmail.com;

Abstract

Invoking an expensive reasoning backend for every robotic decision is wasteful, but deciding when the added reasoning is worth its latency and compute cost remains an open problem. We study this in a simulated multi-robot fleet along three connected axes: whether reasoning-invocation can be learned from outcome data and evaluated against a privileged Offline Oracle, whether reasoning can be usefully delegated to peers rather than performed locally, and whether repeated delegation produces stable behavioral specialization that would justify the label emergent. The Offline Oracle shows positive-value reasoning opportunities are intrinsically sparse (escalation rate 0.14–2.06% across regimes), triggering this study’s own pre-registered stop condition; all subsequent phases are reported as exploratory, not confirmatory. Within this sparse-opportunity regime, a learned binary router avoided a hand-tuned heuristic’s over-escalation, reducing regret relative to the Oracle by roughly 57–152 $\times$. However, on every in-distribution held-out evaluation the router escalated on exactly 0% of decisions, with utility and Oracle-regret numerically identical to a policy that never reasons: its apparent success is fully explained by classifier collapse under severe class imbalance, not by learned discrimination. Peer-to-peer cognitive delegation produced no measurable benefit in calibrated homogeneous fleets, while the heterogeneous capability condition showed higher utility and concentrated delegation activity, without a matched no-delegation control to isolate the causal source. Repeated interaction produced real, path-dependent concentration of delegated work (Gini 0.478 vs. a null-model floor of 0.091), of which a matched comparison attributes roughly one third to two identified implementation artifacts; this specialization does not satisfy a predefined eight-criterion test for emergent collective cognition. We conclude that this environment exposes a sparsity/learnability boundary for outcome-aware reasoning allocation, and that reasoning allocation, delegation, specialization, and collective emergence remain empirically separable phenomena requiring independent evaluation.

Keywords: multi-robot systems, value of computation, outcome-aware routing, cognitive delegation, behavioral specialization, emergent collective cognition

1 Introduction

Embodied agents that can invoke a large-model-style reasoning backend face a basic economic problem: reasoning improves decision quality but costs latency, compute, and, in a fleet, communication. Naively reasoning on every decision is

wasteful; naively never reasoning discards genuine value. A selective-reasoning mechanism must therefore decide *when* the added cost of reasoning is justified by the expected improvement in outcome. This is the classical value-of-computation (VOC) question (Hay, Russell, Tolpin, & Shimony, 2018; Russell & Wefald, 1991).

Prior work in this project’s own lineage (the previous RACA system) used a hand-tuned heuristic to decide when to escalate to expensive reasoning. A heuristic threshold, however, is a proxy: it encodes a designer’s belief about when reasoning helps, not a measurement of when it actually does. This leaves open whether the heuristic’s escalation decisions track genuine outcome value or merely a plausible-looking correlate of it.

The deeper question is therefore not only whether a router can be built, but whether the decision of *when to reason* can be learned directly from outcome data and validated against a privileged upper bound on what any router could achieve. That upper bound is an Offline Oracle with counterfactual access to both the deterministic and the reasoning-backed outcome for the same decision.

In a multi-robot fleet, a second question follows immediately: if reasoning is valuable, does it have to happen on the robot facing the decision, or can it be delegated to a peer that is momentarily better positioned, whether less loaded, more reliable, or already primed by relevant experience? Peer-to-peer cognitive delegation reframes “should I reason” as “who should reason.”

Repeated delegation raises a third question: does letting robots repeatedly choose delegation targets, with no explicit role assignment, cause the fleet to differentiate, with some robots becoming persistent recipients of delegated work? And if it does, is that differentiation properly described as *emergent collective cognition*, or merely as a statistical regularity of a specific decision rule operating on a specific interaction history?

This paper investigates all three questions but deliberately keeps them separate. A positive answer to the first (a router avoids a heuristic’s excess cost) does not imply that the router learned when reasoning is valuable, and a positive answer to the second (delegation helps) does not imply the third (differentiation is emergent in any principled sense). Distinguishing these is the paper’s central methodological commitment: the Offline Oracle result, which shows that positive-value reasoning opportunities in this environment are intrinsically sparse, is reported first, ahead of the routing result it contextualizes, precisely

so that the routing result is read as a sparsity/learnability boundary finding rather than as evidence of successful learned routing.

Contributions.

1. An Offline Oracle analysis quantifying that positive-value reasoning opportunities are intrinsically sparse in the evaluated environment, and a demonstration that a learned binary router, evaluated against that Oracle, collapses to a never-reason policy rather than learning to identify those sparse opportunities. This is a sparsity/learnability boundary result for outcome-aware reasoning allocation, not a positive routing result.
2. A calibrated peer-to-peer delegation architecture evaluated across homogeneous and heterogeneous fleet conditions, reporting an explicit null result for the calibrated homogeneous case, and disclosing that the heterogeneous condition’s benefit cannot be causally attributed to delegation specifically because no matched no-delegation heterogeneous control exists.
3. A quantitative, artifact-controlled characterization of path-dependent behavioral specialization in delegated-work concentration, including a corrected accounting, from a matched control comparison, of how much of an originally observed effect was attributable to implementation artifacts.
4. An explicit, criterion-by-criterion audit against a predefined emergence test, concluding that the observed specialization does not satisfy it.
5. A transparent disclosure of a genuine protocol deviation. A pre-registered stop condition triggered by the Oracle result was not honored, so all subsequent experimental phases are reported with correspondingly reduced evidentiary weight. Statistical and reproducibility limitations affecting specific analyses in this study are also disclosed (Section 6).

2 Related Work

2.1 Value of computation and metareasoning

The value-of-computation framework (Russell & Wefald, 1991) formalizes when a bounded-rational agent should keep reasoning versus act, and has been extended to learned VOC estimation for

planning (Hay et al., 2018) and, more recently, to metareasoning over an LLM’s own inference steps (De Sabbata, Summers, AlKhamissi, Bosse-lut, & Griffiths, 2024). None of this line validates its VOC estimate against a privileged, counterfactually-computed oracle over realized task outcomes, and none addresses a multi-agent delegation question.

2.2 LLM and model routing

RouteLLM (Ong et al., 2025) learns to route a query to a cheap or expensive model from preference data; recent surveys (S. Chen, Liu, Han, Zhang, & Liu, 2024; Moslem & Kelleher, 2026) catalogue a growing space of routing paradigms, and test-time compute scaling (Snell, Lee, Xu, & Kumar, 2024) studies a related but distinct allocation question (how much compute per query, not which backend). These systems evaluate against preference labels or benchmark accuracy rather than a counterfactual-outcome oracle, and none embeds the routing decision in an embodied, physically-costed fleet setting.

2.3 Multi-robot coordination and delegation

Broadcast-based shared task experience (Shida, Jimbo, Odashima, & Matsubara, 2025) and the centralized-versus-decentralized comparison of Y. Chen, Arkin, Zhang, Roy, and Fan (2024) are the closest existing analogues to this paper’s delegation and centralization questions, respectively; neither, however, gates delegation on a learned, oracle-validated value-of-reasoning signal. Jamshidpey et al. (2024) studies a genuine centralization/decentralization performance tradeoff in coverage tasks but without a reasoning-cost dimension. Market-based and auction coordination (Dias, Zlot, Kalra, & Stentz, 2006; Gerkey & Mataric, 2002) is the classical seminal alternative to learned delegation; its allocation rule is hand-specified rather than learned from a reliability signal accumulated online.

2.4 Bandit lock-in and winner-take-all dynamics

The path-dependent hub-formation dynamic this paper characterizes is an instance of a well-known family of results on myopic/greedy bandit lock-in,

tracing at least to Rothschild (1974) and characterized precisely in modern form by Slivkins, Xu, and Zuo (2025) and Bayati, Hamidi, Johari, and Khosravi (2020). It is explicitly *not* a Pólya urn (no proportional reinforcement) and not a direct instance of power-of-two-choices load balancing (Mitzenmacher, 2001), which samples a random subset before choosing greedily; this paper’s delegation rule instead performs a full-information hard argmax over all available peers, a regime in which lock-in is, if anything, more likely.

