From Trace Entropy to Coordination Control: Diagnosing and Intervening in Multi-Agent LLM Systems

Chinmoy Mohapatra* **Suyash Vishnoi** **Abhilasha Katariya**
Rohit Malshe **Julian Pachon**

Amazon

Abstract

Coordination interventions for multi-agent LLM systems are regime-dependent: identical guidance improves accuracy under greedy decoding but degrades it under stochastic decoding. We trace the reversal to over-regularization: exploration-promoting interventions add explicit entropy to policies already receiving implicit entropy from temperature, pushing effective diversity past the optimum. This leads to a decomposition of the intervention space into regime-dependent types (exploration, commitment) and regime-invariant types (consistency, disruption). We derive the Transfer Coefficient, a per-step information-gain diagnostic that detects coordination pathologies from the agent's own trace, and use it to build TRAC (Trace-based Regime-Aware Coordination). TRAC implements the regime-appropriate composition through two gated mechanisms: static self-trend constraints for failures the agent can self-diagnose, and a dynamic monitor for failures requiring external trajectory observation. Neither component alone reliably outperforms the baseline, but they address disjoint failure modes and their gated combination achieves 82.0% on GAIA versus 75.8% without intervention (+6.2%, McNemar p=0.004).

1 Introduction

Multi-agent LLM systems coordinate specialized sub-agents through routing decisions that determine which agent acts at each deliberation step. When coordination fails (loops, premature commitment, redundant search), the failure remains invisible in the final output. The practitioner observes only a wrong answer, with no signal about which decision was suboptimal or why. Without process-level diagnostics, system improvement reduces to trial-and-error prompt engineering.

Prior work has made progress on failure diagnosis: MAST (Cemri et al., 2025) provides a taxonomy of 14 failure modes, SkillTracer (Li et al., 2026) traces failures to their origin in multi-step execution plans, and Zhu et al. (2026) characterize orchestrator entropy dynamics. These approaches characterize failure post-hoc but do not prescribe interventions, nor do they account for the operating regime in which the failure occurs.

We show that the correct intervention depends on the operating regime. The same coordination guidance ("be aware of failure patterns; try different strategies when stuck") improves accuracy under greedy decoding but degrades it under stochastic decoding, with no other change. The mechanism, formalized through maximum-entropy RL (Haarnoja et al., 2018), is over-regularization: stochastic decoding already introduces diversity into the agent's actions, and exploration-promoting guidance adds further diversity, pushing total exploration past the optimum and suppressing correct early hypotheses.

We develop TRAC (Trace-based Regime-Aware Coordination), a system that makes the observation actionable through three elements. First, the Transfer Coefficient $T(t)$, a per-step information-gain metric derived from the agent's coordination trace, provides real-time

*Corresponding author: cmohapat@amazon.com

diagnostics that predict failure from early deliberation steps (§2). Second, a regime-aware decomposition classifies interventions into types that are safe at all temperatures versus types whose effect reverses with the decoding regime (§3). Third, a gated two-component system deploys static prompt constraints for failures the agent can self-diagnose, and activates a dynamic callback monitor only for failures requiring external trajectory analysis (§4).

On the GAIA benchmark (Mialon et al., 2023) with two independent stochastic seeds, TRAC achieves 82.0% task-solve accuracy versus 75.8% without intervention (+6.2%, McNemar p=0.004) while using fewer deliberation steps per task. Ablation confirms that neither component alone reliably outperforms the baseline; their gated combination does, because each uniquely solves tasks the other cannot.

Relation to prior work. Existing self-correction methods operate at different granularities: Reflexion (Shinn et al., 2023) reflects between episodes, LATS (Zhou et al., 2024) backtracks via MCTS within episodes, CRITIC (Gou et al., 2024) verifies post-generation, Inner Monologue (Huang et al., 2023) injects environment feedback between embodied actions, and AgentMonitor (Chi et al., 2024) monitors for safety. Our monitor operates within a single LLM execution at zero additional cost, using the agent's own trace metrics rather than environment signals, with a theory of when feedback helps versus harms. On the information-theoretic side, semantic entropy (Kuhn et al., 2023) detects hallucination, Setlur et al. (2025) reward information gain, and process reward models (Lightman et al., 2024) score reasoning steps but require supervised step labels. MaxEnt RL (Haarnoja et al., 2018) provides our theoretical grounding for regime-dependence. TRAC requires only final-answer correctness, no step supervision, and improves both accuracy and efficiency, relevant to compute-optimal inference (Snell et al., 2024).

2 Diagnostic Framework

Multi-agent LLM coordination is a sequential decision process: at each step the coordinator selects an agent, formulates a hypothesis, and updates its belief based on evidence from search and computation. Under stochastic decoding every decision point exhibits randomness; under greedy decoding the process is deterministic but may still fail through poor routing. To make this process observable, we instrument the coordinator to emit a structured state at each deliberation step: $s_t = (h_t, c_t, E_t^+, E_t^-, M_t, a_t)$, comprising the current hypothesis h_t (embeddable via $e : \mathcal{V} \to \mathbb{R}^d$), self-assessed confidence $c_t \in [0, 1]$, supporting and contradicting evidence $E_t^{\pm}$, unresolved sub-questions M_t, and the next agent selected a_t. A trace $\xi = (s_1, \ldots, s_T)$ makes coordination fully observable without requiring plan graphs (Li et al., 2026) or post-hoc annotation (Cemri et al., 2025), and serves as input to both offline analysis and the real-time monitor described in §4.

2.1 Established Entropy Quantities and Their Interaction Tensions

Given an observable trace, we seek a per-step diagnostic that detects coordination failures in real time. A coordination trace exhibits both discrete routing decisions (which agent acts next) and continuous semantic evolution (how hypotheses change), and each aspect admits an entropy-based measure. We define three such measures and show why none suffices individually, and why their naive combination fails due to interaction tensions.

