DEGREES OF FREEDOM IN LLM MULTI-AGENT COORDINATION: BETWEEN CENTRALIZED AND DECENTRALIZED REGIMES

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language model (LLM)-based multi-agent systems (MAS) are commonly organized around two coordination regimes. Centralized coordination provides explicit planning and control but can create a bottleneck at a single coordination locus, whereas decentralized coordination offers local autonomy but must achieve effective collaboration through distributed decisions and experience. These complementary trade-offs motivate a systematic study of the space between the two regimes. We introduce *Degrees of Freedom* (DoF), a continuous configuration framework that quantifies the size of the decision space delegated to downstream agents. DoF jointly controls the granularity of delegated work and, within each granularity regime, the target probability that each task unit is assigned to the decentralized branch. Its endpoints recover the two pure coordination regimes. Across multiple task domains, DoF reveals a hierarchy of organizational effects. Delegated granularity and within-regime routing produce structured but bounded profile reallocations, whereas task domain and retained experience are associated with stronger differences between the centralized- and decentralized-branch profiles. The resulting performance landscape is task-dependent: different domains favor different DoF configurations, while atomic-target conditions are generally more consistent across runs. In-domain retained experience also brings the two branch profiles closer together. The same controlled sweep reveals a task-dependent coordination landscape: some domains exhibit clearly higher-performing configurations, whereas others show flatter responses, and no single configuration dominates across domains. In five of the six evaluated state–dataset settings, an intermediate DoF configuration achieves higher mean accuracy than both pure endpoints, demonstrating the value of exploring the coordination space between them. Together, these findings provide empirical guidance for studying and designing intermediate coordination regimes in LLM-based MAS.

1 INTRODUCTION

Large language model (LLM)-based multi-agent systems (MAS) extend agentic problem solving by decomposing complex requests into manageable units, assigning them to agents with different capabilities, and coordinating their interactions during execution (Guo et al., 2024). From the perspective of task allocation and decision authority, automatic MAS can be organized around two coordination regimes. Centralized coordination retains planning, assignment, review, and aggregation in an explicit controller, whereas decentralized coordination distributes execution and local interaction decisions among active agents. These regimes provide two distinct ways to organize the same underlying problem-solving process.

Centralized coordination benefits from a global view of the task and explicit control over agent selection and result integration, but concentrating these decisions at one locus can create planning, communication, and integration bottlenecks. Decentralized coordination supports local autonomy and distributed adaptation, but effective collaboration must emerge through successive local execution, splitting, and inter-agent transfer decisions (Cemri et al., 2025; Mao et al., 2026; Yang et al., 2025). The resulting organization is therefore more dependent on interaction paths and accumulated agent state. The two regimes expose complementary advantages and structural costs rather than a

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

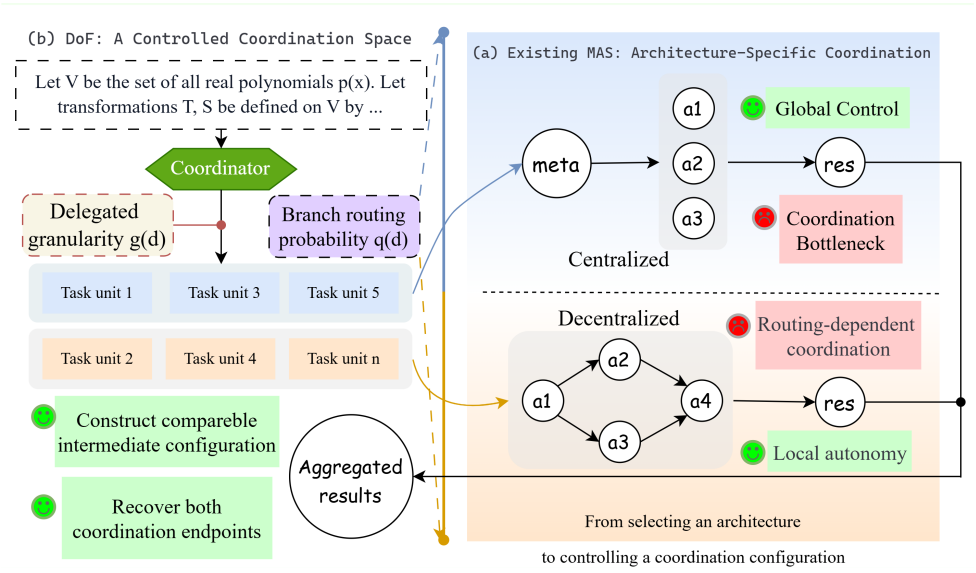

Figure 1: Illustration of the DoF framework. (a) Existing automatic MAS instantiate centralized or decentralized coordination through architecture-specific mechanisms, leaving intermediate configurations difficult to compare under matched conditions. (b) DoF moves from architecture selection to controlled coordination configuration by jointly specifying delegated granularity and per-unit branch-routing probability.

universally preferable organization. Existing systems typically instantiate one complete architecture or optimize selected coordination mechanisms, leaving the space between centralized and decentralized coordination underexplored under matched experimental conditions.