2.5 Engineered versus tested-for role emergence

A substantial body of multi-agent reinforcement learning work *engineers* role diversity as a training objective: ROMA (Wang, Dong, Lesser, & Zhang, 2020) and ACORM (Hu et al., 2024) use mutual-information or contrastive objectives to actively reward inter-role diversity, and R3DM (Goel et al., 2025) extends this to future-behavior-conditioned role discovery. CORY (Ma et al., 2024) uses a scheduled, externally-imposed role-swapping mechanism to stabilize LLM fine-tuning. These systems report engineered specialization as a positive design outcome; they do not test whether unmodified, architecturally symmetric agents develop specialization on their own, and none reports or investigates a negative outcome. Riedl (2026) studies role differentiation in LLM dialogue without physical embodiment or an outcome-value dimension. Broader surveys of collective intelligence in multi-agent and LLM systems (Casadei, 2023) share this paper’s caution about the word “emergence” but do not supply a specific, falsifiable multi-criterion test of the kind used here.

Novelty statement.

Among the works reviewed, we did not identify prior work combining oracle-validated reasoning allocation, calibrated peer delegation, and artifact-controlled specialization/emergence analysis within a multi-robot experimental framework.

3 Methods

3.1 Experimental environment

Experiments use `FleetSimulator`, a deterministic, pure-Python/NumPy multi-robot simulator with two task-difficulty configurations (*small_dense*, *large_sparse*). Determinism was verified directly: repeated runs from the same seed produce bit-identical output (`reproducibility-check.json`, `identical_reruns: true`).

3.2 Decision backends

Four backends are compared: B0 (always deterministic), B1 (always invoke the expensive reasoning backend), B2 (the previous RACA project’s hand-tuned escalation heuristic, reimplemented), and B3 (this project’s learned outcome-aware router). An Offline Oracle, described in Section 3.6, provides a non-deployable upper bound. Figure 1 summarizes how these pieces connect: a decision state is evaluated counterfactually against both backends, the Offline Oracle and the router label are both derived from that same counterfactual pair, B3 outputs an escalation probability compared against a calibrated threshold, and an escalated decision then enters the delegation layer of Section 3.7 before contributing to the specialization measurements of Sections 4.7–4.11.

3.3 Previous RACA baseline

B2 reimplements the routing heuristic used by an earlier, unpublished internal predecessor of this project (referred to throughout as “the previous RACA system”); it is not an external published system, and no independently citable reference or archived configuration exists for it outside this project’s own repository history. Its escalation mechanism was **reimplemented and qualitatively validated** in the current codebase: the reimplementation passes 7/7 unit tests, and the current simulator environment is independently confirmed deterministic (identical output on repeated runs from the same seed). The previously reported *quantitative* advantage of this heuristic over simpler baselines did not reproduce under the current simulator’s cost parameterization (Section 4.1). Two contributing factors were

identified: reasoning-backend latency cost ($\approx 3\text{s} \times \lambda_{\text{latency}} = 0.15 \approx 0.45$ utility-units per call, which swamps typical regret magnitudes of 0.01–0.05 units) and a 3% catastrophic-failure rate in the reasoning backend that dominates the non-robust mean-regret statistic. Whether the original internal system used a different, unrecorded backend/cost parameterization could not be determined, since no archived configuration for it exists.

3.4 Counterfactual outcome dataset

For a decision state x , both the deterministic outcome $\text{Outcome}_D(x)$ and the reasoning-backed outcome $\text{Outcome}_R(x)$ are computed from the same simulator state under independently drawn backend stochasticity, so that a value-of-reasoning target can be constructed without requiring the deployed policy to actually take both actions. This is an approximation: the two outcomes are counterfactually paired at the level of the decision state, not causally identified via an intervention on a single realized trajectory, and the strength of this approximation depends on the simulator’s backend-noise model rather than being independently validated against ground truth. Utility is defined as $U(x) = -\text{regret}(x) - \lambda_{\text{latency}} \cdot \text{latency}(x)$, where $\text{regret}(x)$ is the chosen candidate’s true cost minus the minimum true cost available among all candidates for that decision, and $\text{latency}(x)$ is the backend’s response latency for that decision; $U_D(x)$ and $U_R(x)$ are this same quantity evaluated on the deterministic and reasoning-backed branches respectively.

3.5 Router label, features, and training procedure

The router observes six features per decision: ambiguity, urgency, candidate_count, cost_margin, battery_soc, and recent_latency_mean (z-score normalized against the training split’s mean and standard deviation). Contrary to a magnitude-regression framing, the router is a **binary logistic classifier**, not a regressor of expected value-of-reasoning magnitude: its training label is $y = \mathbb{I}[U_R(x) > U_D(x)]$, i.e. whether the reasoning-backed outcome’s utility exceeds the deterministic outcome’s utility for that decision, with no magnitude information retained. Its output, $\hat{p}(x) \in$

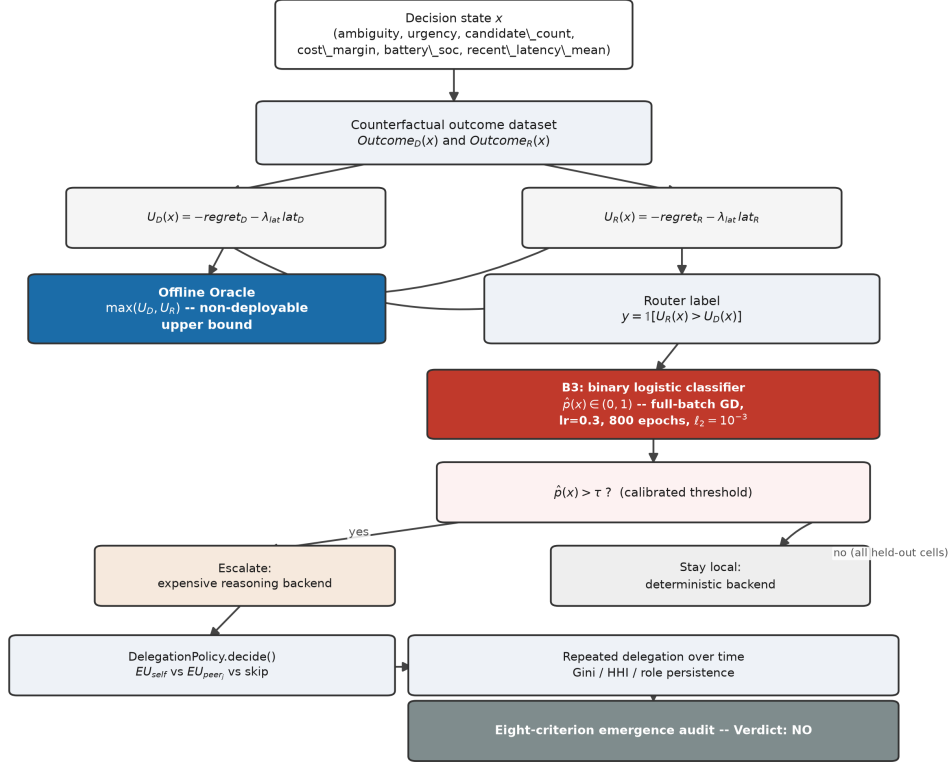


Fig. 1 Decision-layer architecture, from a single decision state through counterfactual evaluation, the Offline Oracle and learned router, delegation, and the specialization/emergence audit. Boxes are labeled with the exact quantities defined in Sections 3.6–3.7; no new mechanism is introduced beyond what those sections describe.

$(0, 1)$, is therefore correctly described as an estimated *probability that reasoning is superior for this decision*, not as an estimate of *how much* reasoning would help. The classifier is trained by full-batch gradient descent (the entire training split is used to compute each gradient step; no mini-batching or momentum term is used) on the training split (learning rate 0.3, 800 epochs, ℓ_2 weight penalty 10^{-3} , weights initialized from $\mathcal{N}(0, 0.01^2)$ with a fixed seed), and a decision threshold τ is subsequently selected on a disjoint calibration split by grid search over $\tau \in \{0.05, 0.10, \dots, 0.95\}$, maximizing mean calibration-split utility. A decision is routed to the expensive backend when $\hat{p}(x) > \tau$. Training uses a dedicated development pool, threshold selection uses a disjoint calibration pool, a disjoint validation pool is used as a sanity check before touching held-out data, and the model is evaluated exactly once on a held-out test pool never used for any prior decision

(Section 3.12). Table 1 summarizes the seed-pool sizes for each split in the original-regime configuration (Section 3.12 gives the exact seed ranges); the revised regime uses the same per-split sample counts under a disjoint seed range.

3.6 Offline Oracle

The Offline Oracle selects, for every decision, whichever of $\text{Outcome}_D(x)$ and $\text{Outcome}_R(x)$ yields higher utility. It is explicitly non-deployable: no online policy has simultaneous access to both counterfactual outcomes before choosing. Its role is to bound the maximum achievable improvement from adaptive routing, isolating how much value a routing policy could in principle capture from how well a specific policy captures it.