Structural Entropy H_s. We measure routing diversity as the conditional entropy of the first-order Markov chain over agent transitions. Zhu et al. (2026) use this quantity to characterize orchestrator behavioral types; we adopt it as one axis of our diagnostic. High H_s indicates the coordinator distributes work across many agents; low H_s indicates repetitive use of the same agent.

$$H_s(\xi) = -\sum_{i,j \in \mathcal{A}} \hat{P}(a_{t+1}{=}j \mid a_t{=}i)\, \log \hat{P}(a_{t+1}{=}j \mid a_t{=}i) \tag{1}$$

Semantic Vector Entropy H_v. We measure hypothesis dispersion by clustering the sequence of hypothesis embeddings and computing the entropy over cluster occupancy. This generalizes the semantic equivalence classes of Kuhn et al. (2023), who group generations by meaning for hallucination detection, to coordination traces where the "generations" are successive hypotheses within a single task. High H_v indicates the agent explored many distinct hypotheses; low H_v indicates focused reasoning around a single idea.

$$H_v(\xi) = -\sum_{k=1}^{K} p_k \log p_k \tag{2}$$

where p_k is the fraction of hypothesis embeddings in cluster k.

Semantic Flux Φ. We measure the rate of semantic change as the mean embedding displacement between consecutive states. High Φ indicates large semantic jumps; low Φ indicates the hypothesis is barely changing.

$$\Phi(\xi) = \frac{1}{T-1} \sum_{t=2}^{T} \|e(h_t) - e(h_{t-1})\| \tag{3}$$

Why these are jointly insufficient. The naive interpretation (low H_s means decisive routing, low H_v means focused reasoning, high Φ means fast learning) fails because the quantities interact.

The H_v–Φ tension. A correctly-solved multi-hop task that progresses through entity identification, date retrieval, and format verification visits 3–4 semantic clusters (high H_v) with large embedding jumps (high Φ). An agent thrashing between wrong hypotheses produces the same signature. Neither metric captures whether movement was directed toward the answer.

The H_s conflation. By the chain rule, $H(a_{t+1} \mid a_t) = H(a_{t+1} \mid a_t, x_t) + I(a_{t+1}; x_t \mid a_t)$. The first term is coordination noise (random agent switching); the second is intelligent specialization (routing to `search` when needing facts, `code` when needing computation). Minimizing raw H_s penalizes both, discouraging the very context-sensitivity that makes multi-agent coordination valuable.

2.2 From Aggregate Metrics to Per-Step Diagnostics

The aggregate metrics cannot pinpoint when coordination failed, in which direction the agent moved, or whether a routing decision was appropriate. Three transformations address these gaps. **Temporal decomposition** localizes each aggregate to individual steps: $h_s(t) = -\log \hat{P}(a_t \mid a_{t-1})$, $h_v(t) = \|e(h_t) - \bar{\mu}_{1:t-1}\|$, $\phi(t) = \|e(h_t) - e(h_{t-1})\|$. **Directional projection** resolves the H_v–Φ tension by projecting movement onto the question direction: $\Phi_d(t) = \cos(\delta_t,\ \mathbf{e}_q - e(h_t))$ where $\delta_t = e(h_t) - e(h_{t-1})$ and $\mathbf{e}_q = e(q)$; positive values indicate progress toward the answer. **Task-conditioning** measures absolute position via the Hypothesis Drift Score $\text{HDS}(t) = 1 - \cos(e(h_t), \mathbf{e}_q)$. Together, Φ_d and HDS distinguish productive exploration (far but heading toward) from stagnation (near but circling).

2.3 The Transfer Coefficient

The ideal per-step diagnostic captures how much each deliberation step reduces uncertainty about the correct answer a^*:

$$\mathcal{I}_t = H(a^* \mid \xi_{1:t-1}) - H(a^* \mid \xi_{1:t}) \tag{4}$$

This is the directed information transfer per step (Schreiber, 2000): positive when a step brings the system closer to the answer, zero when wasted, negative when it increases confusion.

Computing $\mathcal{I}_t$ directly requires knowing a^*, unavailable at inference time. The coordinator's state emission provides a proxy: the self-assessed confidence $c_t \approx P(a^* = \hat{a}_t \mid \xi_{1:t})$. For

open-ended tasks where the answer space is large, increasing confidence monotonically reduces answer entropy, making confidence change a reliable proxy for information gain:

Definition 1 (Transfer Coefficient)**.** *The operational Transfer Coefficient is:*

$$T(t) = c_t - c_{t-1} \tag{5}$$

a first-order proxy for $\mathcal{I}_t$: positive when the step reduced uncertainty, zero when wasted, negative when uncertainty increased.

The simplicity of $T(t)$ is deliberate: the derivation justifies why the confidence delta is the right quantity, while h_s, h_v, and ϕ each capture a necessary but insufficient condition for progress. Empirically, $T(t)$ discriminates correct from incorrect coordination with Cohen's $d = 0.576$ ($p = 0.016$), while unsigned flux $\phi(t)$ achieves only $d = 0.44$.

The enriched form: decomposing progress. The first-order $T(t)$ has a known failure mode: confidence can rise while the hypothesis drifts (false confidence), or remain flat while evidence accumulates (productive stagnation). The enriched Transfer Coefficient decomposes progress into four channels from the state emission, where $|\cdot|$ denotes set size and subscripts denote the step:

$$T_{\text{rich}}(t) = \underbrace{(c_t - c_{t-1})}_{\text{confidence change}} + \underbrace{\beta \cdot (|E_t^+| - |E_{t-1}^+|)}_{\text{new evidence}} + \underbrace{\gamma \cdot (|M_{t-1}| - |M_t|)}_{\text{gaps resolved}} - \underbrace{\delta \cdot (|E_t^-| - |E_{t-1}^-|)}_{\text{new contradictions}} \tag{6}$$

Each term uses quantities already present in the state emission s_t: supporting evidence E_t^+, missing information M_t, and contradicting evidence E_t^-. The confidence term enters with unit weight; we set β=0.2, γ=0.3, δ=0.5. Contradictions receive the highest penalty because they represent active regression. Gap closure is weighted above evidence accumulation because resolving a known unknown is a stronger progress signal than adding one more confirming source.