We address this gap with the Degrees-of-Freedom (DoF) framework, illustrated in Figure 1, which interprets centralized and decentralized coordination as the two endpoints of a space of delegated decision authority. The configured degree of freedom $d \in [0, 1]$ therefore indicates how much decision space is assigned to downstream agents: at $d = 0$, the undecomposed request remains under centralized coordination, leaving no branch-level coordination autonomy to downstream agents, whereas at $d = 1$, the original request is handled by decentralized coordination, exposing the full branch-internal decision space to the downstream agents. For $0 < d < 1$, the DoF Coordinator decomposes the request and controls downstream autonomy along two dimensions. First, delegated granularity determines the semantic scope of each unit: atomic-target units are narrower than subtask-target units and therefore give each downstream execution a smaller decision scope. Second, within either granularity regime, increasing the target probability of decentralized assignment gives more task units to the decentralized branch and enlarges the fraction of coordination decisions made downstream. Concretely, the atomic-target regime uses $0 < d \leq 0.5$ with $q(d) = 2d$, whereas the subtask-target regime uses $0.5 < d < 1$ with $q(d) = 2d - 1$; thus, $q(d)$ increases with d within each regime, while the granularity transition changes the scope of the delegated units. DoF consequently provides a single configuration index for allocating decision authority between the two endpoints without collapsing their distinct mechanisms into a third coordination regime.

We instantiate the centralized and decentralized branches with MetaAgent and AgentNet(Yang et al., 2025), respectively, over the same five-agent roster and Qwen3-Next-80B (Qwen Team, 2025). We evaluate fifteen DoF configurations through repeated sweeps across five reasoning domains. This controlled design compares endpoint and intermediate configurations while keeping the underlying model, agent roster, and branch implementations fixed. The controlled sweeps reveal a progression from organizational adjustments to task outcomes. *First*, delegated granularity and within-regime routing produce structured but bounded changes in worker-selection and node-activation profiles. *Second*, task domain and retained experience are associated with stronger branch-specific differences: domain and execution branch establish distinct profiles, while in-domain retained experience brings the centralized- and decentralized-branch profiles closer together. The derived cooperation paths

provide a descriptive view of the coordination events underlying these profiles. *Finally*, the resulting performance landscape is task-dependent: intermediate configurations attain the highest observed mean in several settings, endpoint or near-endpoint configurations remain preferable in others, and atomic-target conditions are generally more consistent across runs. Together, these observations characterize coordination as a task- and state-conditioned design choice rather than a fixed preference for either endpoint. Our contributions are:

- **A controlled formulation of the coordination space.** We formulate centralized and decentralized coordination as endpoint regimes and identify delegated decision scope and processing locus as joint controls required to study the configurations between them.
- **A piecewise DoF coordination framework.** We introduce DoF as a common configuration index that combines two delegated-granularity regimes with per-unit branch-routing probability under a shared model, agent roster, and pair of branch implementations.
- **An empirical characterization of task-conditioned coordination.** Repeated DoF sweeps show that delegated granularity and within-regime routing produce structured but bounded changes in branch-specific profiles, while task domain and execution branch are associated with stronger profile differences and in-domain retained experience with closer cross-branch profiles. These organizational patterns accompany task-dependent accuracy peaks and greater run-to-run consistency for atomic-target conditions.

2 Related Work

2.1 Automatic Multi-Agent Systems

Centralized coordination regimes in automatic MAS place system construction and execution under an explicit controller or structured workflow. Existing frameworks program agent conversations, assign roles, decompose requests, select models or tools, and aggregate intermediate results (Wu et al., 2024; Li et al., 2023; Hong et al., 2024; Shen et al., 2023; Chen et al., 2024). Hierarchical approaches further combine high-level planning with specialized or dynamically constructed agents (Zhang et al., 2025c; Hou et al., 2025; Wang et al., 2024a). More recent systems represent agents, dependencies, control flow, and message passing as executable graphs, or synthesize domain-specific nodes separately from topology orchestration (Liu et al., 2026; Lin et al., 2026a). These designs enable explicit control over complex organizations, but retain coordination authority in a controller, hierarchy, or generated workflow.

Decentralized coordination regimes in automatic MAS instead distribute execution and interaction decisions among active agents. Dynamic interaction networks allow agents to select local actions and communication partners, so task allocation emerges from the evolving interaction graph rather than a single global plan (Liu et al., 2024; Yang et al., 2025). Shared-context approaches further distribute progress tracking, task claiming, and result integration across parallel agents (Mao et al., 2026). Such systems reduce reliance on a single controller and support local adaptation, but their realized organization depends on sequences of local decisions, peer context, and retained agent state.

The two regimes therefore expose complementary limitations. Centralized systems can concentrate planning and integration at one coordination locus, whereas decentralized systems distribute control but make global organization dependent on local interaction and accumulated state. Selecting either regime alone does not reveal how much decision space should be delegated for a particular task. In this work, we introduce DoF as a unified coordination framework that represents centralized and decentralized coordination regimes as endpoints of a common configuration space and models intermediate systems through delegated granularity and per-unit branch routing. Our experiments show that intermediate configurations attain the highest observed accuracy in several task settings, while endpoint or near-endpoint configurations remain preferable in others, indicating that the appropriate coordination regime is task dependent.