Table 1 Router training/calibration/validation/held-out split sizes, original regime (both configurations use the same per-split counts under disjoint seed ranges).

Split	Seed pool size	Role
Development (training)	10 seeds	Fit classifier weights
Calibration	10 seeds	Threshold τ grid search
Validation	10 seeds	Sanity check before held-out evaluation
Held-out (test)	20 seeds	Single evaluation reported in Table 4

3.7 Cognitive delegation

For a decision, `DelegationPolicy.decide()` compares three options: reasoning locally, delegating to one of the available peers j , or skipping. Delegating to peer j is scored as

$$EU_{\text{peer}_j} = \bar{q}_j - \kappa \cdot \mathcal{K}_{j,\text{cap}} - \lambda_{\text{msg}} \cdot m_j,$$

where $\bar{q}_j$ is an online-accumulated running-mean quality (reliability) estimate for peer j , updated from that peer’s observed outcomes on prior delegated decisions; κ is the `self_load_penalty` parameter; $\mathcal{K}_{j,\text{cap}}$ indicates peer j is at capacity; and m_j is the message count incurred by delegating to peer j . The self-reasoning option is scored analogously using a `self_quality_factor` multiplier on the robot’s own running-mean quality. The policy selects the argmax over all peers (limited to `peer_capacity` simultaneously available peers) plus the self and skip options; utility, feature, and threshold values were themselves calibrated by a grid search over `self_load_penalty`, `lambda_msg`, and `self_quality_factor` on a disjoint calibration pool (Section 4.5), and the reported calibrated optimum sits at the edge of the searched grid on two of these three parameters, disclosed as a limitation (Section 6) rather than treated as a confirmed global optimum. Ties in EU across candidates are broken deterministically by ascending robot ID in the originally released implementation, a design choice later shown (Section 4.7) to be the dominant driver of an artifactual concentration signal, disclosed here as a methods-transparency item rather than omitted. Peer access in all delegation experiments reported here is all-to-all (every robot can consider every other robot as a delegation candidate, subject to `peer_capacity`); no restricted communication topology was evaluated, so topology cannot explain any cross-seed variation observed within these experiments, and topology sensitivity itself remains untested.

3.8 Peer memory

Four memory conditions are compared: C0 (no sharing), C1 (unbounded shared memory), C2 (windowed shared memory), C3 (local-only memory).

3.9 Communication and computational cost proxies

Delegation and centralized-manager conditions incur a per-message latency cost λ_{msg} ; message counts are logged per decision and per episode. No separate measurement of computational cost (e.g. FLOPs, wall-clock CPU time per reasoning call) was taken; latency and reasoning-call frequency are used throughout this paper as proxies for operational and computational burden, and results are reported in those terms rather than as directly measured compute cost.

3.10 Specialization metrics

Concentration of delegate-in activity is measured via the Gini coefficient and Herfindahl–Hirschman Index (HHI) over per-robot delegate-in counts; behavioral diversity via per-robot entropy $H_i = -\sum_k p_i(k) \log p_i(k)$ over decision-type shares; and temporal stability via role persistence, the window-to-window Spearman rank correlation of per-robot delegate-in share across four 200-decision windows.

3.11 Emergence criteria

Eight criteria are used to test whether observed specialization constitutes emergent collective cognition: (1) global role structure not explicitly programmed, (2) structure arising from local interactions alone, (3) recognizable macroscopic structure, (4) cross-seed reproducibility, (5) temporal persistence, (6) measurable, useful system-level

consequence, (7) not solely explained by implementation artifacts, and (8) survival of appropriate controls.

3.12 Experimental protocol

Within each phase, development, calibration, validation, and held-out testing use disjoint seed pools, verified never to overlap within that phase (e.g. Phase 3 original: dev 3000–3009, cal 3100–3109, val 3200–3209, test 3300–3319). No parameter, feature, or threshold was altered after observing held-out results. This disjointness guarantee is within-phase; some later phases intentionally reuse or extend earlier seed pools for paired or follow-up comparisons (e.g. the artifact-control and mechanism-investigation phases reuse the same 30000-series pool across several related sub-experiments by design, to keep those specific comparisons paired), and this manuscript does not claim that no seed value is ever reused anywhere across the full experimental program; it claims only that within-phase train/calibration/validation/test partitions are disjoint. A per-table mapping from manuscript result to raw output file, generating script, and an independent verification status (raw file directly opened and checked, versus not yet independently re-checked) is maintained separately as a reproducibility appendix. At the time of writing, 7 of 15 tracked central results have been independently re-verified against raw output, including every table the paper’s main conclusions depend on (previous-RACA reproduction, Oracle headroom, router/distribution-shift regret, delegation ablation matrix, and the specialization concentration chain); the remaining rows’ unverified status is disclosed rather than assumed.

3.13 Statistical analysis

Reported intervals are 95% percentile bootstrap confidence intervals ($n = 2000$ resamples per interval) unless noted. Two distinct resampling units are used across this paper’s experiments, and the unit matters: for experiments where a seed-level per-seed statistic was archived or could be recomputed from archived per-decision data without regenerating anything (the Oracle

headroom analysis of Section 4.2 and the specialization/perturbation analyses of Sections 4.7–4.10), confidence intervals reported here are bootstrapped over independent seed-level means. For the router’s regret-vs-oracle comparison (Table 4), the originally computed confidence intervals bootstrap over pooled per-decision observations (up to 6000 per cell) rather than over the 20 independent seeds those decisions are drawn from; this is a form of pseudo-replication that can understate interval width. Recomputing this specific comparison at the correct seed level would require regenerating per-decision data, since no per-decision archive was retained for this analysis. A bounded root-cause audit was conducted to determine why: running the unmodified repository script that originally produced these results reproduces the calibrated threshold τ exactly but not the continuous router weights or regret values. Version-control history shows that no commit has modified any file on this computation path since the commit that generated the archived output, and the current environment was independently confirmed self-consistent (repeated runs match each other exactly); dependency pinning (`numpy==2.5.1`) was introduced only in a later commit explicitly aimed at closing reproducibility gaps, meaning the archived result’s original runtime environment was never pinned or recorded. The most probable explanation, consistent with all evidence gathered, is a numpy/Python runtime difference between the unrecorded environment that generated the archive and the current pinned environment, not a code, data, or procedural difference. Table 4’s values are therefore treated as **archived and verified from stored output, but not currently regenerable from this repository’s present state**; they are retained as-is rather than replaced with unverified regenerated numbers. Table 4’s reported confidence intervals should therefore be read as an approximation of, not an exact match to, seed-level uncertainty; the point estimates and the qualitative comparison to B2 are unaffected, since B3’s held-out escalation rate is exactly 0.0 in every cell regardless of resampling unit. Separately, the study evaluated multiple outcomes, ablations, and exploratory comparisons; because a complete prospective family of hypothesis tests was not defined, individual exploratory p -values and confidence intervals should not be interpreted

as family-wise confirmatory evidence, and no formal multiple-comparison correction was applied. Interpretation throughout prioritizes effect size, non-overlapping confidence intervals, and cross-seed consistency over isolated statistical tests, and overlapping confidence intervals are reported as “no detectable difference,” not as proof of equality.

3.14 Confirmatory versus exploratory status

This study’s protocol specified a stop condition: if the Offline Oracle does not meaningfully outperform simple baseline policies, routing-focused experimentation should stop and the finding should be reported rather than pursued further with increasingly sophisticated routers. This condition was triggered during the Offline Oracle analysis (Section 4.2): Oracle escalation rates were below 1% in both tested configurations. **The study nevertheless continued beyond this gate as an explicitly exploratory extension.** This is a genuine deviation from the pre-registered protocol, not a planned contingency, and it is disclosed as such rather than folded into ordinary sequential execution. As a direct consequence, every result from the learned router onward (Sections 4.3–4.11) carries exploratory, not confirmatory, evidentiary weight; only the previous-RACA reproduction (Section 4.1) and the Oracle headroom measurement itself (Section 4.2) were generated under the original prospective protocol.

4 Results

4.1 Previous RACA reproduction

Table 2 summarizes the reproduction outcome. Mechanism, code, and environment reproduce; the quantitative advantage does not. B1 (always-reason) has worse mean regret than B0 in both configurations and both cost regimes; B2 beats B0 on cost only in *small_dense*, not *large_sparse*, and does not clearly beat B0 on regret in either configuration.