T_{rich} reveals distinctions invisible to $T(t)$ alone. A step with $T(t) = 0$ but $T_{\text{rich}} > 0$ indicates productive stagnation: gaps are closing and evidence is accumulating even though confidence has not yet updated. A step with $T(t) > 0$ but $T_{\text{rich}} < 0$ indicates false confidence: certainty is rising while contradictions accumulate. Empirically, T_{rich} achieves Cohen's $d = 0.696$ ($p < 0.0001$) in discriminating correct from incorrect traces, compared to $d = 0.576$ for $T(t)$ alone (both computed as per-trace means).

From diagnosis to action. When $T(t) \approx 0$ persists, decomposing into $(\Delta H_s, \text{HDS}, \Phi_d)$ identifies which precondition for progress has failed: wrong agent type ($\Delta H_s > 0$), hypothesis drifting (HDS increasing), undirected movement ($\Phi_d \approx 0, \phi > 0$), or complete stagnation (all near zero). The framework provides three capabilities: (1) a detection trigger, where persistent $T(t) \approx 0$ signals that intervention is warranted; (2) a failure-mode classifier, where the component decomposition determines which intervention type to apply; and (3) a false-positive filter, where $T_{\text{rich}} > 0$ suppresses intervention during productive stagnation. At runtime, operationally simpler threshold conditions approximate these distinctions; T_{rich} serves primarily as an analytical tool for trace interpretation and system calibration.

Coordination regimes. Beyond per-step diagnosis, the Transfer Coefficient and flux jointly classify the agent's behavioral regime at each step. We use $T(t)$ rather than T_{rich} here because the two-dimensional space preserves a distinction the enriched scalar collapses: whether stagnation involves semantic movement (thrashing) or not (true stagnation), which determines the appropriate response. Combining $T(t)$ (progress) with $\phi(t)$ (movement) yields four modes:

$$\text{Regime}(t) = \begin{cases} \textsc{Productive} & T(t) > \epsilon,\ \phi(t) > \delta \quad (\epsilon, \delta \text{ task-calibrated}) \\ \textsc{Stalled} & |T(t)| \le \epsilon,\ \phi(t) \le \delta \\ \textsc{Thrashing} & |T(t)| \le \epsilon,\ \phi(t) > \delta \\ \textsc{Regressive} & T(t) < -\epsilon \end{cases} \tag{7}$$

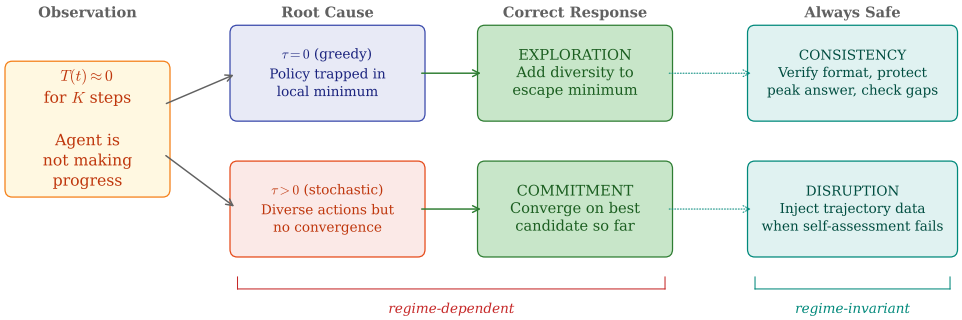

Figure 1: The same pathology ($T(t) \approx 0$) requires opposite interventions depending on the operating regime. Dashed red: applying the wrong type harms accuracy. Consistency and disruption (right) are regime-invariant.

Early predictability and its implications. Coordination failure is predictable from early deliberation. A gradient-boosted classifier trained on $T(t)$ and regime features from the first five states achieves AUC=0.844 in predicting final task outcome, exceeding the AUC=0.813 of a model using full-trace features. Early behavior predicts outcome better than the complete trajectory, implying that pathologies are structural patterns established at onset from which recovery is rare without external intervention (Dziri et al., 2023; Jin et al., 2026). This motivates real-time intervention rather than post-hoc diagnosis.

The classifier's learned feature weights reveal which early patterns matter. The strongest positive predictor is a temporary confidence drop at step 2 (coefficient $+1.26$): agents that reconsider their initial hypothesis before committing outperform those that converge immediately. The strongest negative predictor is high step-1 confidence (coefficient -0.59): agents that assert an answer before searching lock into an unverified hypothesis that subsequent evidence confirms rather than challenges. These findings align with independent observations that premature convergence in multi-turn agent systems degrades performance (Xu et al., 2025) and that LLM agents struggle to self-correct without external signal (Huang et al., 2024).

3 The Intervention Decomposition

The Transfer Coefficient diagnoses coordination pathologies, but diagnosis does not prescribe intervention. The same pathology (persistent $T(t) \approx 0$) arises from different causes at different operating temperatures and requires opposite responses. We now derive a decomposition that resolves this ambiguity.

3.1 Why the Same Pathology Requires Opposite Responses

At decoding temperature $\tau = 0$ (greedy), stagnation arises because the policy is trapped in a local minimum. The agent repeats the same action sequence deterministically and cannot escape without external perturbation. The underlying cause is insufficient exploration.

At $\tau > 0$ (stochastic), stagnation arises because temperature induces diverse actions that fail to cohere into an answer. The agent samples different strategies on each step but none converge. The underlying cause is insufficient convergence.

The naive response ("try something different") addresses only the first mechanism. Applied to the second, it worsens the problem by adding diversity to an already-unconverging system.

3.2 Formalization via Entropy-Regularized Optimization

We adopt the maximum-entropy RL framework (Haarnoja et al., 2018) as an explanatory model for the empirical regime-dependence observed in Table 1. In MaxEnt RL, the optimal policy maximizes $J(\pi) = \mathbb{E}_\pi[\sum r_t + \alpha H(\pi(\cdot|s_t))]$, where α controls the exploration-exploitation balance. Temperature τ and entropy bonus α are mechanistically different (the former rescales logits at inference, the latter shapes the training objective), but both increase behavioral diversity in the action distribution. We model their combined effect as additive: decoding temperature provides implicit diversity $\alpha_{\text{implicit}}(\tau)$, and an exploration intervention (text instructing the agent to try diverse strategies) adds explicit diversity α_{explicit}, yielding $\alpha_{\text{eff}} = \alpha_{\text{implicit}}(\tau) + \alpha_{\text{explicit}}$.