2.2 Empirical Analysis of Automatic Multi-Agent Systems

Prior work analyzes automatic MAS through complementary diagnostic and intervention-based views. A useful organizing perspective treats MAS organization as a sequence of coupled decisions: what work is delegated, where it is routed, who performs it, how the resulting interaction unfolds, and

how the organization adapts with experience. Task-decomposition methods determine the scope and granularity of delegated work, from recursively expanded sub-problems to structured reasoning units (Prasad et al., 2024; Yao et al., 2023), while compositional-routing work shows that decomposition granularity can condition downstream skill retrieval (Gao, 2026). Routing and workflow-construction methods then determine which agents, communication paths, or execution procedures receive each unit (Liu et al., 2024; Yue et al., 2025). This process is coupled with role assignment and agent specialization, which shape the capabilities and division of labor associated with each processing node (Hong et al., 2024; Liu et al., 2024; Yang et al., 2025). Trajectory-level and behavioral studies further examine how roles are activated, how information propagates, and how interaction paths shape realized organization (Cemri et al., 2025; Li et al., 2024; Du et al., 2024; Smit et al., 2024; Jwalapuram et al., 2026). Experience-based methods show that stored reflections, successful trajectories, agent-local memories, shared traces, selectively retained execution states, and transferable orchestration skills can modify later routing and interaction behavior (Shinn et al., 2023; Zhao et al., 2024; Park et al., 2023; Gao & Zhang, 2024; Yang et al., 2025; Rajib et al., 2026; Lin et al., 2026b). Collectively, these analyses show that MAS behavior depends on topology, communication, decomposition, task assignment, branch-internal routing, model binding, and retained experience.

A complementary set of studies treats these organizational dimensions as intervention points, varying communication availability, local leadership, topology, collaboration protocol, orchestration structure, adaptive routing, delegation, or skill retrieval, and measuring changes in performance, cost, robustness, or consistency (Santos et al., 2023; Sugawara et al., 2024; Zhuge et al., 2024; Zhang et al., 2025b;a; Yu, 2026; Chen et al., 2026; Sander et al., 2026; Guan et al., 2026; Cui et al., 2026). Existing work therefore provides important evidence about failures and adaptations, but typically studies one mechanism within a particular architecture or design family rather than a matched configuration space connecting centralized and decentralized coordination regimes. DoF advances this line of analysis by keeping the model, agent roster, and branch implementations fixed while sweeping fifteen endpoint and intermediate configurations. It uses branch-normalized allocation profiles, derived cooperation paths, repeated-run accuracy, and retained-state comparisons to characterize how organizational behavior and task outcomes vary across this controlled space, rather than introducing another endpoint-specific failure taxonomy.

3 Degrees of Freedom Framework

3.1 Problem Formulation and Overview

Given an input task x and a configured degree of freedom $d \in [0, 1]$, the framework produces a final output $\hat{y}$. The endpoints retain the original request: $d = 0$ sends x directly to centralized coordination, whereas $d = 1$ sends x directly to decentralized coordination. For an intermediate configuration $0 < d < 1$, DoF specifies both the target granularity

$$g(d) = \begin{cases} \text{atomic}, & 0 < d \leq 0.5, \\ \text{subtask}, & 0.5 < d < 1, \end{cases} \tag{1}$$

and the target probability that a task unit is assigned to the decentralized branch

$$q(d) = \begin{cases} 2d, & 0 < d \leq 0.5, \\ 2d - 1, & 0.5 < d < 1. \end{cases} \tag{2}$$

Here, $g(d)$ determines the semantic scope of the delegated units, and $q(d)$ is the target probability that an eligible task unit is assigned to the decentralized branch. Thus, d is a continuous configuration index. Granularity changes discretely between two granularity regimes, while the branch-assignment probability varies continuously within each regime. Appendix E compares these target probabilities with the realized decentralized task-unit fractions.

For an intermediate configuration, the Coordinator decomposes the input into a dependency graph

$$G_d(x) = (U_d, E_d) = \mathcal{D}\big(x; g(d)\big), \tag{3}$$

where $\mathcal{D}$ is the granularity-conditioned decomposition operation, U_d is the set of routable task units, and $E_d \subseteq U_d \times U_d$ is the set of directed dependency edges. Once the predecessors of a unit $u \in U_d$

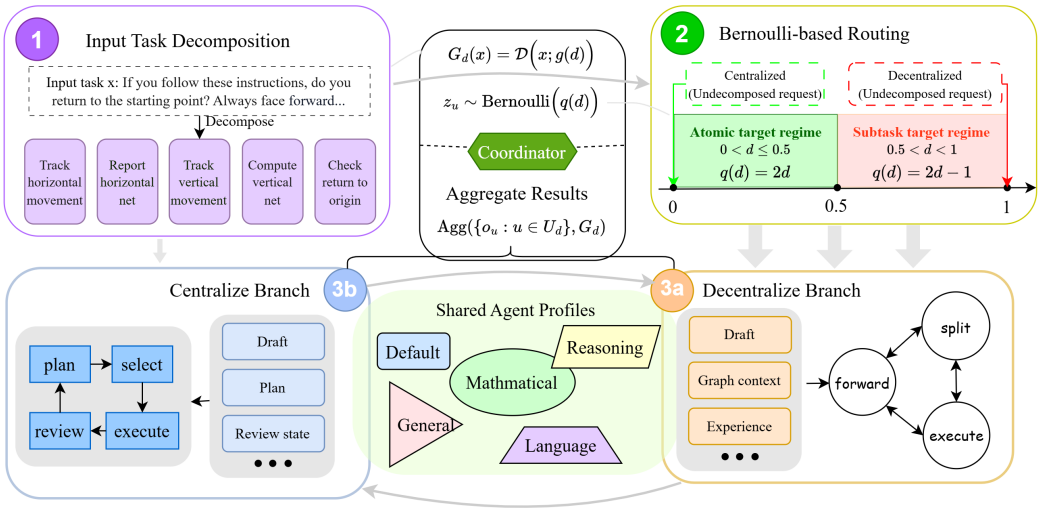

Figure 2: Overview of the DoF workflow. The Coordinator decomposes the input at the configured granularity, assigns each task unit to the centralized or decentralized branch through Bernoulli-based routing, and aggregates the completed unit outputs, linking the piecewise DoF schedule to the two branch implementations.