4.2 Available routing headroom: the Offline Oracle

Table 3 reports the Offline Oracle analysis, computed on 20 seeds per regime (2000–2019

original, 12000–12019 revised; 400 samples/seed, 8000 samples/config total per regime) under the counterfactual-outcome dataset of Section 3.6. Confidence intervals are bootstrapped over the 20 independent seed-level means (Section 3.13), which for this analysis are close to the originally reported pooled-decision intervals (independently re-verified from the archived per-decision records; both approaches agree to within the reported precision). The value of reasoning (Oracle utility minus B0 utility) is small in absolute terms in both tested regimes, and the Oracle escalates to expensive reasoning (i.e. a positive-value reasoning opportunity genuinely exists) on a small minority of decisions even after a deliberate, pre-committed attempt to widen the task-difficulty distribution (the “revised” regime). This is the paper’s central boundary result (Figure 2): positive-value reasoning opportunities are intrinsically sparse in this environment, and this sparsity is the finding that triggered the stop condition described in Section 3.14 and that contextualizes the router result of Section 4.3.

4.3 Outcome-aware routing: a sparsity/learnability boundary result

Table 4 reports regret relative to the Offline Oracle for the heuristic router (B2) and the learned router (B3) on held-out test seeds, together with each policy’s held-out escalation rate. **On every one of the four held-out cells tested (both configurations, both cost regimes), B3’s escalation rate is exactly 0.0%, and B3’s utility and regret-vs-oracle are numerically identical to B0’s, to full floating-point precision, independently re-verified against the archived raw output for this manuscript (e.g. original *small_dense*: $utility_{B3} = -0.012000010580089045$, $utility_{B0} = -0.012000010580089045$, identical to 15 significant figures; the same exact equality holds in all three other cells). The learned router did not learn a discriminating policy; it collapsed to the always-deterministic policy (Figure 3).**

The 90–152× (original regime) and 57–150× (revised regime) improvement of B3’s regret-vs-oracle over B2’s is numerically correct, but its interpretation must be stated precisely: because B3 is identical to B0 in every held-out cell, this

Table 2 Previous RACA reproduction, both cost regimes. “Orig.” = original cost regime (seeds 1000–1019); “Rev.” = revised, difficulty-widened regime (seeds 11000–11019).

Regime	Policy	Config	Compl. rate	Mean cost	Mean regret	Reas.-call rate
Orig.	B0	small_dense	0.968	3.94	0.0091	0.00
Orig.	B1	small_dense	0.969	3.89	0.0193	1.00
Orig.	B2	small_dense	0.969	3.84	0.0109	0.21
Orig.	B0	large_sparse	0.874	14.83	0.0039	0.00
Orig.	B1	large_sparse	0.875	14.72	0.0109	1.00
Orig.	B2	large_sparse	0.874	14.84	0.0067	0.27
Rev.	B0	small_dense	0.946	6.29	0.0296	0.00
Rev.	B1	small_dense	0.946	6.28	0.0338	1.00
Rev.	B2	small_dense	0.946	6.20	0.0288	0.60
Rev.	B0	large_sparse	0.771	27.64	0.0125	0.00
Rev.	B1	large_sparse	0.770	27.71	0.0310	1.00
Rev.	B2	large_sparse	0.769	27.81	0.0286	0.76

Table 3 Offline Oracle headroom, seed-level bootstrap ($n = 20$ seeds/regime).

Regime	Config	Value of reasoning	95% CI (seed-level)	Oracle escalation rate
Original	small_dense	+0.00115	[0.00076, 0.00158]	0.46%
Original	large_sparse	+0.00043	[0.00015, 0.00075]	0.14%
Revised	small_dense	+0.00508	[0.00399, 0.00633]	2.06%
Revised	large_sparse	+0.00251	[0.00187, 0.00322]	1.11%

improvement is entirely a statement that *a policy that never reasons at all beats B2’s heuristic by this margin*. It is not evidence that B3 learned anything about *which* decisions warrant reasoning. The router’s classifier (Section 3.5) never crosses its calibrated threshold τ on held-out, in-distribution data in any tested cell, so it achieves zero true positives among the Oracle’s rare positive-value decisions (the 0.14–2.06% of decisions identified in Section 4.2) and, trivially, zero false positives. This is consistent with a classifier collapsing to the majority class under severe label imbalance: the positive label rate for the binary classification task of Section 3.5 is bounded by the Oracle’s own escalation rate, so a model that predicts the majority class everywhere can achieve a low held-out loss without having learned to discriminate the rare positive cases at all. Phase 11 independently retrained a fresh router instance on disjoint seed pools and again found a large utility advantage over the B2 heuristic (independent B2 utility -0.3361 to -0.3375 ; independent B3 utility -0.0257 to -0.0297 , non-overlapping CIs). This is consistent with the same collapse-to-never-escalate outcome recurring, but the Phase 11 archived output does not record an escalation

rate for this instance, so this cannot be confirmed as directly as the four cells above; only the decision threshold τ was preserved for the Phase 11 retrained instances, not escalation rate or feature weights (Section 6). We report the Phase 11 utility values as consistent with, not as independent proof of, the same collapse mechanism.

4.4 Distribution shift

Transferring the router trained on *large_sparse* to *small_dense* is exactly lossless: shifted regret equals the in-distribution regret to six significant figures in both cost regimes (0% escalation). Transferring from *small_dense* to *large_sparse* degrades substantially: held-out escalation rises to 12.3% (original regime) or 13.85% (revised regime), and regret-vs-oracle grows relative to *large_sparse*’s own in-distribution performance by **66.0** $\times$ in the original cost regime ($0.0009314 \rightarrow 0.061481$) and **27.4** $\times$ in the revised, difficulty-widened regime ($0.0024385 \rightarrow 0.066726$); degradation is smaller, not larger, after the difficulty axis was deliberately widened. B3 still outperforms B2 in absolute terms under shift in both regimes. Because only two configurations were

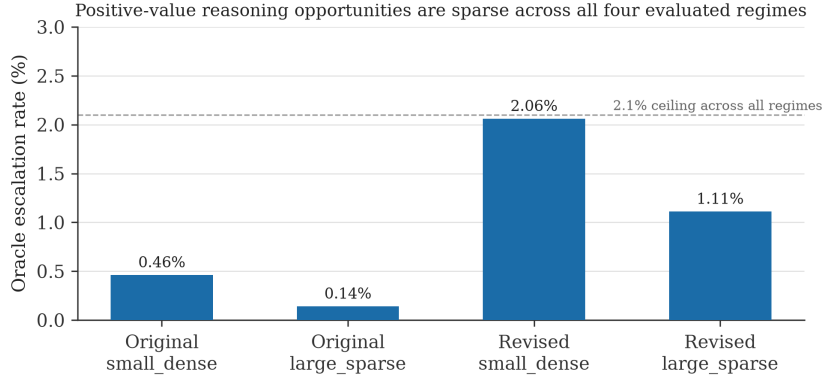


Fig. 2 Oracle escalation rate (point estimates, Table 3) across all four evaluated regime/configuration cells. Escalation stays below 2.1% even after deliberately widening the task-difficulty distribution (the “revised” regime), the central boundary result this paper’s structure is organized around.

Table 4 Regret relative to the Offline Oracle, held-out test seeds. B3’s regret column is identical to B0’s regret-vs-oracle in every cell (not shown as a separate column because the two are the same policy in this evaluation).

Regime	Config	Regret _{B2}	Regret _{B3} = Regret _{B0}	B3 escalation	B3 utility = B0 utility?
Original	small_dense	0.0968	0.00108	0.0%	Yes, exactly
Original	large_sparse	0.1410	0.00093	0.0%	Yes, exactly
Revised	small_dense	0.2885	0.00504	0.0%	Yes, exactly
Revised	large_sparse	0.3654	0.00244	0.0%	Yes, exactly

tested and never independently varied along a geometry axis versus a fleet-size axis, this study cannot decompose the degradation into a geometry effect and a fleet-size effect; only the combined shift is reported, and the decomposition is marked as an open limitation (Section 6).

4.5 Cognitive delegation

Table 5 reports the delegation ablation matrix. Once the routing gate is calibrated (reasoning-call rate 0.027–0.058%), enabling delegation, shared memory, or a centralized manager produces no statistically distinguishable change in fleet utility relative to independent agents (A1): conditions A3 through A7 all fall within overlapping confidence intervals of A1. Under a deliberately uncalibrated policy (reasoning-call rate 16–32%), delegation was measurably *harmful* (paired utility difference -0.0808 , 95% CI $[-0.0954, -0.0661]$, excluding zero).

No measurable benefit of delegation in calibrated homogeneous fleets was found (Figure 4). This is one of the paper’s central findings. The near-zero calibrated reasoning-call rate

leaves almost no reasoning calls for the delegation layer to act on, so a downstream mechanism for deciding *who* should perform those rare calls has negligible opportunity to influence fleet utility.