At $\tau = 0$, $\alpha_{\text{implicit}} = 0$. The system is under-diversified, and α_{explicit} moves α_{eff} toward a better balance. At $\tau > 0$, the default temperature already provides substantial diversity (evidenced by the fact that exploration interventions degrade accuracy at τ=0.2 in Table 1). Adding α_{explicit} on top pushes total diversity past the useful range, producing behavior too diffuse to converge (Ahmed et al., 2019).

This analysis yields a natural classification of interventions by their effect on policy entropy:

Definition 2 (Intervention Types)**.** Exploration *interventions increase H_π (add diversity).* Commitment *interventions decrease H_π (promote convergence).* Consistency *interventions verify state-action alignment without shifting H_π.* Disruption *interventions provide trajectory-level data, enabling the agent to adjust H_π based on observed dynamics rather than a fixed directive.*

> **Regime Safety Principle.** Exploration and commitment are regime-dependent: each improves coordination in one temperature regime and degrades it in the other. Consistency and disruption are regime-invariant. (Validated empirically in Table 1.)

Consistency is regime-invariant because it is H_π-neutral: verifying answer format neither adds nor removes exploration. Disruption is regime-invariant because it is informational rather than directive: supplying trajectory data ("your confidence has been flat for 5 states") without prescribing the response allows the agent to choose the appropriate adjustment. The agent possesses task-specific context that the intervention designer lacks, and disruption leverages this asymmetry.

A complementary interpretation comes from sequential information acquisition. At τ=0, repeated searches return the same results, so additional queries accumulate information without noise. At τ>0, repeated searches return variable results, so later observations may corrupt rather than refine the answer. The commitment intervention ("protect your peak-confidence answer") follows directly: it prevents noisy late searches from displacing a reliable early observation.

Validation. Table 1 validates the Regime Safety Principle on 97 GAIA tasks at both temperatures. As the principle predicts, exploration improves accuracy in the deterministic regime and degrades it in the stochastic regime, while commitment shows the opposite pattern. The Combined row (commitment + disruption) performs worse than commitment alone at τ=0 because the disruption monitor's stagnation threshold, calibrated for the stochastic regime, fires excessively under deterministic execution where confidence is naturally less variable. This interference motivates the gated architecture in §4: the monitor activates only after the static component has had time to act.

The decomposition also explains an otherwise puzzling empirical result. A coordinator prompt mixing exploration guidance ("try a fundamentally different strategy when stuck") with consistency checks ("verify format, check for drift") achieves 80% at τ=0 but degrades accuracy by 3.1% at τ=0.2. The Regime Safety Principle explains the reversal: the exploration component helps at τ=0 but over-regularizes at τ>0, while the consistency component (regime-invariant) cannot compensate. Separating the types and replacing exploration with commitment yields the constraint effective in the stochastic regime.

Table 1: Regime-dependence validation on 97 GAIA tasks. Each cell shows accuracy change when the given intervention type is applied to the coordinator, relative to the same system with no intervention (75.8% at τ=0.2; 78.5% at τ=0).

Type	$\tau = 0$	$\tau = 0.2$	Prediction
Exploration	+1.1%	−3.1%	✓
Commitment	−1.1%	+1.5%	✓
Combined†	−4.3%	−0.5%	✓†

†Combined includes commitment (harmful at τ=0) plus disruption with uncalibrated threshold.

4 Architecture: Static and Dynamic Control

For stochastic agents, the Regime Safety Principle prescribes commitment, consistency, and disruption interventions. The delivery mechanism depends on whether the diagnostic signal is accessible to the agent from its local state. Some failures are *pattern-visible*: a flat confidence sequence is recognizable as stagnation to the agent itself, provided it is instructed to look. Other failures are *pattern-invisible*: an agent repeating the same retrieval strategy for nine states cannot perceive the redundancy from any single state. TRAC implements both through a static component (prompt-based self-assessment) and a dynamic component (runtime trace monitoring), connected by a gating policy that prevents interference.

Static constraints. Four instructions in the coordinator prompt address pattern-visible failures: (1) protect peak-confidence answers from regression; (2) verify critical gaps before synthesis; (3) match answer format; (4) observe the confidence trajectory and commit if flat, continue if rising. The agent evaluates these against its own STATE emissions at each step.

Dynamic monitor. For pattern-invisible failures, a monitor observes the coordinator's state emissions and can inject messages before each LLM call. It evaluates three conditions: stagnation (confidence range $<$0.08 over 5 states), post-peak regression (drop $>$0.15 from maximum), and persistent gaps ($\geq$8 states unresolved). On detection, it injects the agent's actual trajectory statistics, constituting disruption without prescribing a response. The mechanism adds zero LLM cost, fires at most three times per task, and aborts at 30 total LLM calls to bound computation on capability-limited tasks.

Gated activation. Deploying both simultaneously degrades accuracy (75.3%) below static alone (77.3%), because the monitor fires while the static constraint is still resolving the pathology. TRAC prevents this by withholding monitor injection during a self-assessment window of K=5 states. If stagnation persists beyond this window, the static constraint has demonstrably failed and the monitor activates. When static succeeds, the monitor never fires.

Diagnostic-intervention coupling. The static constraints act on confidence patterns the monitor also reads, creating a potential feedback loop. We verify no corruption occurs: confidence distributions shift modestly under intervention (mean 0.47 vs 0.41, KS p=0.0002) in a direction consistent with earlier commitment, with no clustering at threshold boundaries that would indicate strategic reporting.