have completed, the Coordinator samples

$$z_u \sim \text{Bernoulli}(q(d)), \qquad z_u = \begin{cases} 0, & \text{centralized branch,} \\ 1, & \text{decentralized branch.} \end{cases} \tag{4}$$

The execution context for u is $h_u = \{(v, o_v) : (v, u) \in E_d\}$, where o_v denotes the completed output of a predecessor v. Let $\mathcal{B}_{\text{C}}$ and $\mathcal{B}_{\text{D}}$ denote the centralized and decentralized branch executors, respectively, and set $\mathcal{B}_0 = \mathcal{B}_{\text{C}}$ and $\mathcal{B}_1 = \mathcal{B}_{\text{D}}$. The output of unit u is then

$$o_u = \mathcal{B}_{z_u}(u, h_u). \tag{5}$$

After all required units have completed, a dataset-specific aggregation function Agg produces

$$\hat{y}(x; d) = \begin{cases} \mathcal{B}_{\text{C}}(x, \varnothing), & d = 0, \\ \text{Agg}(\{o_u : u \in U_d\}, G_d), & 0 < d < 1, \\ \mathcal{B}_{\text{D}}(x, \varnothing), & d = 1. \end{cases} \tag{6}$$

Equation 6 summarizes the complete input-to-output flow. For intermediate conditions, DoF configures decomposition and per-unit branch routing. The selected branch executes each dependency-ready unit, and the Coordinator aggregates the resulting outputs. Figure 2 illustrates this process.

3.2 DoF-Controlled Execution

Coordinator. The Coordinator is the control interface between the DoF configuration and the two branch implementations. For intermediate conditions, it applies the configured granularity, tracks task units and dependencies, and releases each unit when its predecessors finish. It supplies predecessor outputs as context and routes each unit to a branch according to the configured probability, supporting sequential and parallel profiles. At the two endpoints, it bypasses intermediate decomposition and routing and sends the original request directly to the corresponding branch. When either branch completes a unit, the Coordinator records its output and exposes it to downstream units whose dependencies have become satisfied. After all required units finish, the dataset-specific aggregation procedure combines the unit outputs into the final answer.

Centralized branch We instantiate the centralized branch with MetaAgent, which receives a task unit and context and executes a bounded plan–select–execute–review loop. It constructs a compact plan, selects workers using their task-relevant capabilities and success summaries, and reviews each result and the remaining plan. Insufficient results trigger a redo or replan; otherwise, the accumulated results are synthesized into the unit output.

Decentralized branch We instantiate the decentralized branch with AgentNet (Yang et al., 2025). AgentNet receives the assigned unit and its available predecessor context. At each internal decision point, the active agent selects one of three operations: `execute`, `split`, or `forward`. The router conditions this choice on the current request, the active agent's capabilities, neighboring-agent information, graph weights, and retrieved experience. `execute` produces a result locally, `split` creates further internal work for the decentralized branch, and `forward` transfers the request to another node. The Coordinator's decomposition determines the initial semantic scope of a delegated unit, whereas `split` is an AgentNet-internal operation that can further divide that unit during execution. The retained AgentNet state includes capability vectors, graph edge weights, router experience, and executor experience.

Thus, DoF controls how work is delegated and where each unit is processed, while MetaAgent and AgentNet retain their respective centralized and decentralized execution mechanisms.

3.3 Trace-Derived Organizational Measurements

The preceding subsections define how the framework produces an output. This subsection defines how its recorded coordination events are summarized. Let $c \in \mathcal{C}$ denote an evaluated condition, $r \in \mathcal{R} = \{\mathrm{C}, \mathrm{D}\}$ denote the centralized or decentralized branch, and $a \in \mathcal{P}$ denote a profile in the shared profile set $\mathcal{P}$. For condition c, branch r, and profile a, let $n_{c,r}(a)$ be the corresponding event count. Centralized events count worker-profile selections, whereas decentralized events count node activations along decentralized traces. When branch r contains at least one recorded event, its normalized branch-specific profile distribution is

$$p_{c,r}(a) = \frac{n_{c,r}(a)}{\sum_{a' \in \mathcal{P}} n_{c,r}(a')}. \tag{7}$$

Accordingly, cross-branch comparisons concern branch-specific profile composition rather than equality of work volume or specialization quality.

For two profile distributions P and Q aligned on the common support $\mathcal{P}$, we use base-2 Jensen–Shannon divergence (Lin, 1991):

$$\mathrm{JSD}(P, Q) = H\left(\frac{P+Q}{2}\right) - \frac{H(P) + H(Q)}{2}, \tag{8}$$

where $H(P) = -\sum_{a \in \mathcal{P}} P(a) \log_2 P(a)$ is the base-2 Shannon entropy. Therefore, $\mathrm{JSD}(P, Q) \in [0, 1]$. Conditions are aligned by common task identifiers before paired distribution comparisons are computed.

4 Experimental Setup

Model and branch implementations. All experiments use Qwen3-Next-80B (Qwen Team, 2025) and the same five-agent roster. The centralized and decentralized branches are instantiated with MetaAgent and AgentNet, respectively. AgentNet implements the decentralized coordination regime described in prior work (Yang et al., 2025). The model, agent roster, branch implementations, dataset-specific dependency mode, aggregation rule, and evaluation procedure remain fixed across DoF conditions. DoF changes decomposition granularity, the resulting task-unit schedule, and per-unit branch assignments. Detailed agent configurations and runtime settings are provided in Appendix B; the stable prompt contracts are summarized in Appendix B.4.