Under a controlled capability asymmetry (3 of 10 robots given lower failure probability and lower latency), the heterogeneous condition showed higher utility (homogeneous -0.2403 vs. heterogeneous -0.1541), with 97.5% of delegate-in activity concentrating on the three advantaged robots (versus 30% expected under uniform allocation). Because no matched no-delegation heterogeneous control was run, this result does not isolate whether the utility difference was caused by delegation itself or by the advantaged robots’ superior backend properties.

4.6 Shared memory

Once calibrated, all four memory conditions (C0–C3) are statistically indistinguishable (-0.02449 , overlapping CIs); under the uncalibrated policy, unbounded sharing (C1) was the worst condition, worse than no sharing at all (C0).

The learned router (B3) is numerically identical to the never-reason baseline (B0) in every held-out cell -- its large regret improvement over B2 reflects avoiding B2's over-escalation, not learned discrimination

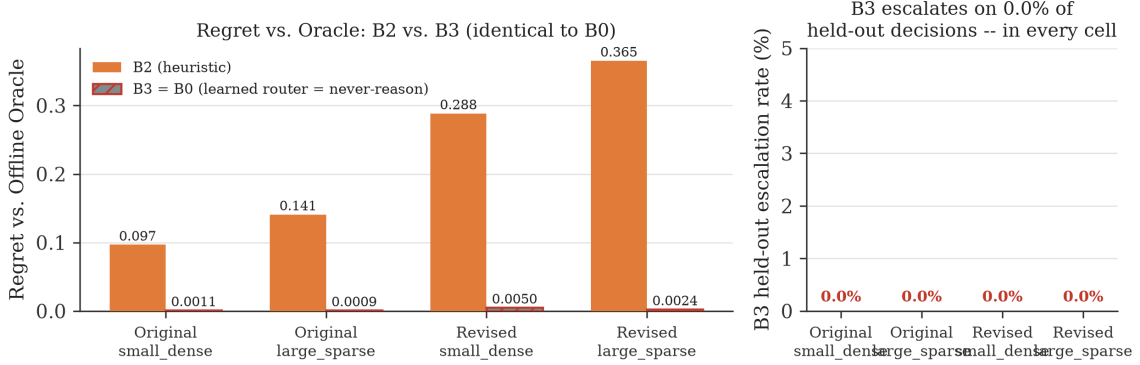


Fig. 3 The headline result of this paper, shown visually to prevent the regret-improvement figure from being misread as evidence of successful learned routing. Left: regret vs. Oracle for B2 (heuristic) and B3 (hatched, drawn identically to B0); B3's bar is not a separate measurement from B0's, since they are the same policy in every held-out cell. Right: B3's held-out escalation rate is exactly 0.0% in all four cells.

Table 5 Delegation ablation matrix, calibrated policy.

Condition	Mean utility	95% CI	Seed pool (n)
A0: independent, heuristic (B2)	-0.3361	[-0.3417, -0.3297]	23100-23107 (8)
A1: independent, learned (B3)	-0.0257	[-0.0283, -0.0233]	23100-23107 (8)
A3: distributed, no delegation	-0.0258	[-0.0284, -0.0235]	23100-23107 (8)
A4: delegation, frozen memory	-0.0258	[-0.0284, -0.0235]	23100-23107 (8)
A5: delegation, local memory	-0.0258	[-0.0284, -0.0235]	23100-23107 (8)
A6: delegation, shared memory	-0.0258	[-0.0284, -0.0235]	23100-23107 (8)
A2: centralized manager	-0.0297	[-0.0398, -0.0237]	25100-25109 (10)
A7: full system	-0.0300	[-0.0406, -0.0237]	25100-25109 (10)

4.7 Behavioral specialization

Under a deliberately uncalibrated delegation policy (distinct from the calibrated, near-inert policy behind Section 4.5), repeated interaction produces measurable concentration of delegate-in activity. The original, uncontrolled measurement showed a Gini coefficient of 0.7403 (± 0.0633 , $n = 6$ seeds). Two implementation artifacts were subsequently identified. The cleanest evidence for each comes from a **matched**, same-6-seed comparison (i.e. each control applied to the identical seed pool as its predecessor, isolating the marginal effect of one change at a time), which we report as the primary evidence; a separate, larger- n comparison exists but uses a different seed pool and is reported only as corroborating, not causal, evidence, since pool differences confound direct attribution:

1. A first-generation RNG-seeding pattern (`base_draw + robot_id`), matched-seed comparison: Gini 0.7403 $\rightarrow$ 0.7116 (± 0.0370 , same 6 seeds), a 3.9% relative reduction; the same robot remained the hub in 5 of 6 seeds even after this fix, so this artifact alone is a real but minor contributor.
2. Deterministic ascending-ID tie-breaking in `DelegationPolicy.decide()` (Section 3.7), matched-seed comparison on top of the RNG fix: Gini 0.7116 $\rightarrow$ 0.4691 (± 0.0710 , same 6 seeds) and hub identity decouples from robot ID. This matched 6-seed comparison is the direct evidence for tie-breaking being the dominant of the two artifacts; it rests on $n = 6$, and we do not claim a tighter interval than that small sample supports.

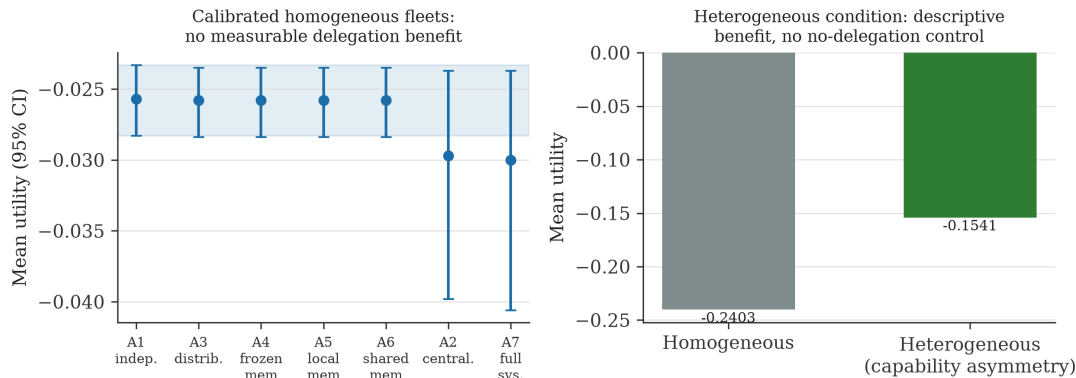


Fig. 4 Left: calibrated homogeneous-fleet conditions (Table 5) with 95% CIs; the shaded band is A1’s interval, and every other condition overlaps it. Right: the heterogeneous-condition utility comparison, shown descriptively. No matched no-delegation heterogeneous control exists, so this panel is not evidence that delegation itself caused the difference.

A separate, larger- n measurement exists at full statistical power ($n = 50$ seeds, a different pool from the 6-seed matched comparison above, with all three controls, randomized IDs, shuffled order, and randomized tie-breaking, applied jointly): Gini 0.478 (Section 4.8). Comparing this $n = 50$ value to the original $n = 6$ uncontrolled baseline (0.7403) gives a 35.4% relative reduction; because the two measurements come from different seed pools, this cross-pool comparison is a **descriptive** reduction figure, not a causal attribution of the joint-controlled reduction to the two named artifacts specifically (other, unidentified factors could in principle differ between the two pools). This descriptive reduction is reported as 35.4%, not as a larger fraction; the matched, same-pool 6-seed comparison above (0.7403 $\rightarrow$ 0.4691, a 36.6% relative reduction) is closer to a directly causal figure but carries its own small- n caveat.

4.8 Artifact controls and residual specialization

After controlling jointly for randomized robot IDs, shuffled processing order, and randomized tie-breaking, a residual concentration remains and is statistically distinguishable from chance (Table 6, Figure 5; $n = 50$ seeds, 30000–30049), compared against a uniform-random-peer null model. Role persistence in Table 6 is computed as a tie-corrected (average-rank) Spearman correlation between consecutive 200-decision windows, recomputed directly from archived per-window per-robot delegate-in-share data; the originally

implemented rank correlation did not correct for ties, which matters here because per-robot shares are frequently tied at exactly zero in early windows. The tie-corrected and original values are close (real: 0.881 tie-corrected vs. 0.875 originally reported; null: 0.269 vs. 0.263) and the qualitative conclusion (non-overlapping confidence intervals between real and null) is unchanged either way; Table 6 reports the tie-corrected values as the more methodologically correct figure.