5 Experiments and Analysis

5.1 Setup

We evaluate on 97 GAIA tasks (Mialon et al., 2023) using Claude Opus with four sub-agents (search, code, reader, synthesizer). Only the coordinator's prompt and callbacks vary; all other infrastructure is shared. We compare: **Vanilla** (no guidance), **Static** (the four prompt-based constraints from §4 that instruct the agent to observe its own confidence trend), **Combined** (static + dynamic monitor, no gating), and **TRAC** (gated deployment per §4), each at τ=0.2 with two independent seeds. For regime validation, we additionally test

Exploration at both temperatures. We assess significance using McNemar's exact test on discordant task pairs, pooled across both seeds.

5.2 Main Result

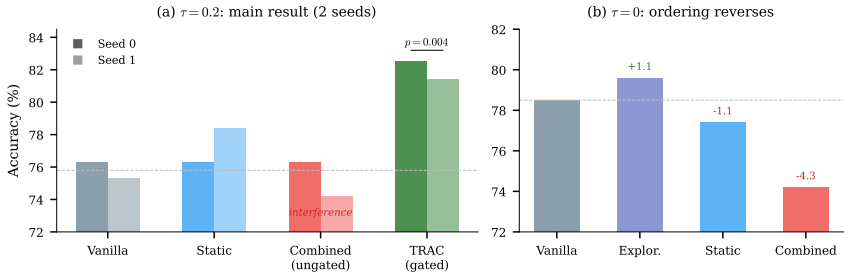

Figure 2: (a) τ=0.2: neither Static nor Combined (ungated) reliably outperforms the baseline; TRAC achieves +6.2% on both seeds (p=0.004). (b) τ=0: ordering reverses as predicted.

Table 2: 97 GAIA tasks, $\tau = 0.2$. Combined deploys both components simultaneously without gating. TRAC gates the monitor behind the static self-assessment window.

	Seed 0	Seed 1	Avg	Δ
Vanilla	76.3%	75.3%	75.8%	–
Static	76.3%	78.4%	77.3%	+1.5
Combined (ungated)	76.3%	74.2%	75.3%	−0.5
TRAC (gated)	**82.5%**	**81.4%**	**82.0%**	**+6.2**

Table 2 presents the primary result. TRAC achieves +6.2% accuracy on both seeds (pooled McNemar: 14 wins, 2 losses, $p = 0.004$) while using 2.8% fewer mean deliberation steps per task, a Pareto improvement in both accuracy and compute. Static alone shows modest, inconsistent gains across seeds. Combined (ungated) performs below Static: the monitor over-fires on tasks where the static constraint is already resolving the issue. TRAC's gating eliminates this interference by withholding the monitor until the self-assessment window elapses without resolution. On seed 0, the static component uniquely solves 6 tasks and the dynamic component uniquely solves 6 different tasks, with 68 solved by both and 17 by neither (tasks requiring capabilities outside the system, such as video analysis or paywalled data).

At τ=0 (97 tasks), the ordering reverses as predicted: exploration +1.1%, commitment −1.1%, combined −4.3%. This confirms that TRAC's use of commitment and disruption is appropriate for the stochastic regime, since both degrade accuracy under deterministic execution.

5.3 Mechanism

Trace analysis (Appendix D) reveals how TRAC converts wrong answers to correct ones. In all tasks uniquely solved by the dynamic component, the monitor injection immediately precedes the state where the agent changes strategy: $T(t) \approx 0$ before injection, $T(t) > 0$ afterward.

Two mechanisms account for the improvements. In the first (strategy pivot), an agent trapped in a repetitive search loop for nine states receives its trajectory statistics and pivots to a qualitatively different approach. Although the agent is instructed to monitor its confidence trend (the static constraint), this case involved confidence fluctuating between 0.30 and 0.40 without a clear flat pattern, making it ambiguous whether stagnation or slow progress was occurring. The monitor resolved the ambiguity by computing the aggregate statistic (range $<$0.08 over 5 states) that the agent's self-assessment could not reliably classify. In the second (search diversification), an agent repeatedly attempting a blocked resource receives

stagnation feedback and tries alternative retrieval strategies (web archives, academic papers) that were always available but never attempted. The monitor's literal advice ("commit") was inapplicable since the agent had no answer, yet the interruption of the repetitive loop enabled the agent to select a new approach. In the uncontrolled trace, 72% of states repeated the same blocked retrieval strategy; with the monitor, only 17% did.

In both cases, the monitor's value lies not in the content of its message but in its function as a perturbation that breaks fixed points of the coordination dynamics. The static component cannot achieve this because the confidence pattern in these cases is ambiguous enough that the agent's qualitative self-assessment ("am I flat or slowly progressing?") fails to trigger commitment.

T_{rich} diagnostic. Post-hoc T_{rich} analysis confirms the gating mechanism's decisions: states where the monitor correctly withholds intervention show $T(t) = 0$ but $T_{\text{rich}} > 0$ (productive stagnation with gap closure), while states where the monitor correctly fires show both $T(t) = 0$ and $T_{\text{rich}} = 0$ (genuine stagnation with no progress on any channel). After intervention, T_{rich} rises, confirming that the disruption converted stagnation to multi-channel progress. Full per-step traces are reported in Appendix A.

5.4 Failure Taxonomy and Efficiency

Seventeen tasks remain incorrect under all conditions, representing a ceiling that no coordination intervention can breach. Manual inspection reveals that 10 require capabilities the system lacks (video analysis, JavaScript rendering, paywalled data), 4 retrieve the correct source but misinterpret the evidence, and 3 produce the right value in the wrong format. These tasks share a distinctive trace signature: $T(t) \approx 0$ combined with high routing diversity H_s, indicating the system exhausted all available agent types without making progress. This signature allows the practitioner to distinguish coordination failures (where better routing would help) from capability gaps (where new tools are needed) without access to the gold answer.

TRAC averages 45.0 steps per task versus 46.3 for vanilla and 50.1 for Static, with budget abort compressing the worst case from 474 steps to 257 (full distribution in Appendix E, Figure 5). TRAC is the only condition achieving both higher accuracy and lower total computation than the baseline.

Taken together, the results validate the central theoretical claim from §2: good coordination is entropy *resolution* (decreasing answer uncertainty) rather than entropy minimization, and the correct path to resolution depends on the operating regime.