Benchmarks and retained states. The primary retained state is trained on a BBH collection and evaluated on BBH (Suzgun et al., 2023), AIME (HuggingFaceH4, 2025; OpenCompass, 2025; Zhang & Math-AI Team, 2026), MMLU-Pro (Wang et al., 2024b), RBench-T (Guo et al., 2025), and SMFR (Jwalapuram et al., 2026) collections. We additionally construct a retained state trained on an MMLU-Pro collection and evaluate it on MMLU-Pro. Training and evaluation items are disjoint in every setting. Each retained state is loaded before an evaluation sweep and remains fixed throughout that sweep.

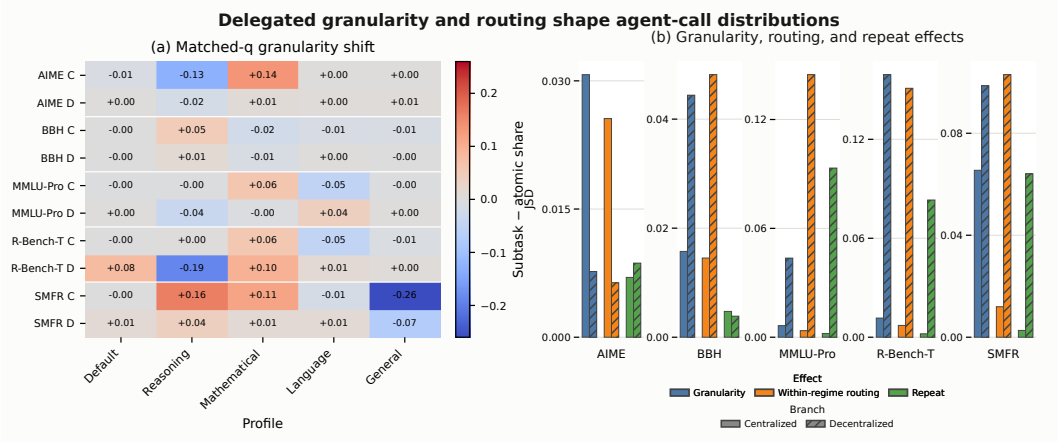

Figure 3: Delegated granularity and routing affect branch-specific profiles. Panel (a) reports the change in branch-specific profile composition from atomic-target to subtask-target delegation. Panel (b) compares matched-q granularity differences, within-regime routing differences, and same-DoF repeated-run effects. Colors identify the effect type, while opacity distinguishes the centralized and decentralized branches.

DoF configurations and repeated sweeps. Each dataset is evaluated with three repeated DoF sweeps (MMLU-Pro-trained has two). Each sweep evaluates fifteen DoF values covering the centralized endpoint, the decentralized endpoint, the atomic-target granularity regime, and the subtask-target granularity regime. The fifteen evaluated DoF values are

$$\{0, 0.0001, 0.1, 0.2, 0.25, 0.4, 0.5, 0.5001, 0.6, 0.7, 0.75, 0.85, 0.9, 0.9999, 1\}.$$

The matched target-routing pairs used for granularity comparisons are

$$(0.0001, 0.5001),\ (0.1, 0.6),\ (0.2, 0.7),\ (0.25, 0.75),\ (0.4, 0.9),\ (0.5, 0.9999).$$

These pairs have matched target probabilities of task-unit assignment to the decentralized branch. The evaluation-allocation matrix are provided in Appendix C.

5 Results

We organize the results as a progression from coordination configuration to organizational conditions and task outcomes. We first measure the profile changes associated with delegated granularity and within-regime routing, then compare the stronger organizational differences associated with task domain and retained experience, and finally relate these profiles to task-dependent accuracy peaks and run-to-run consistency.

5.1 Delegated granularity and routing affect branch-specific profiles

Figure 3 compares these two interventions using normalized profile distributions within each execution branch. The matched-q comparison contrasts atomic-target and subtask-target conditions at the same target branch-assignment probability. Values are computed as the subtask-target share minus the atomic-target share. The within-regime comparison contrasts DoF values while holding the granularity regime fixed. These comparisons assess profile differences under the stated controls.

Changing delegated granularity produces structured but limited profile reallocation. In Figure 3(a), the subtask-target minus atomic-target differences are concentrated in selected profiles and dataset–branch pairs. The clearest signed changes occur in the centralized AIME, RBench-T, and SMFR rows, whereas BBH and MMLU-Pro remain closer across the matched pairs. Changing the semantic scope of a delegated unit does not erase the branch's existing profile preference. Instead, it changes how that preference is expressed for particular task domains. The effect is therefore consistent with granularity acting as a conditioning signal on an already learned branch-specific profile rather than creating a new organization independently of task and branch.