No detectable difference (overlapping confidence intervals) is not treated as proof of equality anywhere in this section or elsewhere in the paper (Section 3.13); the non-overlapping intervals reported in Table 6 are the basis for calling this residual concentration statistically distinguishable from the null model, not an overlap-based argument.

Five candidate explanatory mechanisms beyond the two named artifacts were tested and ruled out: early-round success-rate “rich-get-richer” dynamics ($r = 0.086$, $\rho = -0.063$, $n = 227$), whole-episode success-rate correlation ($r = 0.086$ – 0.154 , $\rho = 0.046$ – 0.047), unbounded versus windowed peer-memory accumulation (overlapping CIs), queue-position/decision-opportunity count (shown circular via a confound with `unavailable_count`, $r = 0.826$), and communication topology, task proximity, and spawn geometry (ruled out by construction, since none of these concepts exist in the simulator).

The residual is instead **path-dependent**. The intervention tested is specific and should be described precisely: at a single early round

Table 6 Fully-controlled residual specialization vs. null model ($n = 50$ seeds). Persistence uses tie-corrected Spearman rank correlation.

Condition	Gini	95% CI	Persistence	95% CI
Real, fully controlled	0.478	[0.446, 0.512]	0.881	[0.853, 0.903]
Null model	0.091	[0.064, 0.126]	0.269	[0.193, 0.345]

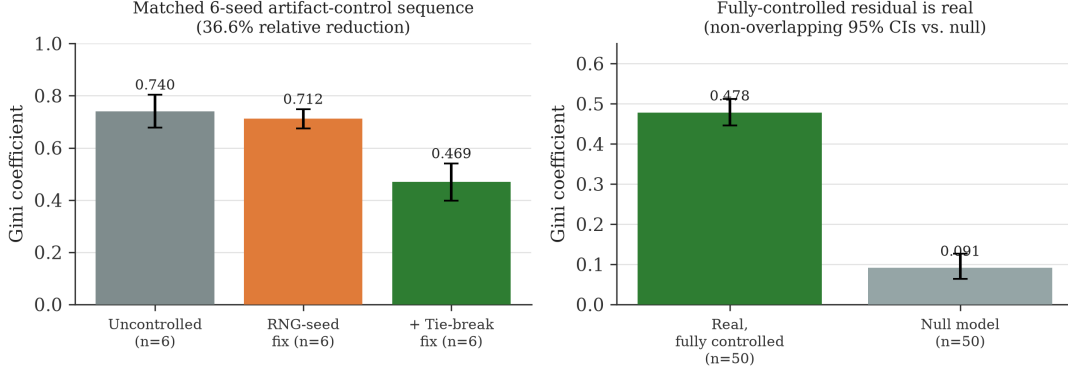


Fig. 5 Left: the matched, same-6-seed artifact-control sequence of Section 4.7, isolating the marginal effect of each fix. Right: the fully-controlled $n = 50$ residual (Table 6) against the uniform-random-peer null model. Non-overlapping confidence intervals establish the residual as real, not solely artifactual, though it still does not satisfy the emergence criteria of Section 4.11.

where two or more peers were in an exact expected-utility tie (an “indifference” point, where the argmax rule’s output is determined by tie-breaking rather than by any quality difference between candidates), the tie-break outcome was reversed, changing which peer was selected at that one decision; the fleet’s subsequent trajectory was then re-simulated forward under the same random stream from that point. This single reversed tie-break flips which robot becomes the overall hub by the end of the episode in 70.0% of seeds (21/30), while the overall magnitude of concentration remains comparatively stable. A follow-up injection-depth sweep shows the flip rate decaying with the round at which the tie is injected: 0.621, 0.409, and 0.143 at rounds 0, 200, and 400 respectively (29, 22, and 14 of 30 seeds valid at each depth, since exact ties become mechanically rarer as the episode progresses). The round-600 cell has only 7 valid seeds (point estimate 0.286, 95% CI [0.0, 0.571]) and is reported as **inconclusive**, not as a trend-breaking data point, since its wide interval is consistent with both neighboring estimates. A two-regime (indifference/differentiated) mathematical model was developed *after* this path-dependence pattern was

first observed, as a post-hoc mechanistic explanation for it; its two derived predictions (the flip-rate decay above, and a Gini-vs-peer-capacity relationship) were then evaluated against fresh, disjoint seeds not used to originally observe the pattern (flip-rate sweep: seeds 40000–40029; peer-capacity sweep: seeds 42000–42019), and both predictions were confirmed in that fresh-seed evaluation. The peer-capacity sweep found Gini of 0.207, 0.422, 0.489, 0.503, 0.503 at capacities 1, 2, 3, 5, 10 respectively; capacities 5 and 10 agree to four significant figures in this configuration, which we report as an observed plateau between capacities 5 and 10. We do not claim this establishes a general theoretical saturation threshold at $N - 1$ peers, only that the response plateaued in this specific tested range and fleet size.

4.9 Perturbation and reorganization

After removing the current hub robot, mean fleet utility measured in the rounds that follow is 0.0226 (± 0.0172 , 95% CI [0.0165, 0.0288], $n = 30$) lower than before removal, and a new top delegate-in-recipient is observed in every seed (30/30). Two caveats limit the causal interpretation of both figures: no matched no-removal control trajectory

was run over the same time window, so the utility decline cannot be distinguished from ordinary episode-to-episode drift without a counterfactual baseline; and because removing the fleet’s current top delegate-in recipient mechanically guarantees that *some other* robot becomes the new top recipient among those remaining, the 30/30 rate is not by itself evidence of active reorganization; it is close to a structural inevitability of the measurement. What the data do support is that concentration among the remaining robots becomes more diffuse after removal (Gini 0.7744 $\rightarrow$ 0.6975, non-overlapping CIs) rather than being cleanly reassigned to a single successor; this is reported descriptively as diffuse load redistribution, not as evidence of functional role replacement or causally-attributed “spontaneous reorganization.”

4.10 Scaling

Across fleet sizes $N \in \{10, 25, 50, 100\}$ (4 seeds/size), Gini weakens slightly with scale (0.703, 0.664, 0.655, 0.646) while mean utility worsens with decelerating magnitude. $N = 5$ and $N = 20$ were never tested, so this is not a continuous sweep. A separate comparison of centralized (A2) and distributed (A3) architectures at $N = 10$ and $N = 50$ found statistically indistinguishable utility (A2: $-0.0262/-0.0165$; A3: $-0.0254/-0.0159$) and zero non-deterministic task assignments traced across a full 1500-decision episode; the only clean structural difference is a fixed communication cost (A2 pays 2 messages/decision at every N tested; A3 pays 0). No centralized/distributed performance breakpoint was identified in the tested range, and this is reported as insufficient evidence for a breakpoint, not as evidence against one.

4.11 Emergence audit

Table 7 applies the eight predefined criteria of Section 3.11 to the cumulative evidence above.

Verdict: the observed specialization does not satisfy the criteria for emergent collective cognition (NO). Criterion 6 is the binding failure: even where structure is real, reproducible, and persistent, it delivers no measurable fleet-level benefit under the fleet’s calibrated operating point, and criteria 2, 7, and 8 are only partially satisfied once designed capability effects and implementation artifacts are excluded.

5 Discussion

Can robots learn when reasoning is valuable?

Not demonstrated by this study. The learned router avoided the heuristic’s over-escalation and thereby achieved much lower regret relative to the Oracle, but in every in-distribution held-out cell it did so by collapsing to a never-reason policy identical to B0, not by identifying the sparse decisions for which reasoning genuinely helped (Section 4.3). The 57–152 $\times$ regret improvement over the heuristic is real and independently replicated, but it demonstrates that *avoiding gratuitous over-escalation is valuable*, not that *outcome-aware discrimination of individual decisions is learnable* under this environment’s opportunity distribution.

How much opportunity actually exists for routing?

Not much, and this is now understood to be causally connected to the router’s collapse, not merely a contextual caveat on its result: Oracle escalation stays below 2.1% of decisions even after deliberately widening the task-difficulty distribution, and a positive-label rate this low is a regime in which a binary classifier can trivially minimize held-out loss by predicting the majority class everywhere. This scale should anchor the reading of the router result above, and this ordering (Oracle before router) is deliberate in this manuscript’s structure for that reason.

What does the router’s collapse reveal about outcome-aware reasoning allocation in general?