6 Conclusion

We have shown that coordination interventions for multi-agent LLM systems decompose into regime-dependent types (exploration, commitment) whose effect reverses with decoding temperature, and regime-invariant types (consistency, disruption) that remain safe across operating conditions. The decomposition follows from the interaction between a policy's built-in stochasticity and externally-imposed diversity: adding explicit entropy to an already-stochastic policy over-regularizes it, while adding convergence pressure to a deterministic policy prevents necessary exploration.

TRAC operationalizes this decomposition through two complementary mechanisms (static self-trend constraints and a dynamic trace monitor) connected by a gating policy that prevents interference between them. On the GAIA benchmark, TRAC achieves +6.2% accuracy (p=0.004) while using fewer total deliberation steps than the uncontrolled baseline, demonstrating that principled coordination control can improve both accuracy and efficiency simultaneously.

The decomposition is architecture-agnostic. Any multi-agent system with observable per-step state and adjustable policy entropy admits the same four-type classification. Specific detection thresholds require calibration per system and temperature, but the regime-safety

properties derive from the structure of entropy-regularized optimization rather than from architectural details.

The broader methodological implication is that intervention design for multi-agent systems must account for the operating regime. Evaluating coordination interventions at a single decoding temperature, as is common practice, risks missing interaction effects that reverse an intervention's sign.

References

Zafarali Ahmed, Nicolas Le Roux, Mohammad Norouzi, and Dale Schuurmans. Understanding the impact of entropy on policy optimization. *International Conference on Machine Learning*, 2019.

Mert Cemri, Melissa Z. Pan, Shuyi Yang, Lakshya Agrawal, et al. Why do multi-agent LLM systems fail? In *NeurIPS Track on Datasets and Benchmarks*, 2025.

Junlin Chi et al. AgentMonitor: A plug-and-play framework for predictive and secure multi-agent systems. *arXiv preprint*, 2024.

Nouha Dziri, Ximing Lu, Melanie Sclar, Xiang Lorraine Li, Liwei Jian, Bill Yuchen Lin, Sean Welleck, Peter West, Chandra Bhagavatula, Ronan Le Bras, et al. Faith and fate: Limits of transformers on compositionality. In *NeurIPS*, 2023.

Zhibin Gou, Zhihong Shao, Yeyun Gong, Yelong Shen, Yujiu Yang, Nan Duan, and Weizhu Chen. CRITIC: Large language models can self-correct with tool-interactive critiquing. *International Conference on Learning Representations*, 2024.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *ICML*, 2018.

Jie Huang, Chen Xia, et al. Large language models cannot self-correct reasoning yet. In *ICLR*, 2024.

Wenlong Huang, Fei Xia, Ted Xiao, Harris Chan, Jacky Liang, Pete Florence, Andy Zeng, Jonathan Tompson, Igor Mordatch, Yevgen Chebotar, et al. Inner monologue: Embodied reasoning through planning with language models. *Conference on Robot Learning*, 2023.

Hongyi Henry Jin, Wenhan Yang, Meysam Ghaffari, Carlos Morato, and Baharan Mirzasoleiman. Reasoning quality emerges early: Data curation for reasoning models. In *International Conference on Machine Learning*, 2026.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In *ICLR*, 2023.

Yuyang Li, Yiran Dou, Jie-Jing Shao, Yueming Lyu, Ivor Tsang, and Haiyan Yin. SkillTracer: Structural failure attribution and refinement of agentic skills in long-horizon web tasks. In *ICLR Workshop on MALGAI*, 2026.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *ICLR*, 2024.

Grégoire Mialon, Clémentine Fourrier, Craig Swift, Thomas Wolf, Yann LeCun, and Thomas Scialom. GAIA: A benchmark for general AI assistants. *arXiv preprint arXiv:2311.12983*, 2023.

Thomas Schreiber. Measuring information transfer. *Physical Review Letters*, 85(2):461–464, 2000.

Amrith Setlur, Chirag Nagpal, Adam Fisch, Xinyang Guo, Sergey Levine, Alekh Agarwal, Shreyas Shankar, and Aviral Kumar. Rewarding progress: Scaling automated process verifiers for LLM reasoning. In *ICLR*, 2025.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *NeurIPS*, 2023.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling LLM test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*, 2024.

Wujiang Xu, Wentian Zhao, Zhenting Wang, Yu-Jhe Li, Can Jin, Mingyu Jin, Kai Mei, Kun Wan, and Dimitris N. Metaxas. EPO: Entropy-regularized policy optimization for LLM agents reinforcement learning. *arXiv preprint arXiv:2509.22576*, 2025.

Andy Zhou, Kai Yan, Michal Shlapentokh-Rothman, Haohan Wang, and Yu-Xiong Wang. Language agent tree search unifies reasoning acting and planning in language models. *International Conference on Machine Learning*, 2024.

Junze Zhu, Weihao Chen, Xuanwang Zhang, Zhen Wu, and Xinyu Dai. Recognize your orchestrator: An entropy dynamics perspective for LLM multi-agent systems. In *ICML*, 2026.

A T_{rich} Diagnostic Summary

Table 3 reports T_{rich} diagnostics across the 15 tasks with highest total stagnation. Wrong tasks average 8.0 genuine stagnation steps per task versus 1.6 for correct tasks, confirming that persistent $T_{\text{rich}} \approx 0$ reliably identifies coordination failure. Productive stagnation (flat confidence but gaps closing, $T_{\text{rich}} > 0$) occurs in both correct and wrong tasks, validating that T_{rich} distinguishes cases where the monitor should remain silent from cases where intervention is warranted.

Table 3: T_{rich} diagnostic for the 15 highest-stagnation tasks. Genuine stagnation (T=0, T_{rich}=0) indicates monitor should fire. Productive stagnation (T=0, T_{rich}>0) indicates progress beneath flat confidence.