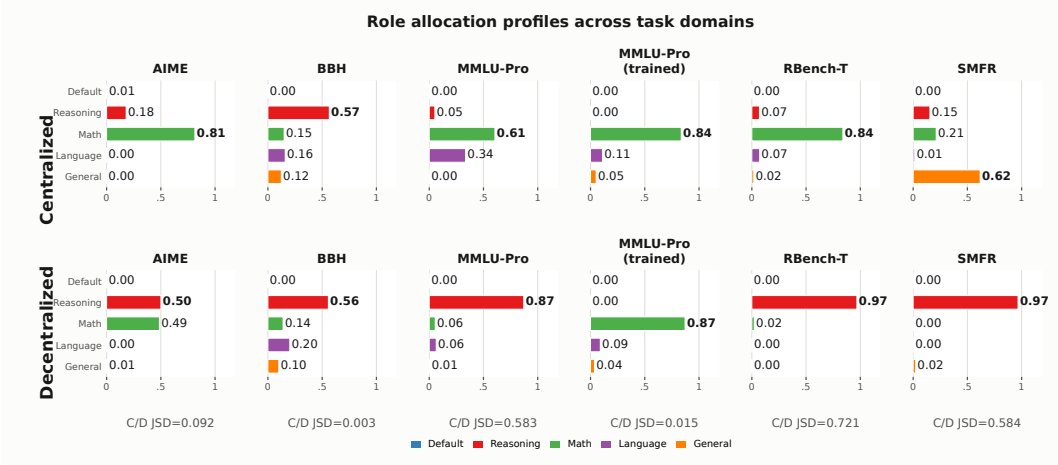

Figure 4: Branch-specific profiles across task domains and retained agent states. Bars show branch-normalized profile-call shares pooled over intermediate DoF conditions ($0 < d < 1$). The upper row reports centralized worker-profile selections, and the lower row reports decentralized node activations. The two MMLU-Pro columns distinguish the BBH-trained and MMLU-Pro-trained retained states. The branches are therefore compared by profile composition rather than by absolute workload volume.

Changing DoF within a fixed granularity regime has a comparable, also bounded effect. Figure 3(b) places within-regime routing JSD beside matched-q granularity JSD and repeated-run JSD. The within-regime and granularity differences are of similar order for several dataset–branch pairs, indicating that changing the target branch-routing probability can affect a branch-specific profile even without crossing the atomic–subtask boundary. At the same time, all plotted JSD values are below 0.2. Both interventions therefore produce modest reallocations within broadly similar distributions rather than transitions between qualitatively different branch-specific profiles. The repeated-run bars provide the reference scale: configuration changes are observable but remain bounded relative to the overall branch-specific profile. This result suggests that DoF tunes an existing coordination landscape rather than replacing one stable branch-specific profile with another. These bounded configuration effects motivate the next comparison: whether task domain and retained experience impose stronger organizational conditions on the branch-specific profiles.

5.2 Branch-specific profiles vary across task domains and retained states

Task domain and execution branch are associated with distinct profile compositions. Figure 4 shows branch-normalized profile shares pooled over intermediate DoF conditions. Mathematical profiles dominate centralized AIME and RBench-T, centralized SMFR emphasizes the general profile, and decentralized RBench-T and SMFR concentrate on reasoning. These differences are larger and more systematic than the modest within-regime shifts in Figure 3. DoF adjusts the branch-specific profile, but task domain and execution branch provide the stronger organizational conditions under which those adjustments occur.

Retained experience is associated with a reorganization of branch-specific profiles. On the same MMLU-Pro evaluation set, the BBH-trained state gives a centralized–decentralized JSD of approximately 0.583, whereas the MMLU-Pro-trained state reduces this value to approximately 0.015. The in-domain retained state therefore brings the two branch-specific profiles substantially closer. This reduction is large relative to the within-regime DoF effects, which suggests that retained experience can reorganize the relationship between branches more strongly than a local change in target branch-routing probability. The derived cooperation paths provide a descriptive view consistent with this aggregate profile comparison, but do not establish a component-wise causal mechanism. Together with the domain differences above, this result indicates that task domain and retained experience impose stronger organizational conditions than the local granularity and routing adjustments examined in the first subsection.