A sparsity/learnability boundary: when the fraction of decisions for which reasoning genuinely has positive value is small enough, a learned router can appear to succeed on the standard evaluation metric (regret relative to a strong baseline) purely by defaulting to inaction, without any evidence that it discriminates the rare positive cases the metric is meant to reward it for finding. Distribution shift (Section 4.4) is informative here: escalation is not identically zero under shift (12.3–13.85%), which shows the classifier is not a constant function (it does respond to sufficiently different inputs), but under the in-distribution held-out evaluation

Table 7 Emergence criteria audit.

Criterion	Evidence	Verdict
1. Not explicitly programmed	No role-assignment code exists; concentration arises from an unmodified argmax rule	PASS
2. Local-interaction origin	Path dependence supported (Sec. 4.8); capability-driven differentiation (heterogeneous case) not addressed by this criterion	INCONCLUSIVE
3. Macroscopic structure	Real, reduced-magnitude concentration confirmed vs. null model	PASS (reduced magnitude)
4. Cross-seed reproducibility	Non-overlapping CIs vs. null, $n = 50$	PASS
5. Temporal persistence	Role persistence 0.881 vs. null 0.269 (tie-corrected), non-overlapping	PASS
6. Measurable, useful system-level consequence	No positive benefit under the calibrated policy (Table 5); negative only under the uncalibrated one	FAIL
7. Not solely artifact	~35% of original magnitude traced to two identified artifacts; residual survives	PARTIAL
8. Survives controls	Survives ID/order/tie-break controls jointly, at reduced magnitude	PARTIAL

that matters for this paper’s headline comparison, that responsiveness never activates. Whether this is a property of this specific label-imbalance regime, this specific classifier family, or this specific calibration procedure is not established by this study and is left as a direction for future work (Section 7).

Why did delegation fail to improve homogeneous fleets?

Because the calibrated routing gate produces a reasoning-call rate below 0.06%, leaving the delegation layer with very few opportunities to influence fleet utility. This makes the homogeneous null result mechanistically unsurprising rather than evidence that delegation is useless in general. The heterogeneous condition shows higher utility and delegation activity concentrating on the advantaged robots, but because no matched no-delegation heterogeneous control was run, this study cannot isolate whether the difference was caused by delegation itself or by the advantaged robots’ superior backend properties.

Why can specialization appear without producing collective benefit?

Because the mechanism producing it, a hard-argmax winner-take-all dynamic over peer expected utilities that is sensitive to early tie-breaking, is a structural property of the delegation rule itself, not a signal that the fleet is discovering a useful division of labor. The same mechanism

that produces the observed Gini concentration operates whether or not that concentration helps task performance, and under calibration it does not.

Why is specialization not equivalent to emergence?

Because emergence, as defined here, additionally requires a measurable, useful system-level consequence (criterion 6) and a demonstration that the structure is not substantially explained by implementation artifacts (criteria 7–8). Both are the specific points at which this study’s evidence falls short, not a general skepticism about the possibility of emergence in principle.

What does low Oracle headroom imply for future LLM-integrated robotics research?

Routing sophistication cannot manufacture headroom that the underlying decision distribution does not contain. A more expressive router architecture, evaluated in this same environment, would still be bounded by the same Oracle ceiling; the more productive question for a follow-up study is whether a different task-difficulty distribution genuinely contains more exploitable routing opportunity, not whether a better router can extract more value from this one.

6 Limitations

- The learned router (B3) did not learn to discriminate positive-value reasoning decisions; on

every in-distribution held-out cell it is numerically identical to the always-deterministic baseline. The 57–152× regret improvement over the heuristic router should not be read as evidence of learned outcome-aware routing (Section 4.3).

- The previous RACA project’s quantitative advantage did not reproduce; two contributing factors were identified but the original internal system’s exact backend/cost parameterization could not be recovered for direct comparison, and no independently citable external reference for that system exists.
- The Oracle stop condition was triggered but not honored (Section 3.14); every result beyond Sections 4.1–4.2 carries exploratory, not confirmatory, evidentiary weight.
- The router’s regret-vs-oracle confidence intervals (Table 4) were computed by bootstrapping pooled per-decision observations rather than seed-level means, a pseudo-replication concern that could not be corrected without regenerating per-decision data. A bounded root-cause audit (Section 3.13) found the computation-path code unchanged since the results were archived and the current environment internally self-consistent, isolating the most probable cause to an unrecorded, unpinned numpy/Python runtime version active at the time of original generation. This is a class of gap this repository’s own history shows was recognized and fixed only prospectively (dependency pinning was added in a later commit, not retroactively applicable to already-archived results). Table 4’s values are archived and verified from stored output, but not currently regenerable from this repository’s present state.
- Feature weights for the independently-retrained router instances used in the delegation ablation matrix (A1, A2, A7) were not preserved; only the decision threshold was recorded, so feature-importance conclusions rely on the original router instance only, and Phase 11’s independent replication of the collapse-to-never-escalate outcome (Section 4.3) is inferred from utility values rather than a directly recorded escalation rate.
- The 35.4% artifact-attribution figure for specialization (Section 4.7) is a descriptive, cross-seed-pool comparison, not a matched causal

estimate; the matched, same-pool 6-seed comparison (36.6% relative reduction from tie-break randomization alone) is closer to causal but rests on a small sample.

- The heterogeneous-delegation utility benefit (Section 4.5) cannot be causally attributed to delegation itself, since no matched no-delegation heterogeneous control was run; the advantaged robots’ superior backend properties alone could explain some or all of the observed benefit.
- Perturbation results (Section 4.8) lack a matched no-removal control trajectory, so the measured utility decline after hub removal cannot be fully separated from ordinary episode-to-episode variation; the 30/30 new-top-robot rate is partly a structural consequence of the measurement (removing the top recipient guarantees a different robot becomes top among those remaining) rather than purely evidence of active reorganization.
- Distribution-shift degradation was measured only as a combined effect of task geometry and fleet density changing together; the two were never independently varied, so no geometry-versus-fleet-size decomposition is offered.
- Task-type specialization and network centrality were never measured; specialization claims in this paper are limited to delegate-in concentration and persistence only.
- The round-600 injection-depth measurement rests on 7 valid seeds and is reported as inconclusive.
- Fleet sizes $N = 5$ and $N = 20$ were never tested; the scaling sweep is not continuous.
- No centralized/distributed performance breakpoint was identified in the tested range ($N = 10$ –50); this is reported as insufficient evidence, not as evidence of no breakpoint at any scale.
- A minimum-population threshold for the observed specialization effect was not determined; $N = 10$, the smallest size tested, already shows the effect.
- Peer access was held fixed as all-to-all throughout; communication-topology sensitivity was never evaluated.
- The study evaluated multiple outcomes, ablations, and exploratory comparisons; because a complete prospective family of hypothesis tests was not defined, individual exploratory p -values should not be interpreted as family-wise

confirmatory evidence, and no formal multiple-comparison correction was applied. Conclusions throughout are drawn from effect sizes and cross-seed consistency rather than isolated statistical tests.

- All results are from a pure-Python/NumPy simulator; none of this work has been validated against a physics-based simulator (e.g. ROS2/Gazebo) or a physical robot fleet.

Path to physical validation.

Although the present evaluation is simulation-based, the architecture is intentionally separated from robot-specific control. The routing and delegation layers operate on abstract decision-state, latency, reliability, and peer-capability signals rather than simulator-specific physics. A practical transition path is therefore to retain the same cognitive decision layer while replacing the lightweight simulator with ROS 2/Gazebo interfaces, followed by hardware-in-the-loop and physical multi-robot validation. The primary quantities requiring re-estimation in deployment are backend latency, communication delay, failure probability, and task-specific utility, while the routing and delegation logic itself remains unchanged.

7 Future Work

Future work motivated directly by the above limitations includes: environments deliberately constructed to contain larger Oracle routing headroom, to test whether the router result of Section 4.3 generalizes to a regime where the absolute opportunity is larger and whether the classifier-collapse pattern persists, weakens, or disappears as positive-label prevalence increases; class-imbalance-aware training procedures (e.g. class weighting, oversampling of positive-value decisions, precision/recall-based threshold selection rather than aggregate-utility-based threshold selection) to test whether the collapse is a property of this environment’s sparsity or of this specific training procedure; explicit task-type classes to test for semantic (not only statistical) specialization; a continuous fleet-size sweep including $N = 5$ and $N = 20$ to resolve the minimum-population question; a wider delegation-calibration grid, since the current calibration optimum sits at the edge of the searched range on two of three parameters; physics-based and physical-robot validation;

and an independently varied geometry/fleet-size distribution-shift design to decompose the combined effect measured here.