Task	Result	States	Genuine	Productive	mean T_{rich}
5f982798	✓	49	33	5	+0.064
0b260a57	×	40	22	4	+0.068
624cbf11	×	36	20	3	+0.049
20194330	×	28	20	0	+0.007
23dd907f	×	19	15	1	+0.028
ebbc1f13	×	18	15	1	+0.024
8131e2c0	×	19	14	0	+0.031
384d0dd8	×	16	9	3	+0.083
8b3379c0	×	13	11	0	+0.008
a0c07678	×	17	10	0	+0.045
65638e28	✓	11	8	0	+0.105
05407167	✓	12	6	1	+0.109
0e9e85b8	×	19	5	2	+0.161
7a4a336d	×	10	5	1	+0.180
d1af70ea	✓	13	4	2	+0.246

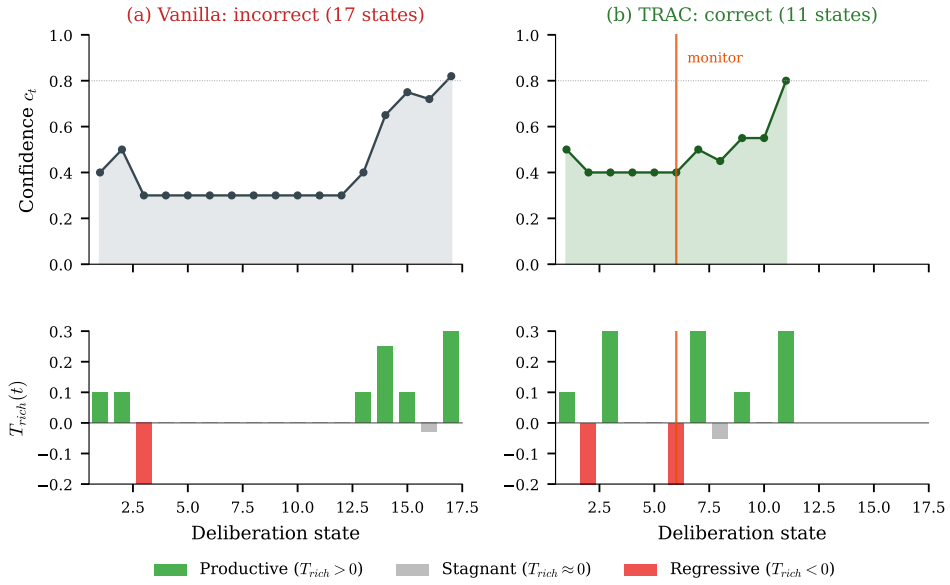

Figure 3: T_{rich} diagnostic on the Yoshida task. Filled area shows confidence trajectory; colored bars encode per-step T_{rich} (green: productive, gray: stagnant, red: regressive). (a) Vanilla: 9 consecutive gray bars confirm genuine stagnation. (b) TRAC: monitor fires at state 6, converting gray bars to green as the agent pivots strategy.

B Regime-Dependence Validation ($\tau = 0$)

Table 4: 97 tasks at $\tau = 0$. Ordering reverses as predicted.

	Accuracy	Δ
Vanilla	78.5%	–
Exploration	79.6%	+1.1
Static (commitment)	77.4%	−1.1
Combined (commitment + monitor)	74.2%	−4.3

C Regime Reversal Examples

Table 5: Individual tasks demonstrating regime reversal. Exploration correct at τ=0, wrong at τ=0.2; vanilla solves all at both temperatures.

Task	Expl. τ=0	Expl. τ=0.2	Failure mechanism
Survivor	✓	× (171 steps)	Correct parametric answer displaced by noisy verification
Footnote 397	✓	× (339 steps)	Direct access strategy abandoned for broad search
Tri-Rail	✓	× (143 steps)	Correct schedule reading overwritten by re-search

Vanilla solves all three in 9, 47, and 111 steps respectively at τ=0.2.

D Case Studies

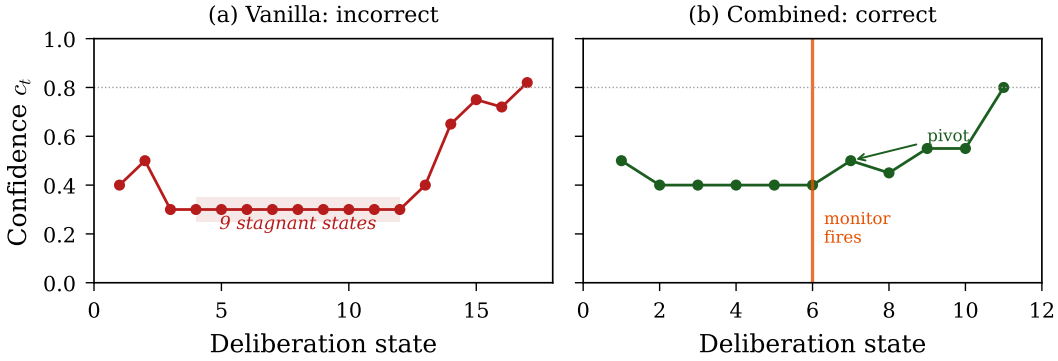

Figure 4: Confidence trajectory for the Yoshida task. (a) Vanilla: confidence plateaus at 0.30 for nine consecutive states while the agent repeats the same search strategy. (b) TRAC: the monitor fires at state 6 with concrete trajectory data, triggering a strategy pivot that resolves the task in fewer total steps.

D.1 Case Study 1: Strategy Pivot

Task. Identify the pitchers with the jersey numbers adjacent to Taishō Tamai's on the Nippon-Ham Fighters as of July 2023. Gold answer: Yoshida, Uehara. (Throughout: "steps" = total LLM calls across all agents; "states" = coordinator decision points with STATE emissions.)

Vanilla (incorrect, 105 steps, 17 states). The agent identifies Tamai as #19 and begins searching for the #18 holder. Confidence remains at 0.30 for nine consecutive states (4 through 12), each with identical hypothesis text and minor query reformulations. The agent eventually guesses "Uwasawa" without sufficient evidence.