	BBH	AIME	MMLU-Pro (BBH-trained)	MMLU-Pro (MMLU-Pro-trained)	R-Bench-T	SMFR
$d = 0$	$0.217^{+0.013}_{-0.017}$	$0.619^{+0.037}_{-0.019}$	$0.662^{+0.004}_{-0.007}$	$0.705^{+0.003}_{-0.003}$	$0.558^{+0.026}_{-0.052}$	$0.243^{+0.007}_{-0.014}$
$d = 0.0001$	$0.750^{+0.010}_{-0.010}$	$0.600^{+0.022}_{-0.022}$	$0.824^{+0.009}_{-0.009}$	$0.839^{+0.018}_{-0.018}$	$0.633^{+0.022}_{-0.026}$	$\mathbf{0.411}^{+0.036}_{-0.036}$
$d = 0.1$	$0.807^{+0.003}_{-0.007}$	$0.611^{+0.022}_{-0.033}$	$0.832^{+0.013}_{-0.010}$	$0.827^{+0.024}_{-0.024}$	$0.690^{+0.006}_{-0.012}$	$0.378^{+0.069}_{-0.056}$
$d = 0.2$	$0.813^{+0.007}_{-0.003}$	$0.607^{+0.015}_{-0.030}$	$0.835^{+0.010}_{-0.013}$	$0.842^{+0.015}_{-0.015}$	$0.704^{+0.022}_{-0.014}$	$0.396^{+0.010}_{-0.010}$
$d = 0.25$	$0.815^{+0.005}_{-0.005}$	$0.594^{+0.028}_{-0.028}$	$0.838^{+0.007}_{-0.010}$	$\mathbf{0.848}^{+0.009}_{-0.009}$	$0.692^{+0.010}_{-0.014}$	$0.375^{+0.042}_{-0.052}$
$d = 0.4$	$0.810^{+0.060}_{-0.040}$	$0.596^{+0.015}_{-0.019}$	$0.842^{+0.003}_{-0.003}$	$0.839^{+0.006}_{-0.006}$	$0.696^{+0.006}_{-0.006}$	$0.399^{+0.017}_{-0.014}$
$d = 0.5$	$0.883^{+0.017}_{-0.023}$	$0.581^{+0.007}_{-0.004}$	$\mathbf{0.856}^{+0.007}_{-0.004}$	$0.839^{+0.006}_{-0.006}$	$0.698^{+0.022}_{-0.026}$	$0.344^{+0.010}_{-0.010}$
$d = 0.5001$	$0.747^{+0.023}_{-0.037}$	$0.659^{+0.052}_{-0.037}$	$0.841^{+0.010}_{-0.007}$	$\mathbf{0.848}^{+0.003}_{-0.003}$	$\mathbf{0.730}^{+0.050}_{-0.046}$	$0.365^{+0.021}_{-0.031}$
$d = 0.6$	$0.787^{+0.023}_{-0.027}$	$\mathbf{0.667}^{+0.011}_{-0.011}$	$0.818^{+0.021}_{-0.021}$	$0.830^{+0.003}_{-0.003}$	$0.704^{+0.022}_{-0.020}$	$0.354^{+0.031}_{-0.052}$
$d = 0.7$	$0.827^{+0.053}_{-0.027}$	$0.604^{+0.019}_{-0.026}$	$0.826^{+0.013}_{-0.016}$	$0.833^{+0.000}_{+0.000}$	$0.679^{+0.024}_{-0.030}$	$0.396^{+0.021}_{-0.010}$
$d = 0.75$	$0.840^{+0.040}_{-0.030}$	$0.633^{+0.011}_{-0.011}$	$0.827^{+0.030}_{-0.024}$	$0.839^{+0.012}_{-0.012}$	$0.710^{+0.022}_{-0.014}$	$0.385^{+0.021}_{-0.042}$
$d = 0.85$	$0.867^{+0.023}_{-0.017}$	$0.619^{+0.015}_{-0.019}$	$0.841^{+0.010}_{-0.007}$	$0.824^{+0.021}_{-0.021}$	$0.698^{+0.004}_{-0.002}$	$0.382^{+0.045}_{-0.049}$
$d = 0.9$	$0.873^{+0.017}_{-0.023}$	$0.563^{+0.037}_{-0.041}$	$0.830^{+0.015}_{-0.015}$	$0.830^{+0.003}_{-0.003}$	$0.708^{+0.018}_{-0.024}$	$0.389^{+0.028}_{-0.024}$
$d = 0.9999$	$0.887^{+0.023}_{-0.017}$	$0.581^{+0.041}_{-0.026}$	$0.839^{+0.006}_{-0.012}$	$0.586^{+0.033}_{-0.033}$	$0.683^{+0.020}_{-0.034}$	$0.295^{+0.122}_{-0.201}$
$d = 1$	$\mathbf{0.953}^{+0.007}_{-0.003}$	$0.481^{+0.041}_{-0.026}$	$0.841^{+0.016}_{-0.020}$	$0.643^{+0.179}_{-0.179}$	$0.554^{+0.006}_{-0.006}$	$0.042^{+0.010}_{-0.010}$
$\mathrm{Var}_{\mathrm{subtask}}/\mathrm{Var}_{\mathrm{atomic}}$	1.108	1.854	3.183	1.174	2.432	4.012

Table 1: Accuracy across the canonical 15 DoF conditions. Cell deviations are observed run ranges around the mean, not confidence intervals; red is mean-to-min and green is max-to-mean.

The preceding comparisons establish the organizational side of the coordination landscape: local configuration changes are bounded, whereas task domain and retained experience are associated with stronger profile differences. We next examine how this task- and state-conditioned organization is reflected in accuracy peaks and run-to-run consistency.

5.3 Coordination configurations affect accuracy consistency and task performance

The performance consequence of coordination is task dependent. Across the 15 DoF values, no configuration dominates. BBH peaks at the decentralized endpoint, while AIME, both MMLU-Pro settings, RBench-T, and SMFR attain their highest observed mean accuracy between the endpoints. The MMLU-Pro-trained setting has two tied intermediate peaks. These results favor intermediate configurations in five of six settings, but do not imply that mixing branches causes the gains. The BBH-trained MMLU-Pro peak uses only the decentralized branch, whereas the SMFR peak is nearly centralized. Their advantage over undecomposed endpoints is consistent with a contribution from task decomposition, although changes in execution and aggregation prevent us from isolating that contribution. Together with the distinct branch-specific profiles across tasks and retained states, these peaks support task-conditioned configuration rather than a universally best DoF.

Atomic-target conditions are generally more consistent across repeated runs than subtask-target conditions. For the datasets with valid complete-run variance estimates, the subtask-to-atomic variance ratios exceed one in every reported comparison. This pattern suggests that broader subtask delegation exposes more internal choices to the receiving branch, making small generation differences more likely to propagate into different outcomes. It also separates performance level from performance stability. An intermediate configuration may have a high mean accuracy without being the most reproducible configuration across runs.