8 Conclusion

The Offline Oracle establishes that positive-value reasoning opportunities are intrinsically sparse in the evaluated environment. Within that sparse-opportunity regime, a learned router avoided a hand-tuned heuristic’s substantial over-escalation and thereby achieved much lower regret relative to the Oracle, but on every in-distribution held-out evaluation it did so by collapsing to a policy numerically identical to always-deterministic, not by learning to discriminate the sparse decisions for which reasoning had positive value. This is best understood as a sparsity/learnability boundary result: under sufficiently rare positive-value events, a learned router can score well on an aggregate regret metric purely by defaulting to inaction, and that outcome must not be mistaken for evidence of successful outcome-aware routing. Extending reasoning allocation to peer delegation produces no measurable benefit in calibrated homogeneous fleets; it shows a benefit under designed capability heterogeneity, though this study cannot isolate whether that benefit is caused by delegation itself or by the advantaged robots’ properties alone, since no matched no-delegation heterogeneous control was run. Repeated interaction under an uncalibrated delegation policy nevertheless produces real, path-dependent, artifact-controlled behavioral concentration in which some robots receive disproportionately more delegated work than others. A matched comparison attributes roughly one third of an originally larger apparent effect to two identified implementation artifacts, not the majority, though the full cross-pool reduction (35.4%) is descriptive rather than strictly causal. This specialization does not satisfy the pre-defined criteria for emergent collective cognition, chiefly because it delivers no measurable system-level benefit once the routing gate is calibrated. The paper’s main conclusion is therefore twofold: first, that this environment exposes a genuine sparsity/learnability boundary for outcome-aware reasoning allocation, where the standard evaluation metric cannot on its own distinguish a router that has learned discrimination from one that

has learned only to abstain; and second, that outcome-aware reasoning allocation, cognitive delegation, behavioral specialization, and collective emergence remain empirically separable phenomena, none of which should be assumed, without separate evidence, to accompany the others.

Declarations

- **ORCID.** Pouya Mansournia: 0009-0003-1515-6550.
- **Funding.** No external funding was received for this study.
- **Competing interests.** The author declares no competing interests.
- **Ethics approval.** Not applicable. This study consists entirely of software simulation; it involves no human participants, no animal subjects, and no physical experimentation.
- **Consent to participate.** Not applicable; no human participants are involved in this study.
- **Consent for publication.** Not applicable; no identifiable individuals are involved in this study.
- **Availability of data and materials.** All simulation configurations, per-seed and summary result files, and analysis scripts backing every table and figure in this manuscript are available in a public repository at <https://github.com/Pouya-Mansournia/RACA-Collective-Public>, archived at Zenodo under DOI [10.5281/zenodo.21960745](https://doi.org/10.5281/zenodo.21960745). One result (Table 4, Phase 3 router regret-vs-oracle) is archived and independently verified from stored output but, per the bounded root-cause audit described in Section 3.13, is not currently exactly regenerable from the repository’s present environment; this is disclosed there and in the accompanying reproducibility appendix rather than presented as fully reproducible.
- **Code availability.** All simulator, backend, router, delegation-policy, and analysis code used to produce the results in this manuscript is available in the same public repository referenced above.
- **Authors’ contributions.** The sole author conceived the study, developed the methodology and implementation, ran and analyzed all experiments, and wrote the manuscript.
- **Acknowledgements.** None.

- **Use of AI-assisted tools.** Portions of this manuscript’s drafting, revision, and verification process used AI-assistant tools under direct human scientific direction and review; all scientific claims, evidence verification, and final editorial decisions were made and are attested to by the author. No generative AI tool is listed as an author.

References

- Bayati, M., Hamidi, N., Johari, R., Khosravi, K. (2020). The unreasonable effectiveness of greedy algorithms in multi-armed bandit with many arms. *arXiv preprint arXiv:2002.10121*, , (NeurIPS 2020)
- Casadei, R. (2023). Artificial collective intelligence engineering: A survey of concepts and perspectives. *Artificial Life*, 29(4), 433–467, <https://doi.org/10.1162/artl.a.00408> (arXiv:2304.05147)
- Chen, S., Liu, Y., Han, W., Zhang, W., Liu, T. (2024). A survey on LLM-based multi-agent system: Recent advances and new frontiers in application. *arXiv preprint arXiv:2412.17481*, ,
- Chen, Y., Arkin, J., Zhang, Y., Roy, N., Fan, C. (2024). Scalable multi-robot collaboration with LLMs: Centralized or decentralized systems? *Ieee international conference on robotics and automation (icra)*. (arXiv:2309.15943)
- De Sabbata, C.N., Summers, T.R., AlKhamissi, B., Bosselut, A., Griffiths, T.L. (2024). Rational metareasoning for large language models. *arXiv preprint arXiv:2410.05563*, ,
- Dias, M.B., Zlot, R., Kalra, N., Stentz, A. (2006). Market-based multirobot coordination: A survey and analysis. *Proceedings of the IEEE*, 94(7), 1257–1270, <https://doi.org/10.1109/JPROC.2006.876939>

- Gerkey, B.P., & Mataric, M.J. (2002). Sold!: Auction methods for multirobot coordination. *IEEE Transactions on Robotics and Automation*, 18(5), 758–768, <https://doi.org/10.1109/TRA.2002.803462>
- Goel, H., Omama, M., Chalaki, B., Tadiparthi, V., Moradi Pari, E., Chinchali, S.P. (2025). R3DM: Enabling role discovery and diversity through dynamics models in multi-agent reinforcement learning. *International conference on machine learning (icml)*. (arXiv:2505.24265)
- Hay, N., Russell, S., Tolpin, D., Shimony, S.E. (2018). Computing the value of computation for planning. *arXiv preprint arXiv:1811.03035*, ,
- Hu, Z., Zhang, Z., Li, H., Chen, C., Ding, H., Wang, Z. (2024). ACORM: Attention-guided contrastive role representations for multi-agent reinforcement learning. *International conference on learning representations (iclr)*. (arXiv:2312.04819)
- Jamshidpey, A., Wahby, M., Allwright, M., Zhu, W., Dorigo, M., Heinrich, M.K. (2024). Centralization vs. decentralization in multi-robot coverage: Ground robots under UAV supervision. *arXiv preprint arXiv:2408.06553*, ,
- Ma, H., Hu, T., Pu, Z., Liu, B., Ai, X., Liang, Y., Chen, M. (2024). Coevolving with the other you: Fine-tuning LLM with sequential cooperative multi-agent reinforcement learning. *Advances in neural information processing systems (neurips)*. (arXiv:2410.06101)
- Mitzenmacher, M. (2001). The power of two choices in randomized load balancing. *IEEE Transactions on Parallel and Distributed Systems*, 12(10), 1094–1104, <https://doi.org/10.1109/71.963420>
- Moslem, Y., & Kelleher, J.D. (2026). Dynamic model routing and cascading for efficient LLM inference: A survey. *arXiv preprint arXiv:2603.04445*, ,
- Ong, I., Almahairi, A., Wu, V., Chiang, W.-L., Wu, T., Gonzalez, J.E., ... Stoica, I. (2025). RouteLLM: Learning to route LLMs with preference data. *International conference on learning representations (iclr)*. (arXiv:2406.18665)
- Riedl, C. (2026). Emergent coordination in multi-agent language models. *International conference on learning representations (iclr)*. (arXiv:2510.05174)
- Rothschild, M. (1974). A two-armed bandit theory of market pricing. *Journal of Economic Theory*, 9(2), 185–202, [https://doi.org/10.1016/0022-0531\(74\)90066-0](https://doi.org/10.1016/0022-0531(74)90066-0)
- Russell, S., & Wefald, E. (1991). Principles of metareasoning. *Artificial Intelligence*, 49(1-3), 361–395, [https://doi.org/10.1016/0004-3702\(91\)90015-C](https://doi.org/10.1016/0004-3702(91)90015-C)
- Shida, Y., Jimbo, T., Odashima, T., Matsubara, T. (2025). Reinforcement learning of multi-robot task allocation for multi-object transportation with infeasible tasks. *Ieee/sice international symposium on system integration (sii)*. (arXiv:2404.11817)
- Slivkins, A., Xu, Y., Zuo, S. (2025). Greedy algorithm for structured bandits: A sharp characterization of asymptotic success/failure. *arXiv preprint arXiv:2503.04010*, , (NeurIPS 2025)
- Snell, C., Lee, J., Xu, K., Kumar, A. (2024). Scaling LLM test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*, ,

Wang, T., Dong, H., Lesser, V., Zhang, C.
(2020). ROMA: Multi-agent reinforcement learning with emergent roles. *International conference on machine learning (icml)*. (arXiv:2003.08039)