TRAC (correct, 69 steps, 11 states). States 1 through 5 show confidence $\{0.50, 0.40, 0.40, 0.40, 0.40\}$. At state 6, the monitor injects: "Your confidence has been [0.40, 0.40, 0.40, 0.40, 0.40]. You are not making progress." At state 7, the agent responds by searching the team's Wikipedia roster page, a strategy never attempted in the vanilla trace. Confidence rises to 0.80 by state 11 with the correct answer confirmed.

Why TRAC succeeds. The confidence pattern (fluctuating between 0.30 and 0.40) was ambiguous to the agent's qualitative self-assessment but unambiguous to the monitor's numerical threshold (range < 0.08). The monitor's intervention did not prescribe a new strategy; it provided the trajectory data that enabled the agent to recognize its own stagnation and choose an alternative approach.

D.2 Case Study 2: Search Diversification

Task. Trace a 5-hop chain: Yola word "gimlie" $\rightarrow$ Latin root "caminata" $\rightarrow$ Spanish "caminata" $\rightarrow$ Collins dictionary 1994 example sentence $\rightarrow$ source title $\rightarrow$ Google translation. Gold answer: The World of the Twenty First Century.

Vanilla (incorrect, 114 steps, 18 states). The agent identifies the Collins dictionary page but receives HTTP 403. It spends 13 of 18 states (72%) attempting the same access pattern with URL variations, never trying alternative information sources.

Static only (incorrect, 146 steps, 23 states). The same failure occurs: 16 of 23 states (70%) repeat blocked access attempts. The static constraint detects the flat confidence pattern and advises commitment, but the agent has no answer to commit to.

TRAC (correct, 187 steps, 30 states). The monitor fires at states 5, 6, and 8 with stagnation feedback. The agent diversifies its retrieval strategy to include Wayback Machine archives, academic papers citing the dictionary entry, and PDF parsing. It finds the source title through an archived page at state 30.

Why TRAC succeeds. The static constraint correctly diagnosed the flat pattern but prescribed the wrong action (commitment without an answer). The monitor's disruption signal broke the repetitive loop, enabling the agent to try retrieval strategies that were always available but never considered. The proportion of redundant states dropped from 72% to 17%.

E Task-Level Outcomes and Efficiency

Table 6 lists all tasks whose outcome changed between the vanilla baseline and TRAC on seed 0. The "Source" column indicates whether the improvement is attributable to the static component, the dynamic monitor, or both (the task is solved by either ablation independently). The single regression (Diamond) occurred with zero monitor interventions, indicating stochastic variation rather than intervention-caused harm.

Table 6: Task outcome changes: TRAC vs. uncontrolled baseline (seed 0, $\tau = 0.2$).

Task	Source	Mechanism
Improvements (7):		
Yoshida/Uehara	Both	Stagnation detection $\rightarrow$ strategy pivot
Yola/gimlie	Dynamic	Stagnation $\rightarrow$ search diversification
King of Pop	Dynamic	Persistence through low-confidence states
South Station	Static	Commitment at appropriate confidence
Ice cream headstone	Static	Gap-checking before synthesis
Lego images	Both	Improved synthesis formatting
Info availability	Both	Format constraint enforcement
Regressions (1):		
Diamond	Neither	Stochastic (0 interventions fired)

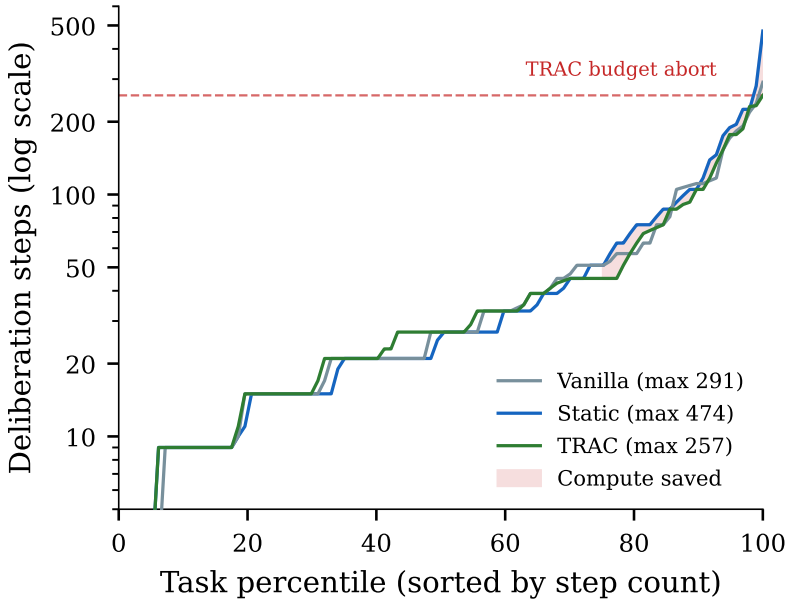

Figure 5: Step distribution across conditions. TRAC's budget abort compresses the right tail, capping computation on capability-limited tasks. The shaded region shows compute saved relative to Static.

F Implementation

Static constraint text (verbatim). The following text is appended to the coordinator's system prompt:

> Before synthesizing your final answer: (1) PROTECT YOUR BEST ANSWER: if confidence previously exceeded 0.7 and has dropped without new contradicting evidence, revert. (2) CHECK GAPS: if missing information is essential, make one targeted attempt; otherwise commit. (3) MATCH FORMAT: full numeric values, exact decimal places, exact units. (4) OBSERVE PROGRESS: if confidence pattern is flat (0.3, 0.3, 0.3, 0.35), commit; if rising (0.2, 0.3, 0.4, 0.5), continue.

Dynamic monitor. The monitor attaches to the coordinator via the agent framework's callback mechanism. An `after_model_callback` parses structured STATE emissions from each coordinator response. A `before_model_callback` evaluates three pathology conditions before each coordinator LLM call:

- Stagnation: confidence range <0.08 over the last 5 emitted states.
- Post-peak regression: confidence has dropped >0.15 below the trace maximum.
- Gap persistence: ≥ 8 states with unresolved missing information items.

On detection, the monitor injects a user-role message containing the agent's trajectory statistics. Maximum 3 interventions per task; budget abort at 30 total LLM calls across all agents.