6 Conclusion

In this work, we introduce Degrees of Freedom (DoF), a controlled framework that connects centralized and decentralized LLM-based multi-agent systems through the size of decision space delegated to downstream agents. Experiments establish a coherent sequence. Configuration changes modestly affect branch-specific profiles. Task domain, execution branch, and retained state provide the stronger organizational conditions. Execution traces show how these profiles unfold as coordination events. The resulting organization is associated with both task performance and stability across runs. Together, these findings provides a new aspect for creating, comparing and evaluating LLM-based MAS.

ETHICS STATEMENT

This work evaluates automated multi-agent systems on established reasoning benchmarks and does not involve new human-subject data collection. We do not collect personal or sensitive information, and the experiments use publicly available benchmark data together with model-generated execution records. The study concerns general reasoning, question answering, and benchmarked interactive tasks rather than decisions about individuals or demographic groups. No component is designed to target, rank, or disadvantage any population. We use the benchmark collections in accordance with their intended research settings and applicable licenses, and we report the methodology, evaluation protocol, and limitations to support transparent and reproducible research.

REPRODUCIBILITY STATEMENT

The experiment implementation defines the complete DoF schedule, piecewise routing functions, Coordinator behavior, MetaAgent and AgentNet branch interfaces, dataset profiles, aggregation rules, and trace-derived organizational measurements. The experimental protocol specifies the model, five-agent roster, retained states, evaluation collections, repeated DoF sweeps, matched-routing comparisons, error-handling rules, and sampling quotas. Appendix B summarizes the implementation settings and stable prompt contracts, Appendix C documents the datasets and evaluation protocol, and Appendix F describes the derived records and current trace-availability limits. Subject to the venue's artifact policy and applicable benchmark licenses, the experiment code, analysis scripts, sampled task identifiers, split provenance, and derived result tables will be released after acceptance. Raw model prompts and execution traces will be released only when their contents can be shared consistently with benchmark, privacy, credential, and retained-memory requirements.

AI USE STATEMENT

Separate from the language models evaluated in the experiments, LLM-based assistants were used during manuscript preparation to improve grammar, clarity, organization, and LaTeX formatting. They were also used to assist with localized restructuring of exposition and appendix presentation. The assistants did not generate or alter the experimental data, benchmark labels, reported numerical results, model configurations, or methodological claims. All assisted text was manually reviewed by the authors, who remain responsible for the accuracy, interpretation, and final content of the manuscript.

REFERENCES

Mert Cemri, Melissa Z. Pan, Shuyi Yang, Lakshya A. Agrawal, Bhavya Chopra, Rishabh Tiwari, Kurt Keutzer, Aditya Parameswaran, Dan Klein, Kannan Ramchandran, Matei Zaharia, Joseph E. Gonzalez, and Ion Stoica. Why do multi-agent LLM systems fail? In *Advances in Neural Information Processing Systems*, 2025. URL https://arxiv.org/abs/2503.13657. Datasets and Benchmarks Track.

Huan Chen, Xiang Song, Jian Jin, Pan Ren, and Liang-Jie Zhang. Toward an organizational science of multi-agent LLM systems: Decoupling who, how, and which algorithm, 2026. URL https://arxiv.org/abs/2607.25446.

Weize Chen, Yusheng Su, Jingwei Zuo, Cheng Yang, Chenfei Yuan, Chi-Min Chan, Heyang Yu, Yaxi Lu, Yi-Hsin Hung, Chen Qian, Yujia Qin, Xin Cong, Ruobing Xie, Zhiyuan Liu, Maosong Sun, and Jie Zhou. AgentVerse: Facilitating multi-agent collaboration and exploring emergent behaviors. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/578e65cdee35d00c708d4c64bce32971-Abstract-Conference.html.

Zhiqing Cui, Haotong Xie, Jiahao Yuan, Cheng Yang, Hanqing Wang, Yuxin Wu, Yifan Wu, Siru Zhong, Tao Yu, Yifu Guo, Siyu Zhang, Xinlei Yu, Qibing Ren, and Usman Naseem. Uno-Orchestra: Parsimonious agent routing via selective delegation, 2026. URL https://arxiv.org/abs/2605.05007.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. Improving factuality and reasoning in language models through multiagent debate. In *Proceedings of the*

41st International Conference on Machine Learning, volume 235 of *Proceedings of Machine Learning Research*, pp. 11733–11763. PMLR, 2024. URL https://proceedings.mlr.press/v235/du24e.html.

Hang Gao and Yongfeng Zhang. INMS: Memory sharing for large language model based agents, 2024. URL https://arxiv.org/abs/2404.09982. Version 3, revised 4 March 2026.

Xueping Gao. Compositional skill routing for LLM agents: Decompose, retrieve, and compose, 2026. URL https://arxiv.org/abs/2606.18051.

Zhaoyang Guan, Huixi Cao, Ming Zhong, Yin Wang, Guanyu Liu, Eric Yang, Lynn Ai, Yongxin Ni, and Bill Shi. Symphony-Coord: Adaptive routing for multi-agent LLM systems, 2026. URL https://arxiv.org/abs/2602.00966.

Meng-Hao Guo, Jiajun Xu, Yi Zhang, Jiaxi Song, Haoyang Peng, Yi-Xuan Deng, Xinzhi Dong, Kiyohiro Nakayama, Zhengyang Geng, Chen Wang, Bolin Ni, Guo-Wei Yang, Yongming Rao, Houwen Peng, Han Hu, Gordon Wetzstein, and Shi-Min Hu. RBench: Graduate-level multi-disciplinary benchmarks for LLM & MLLM complex reasoning evaluation. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 21184–21201, 2025. URL https://proceedings.mlr.press/v267/guo25r.html.

Taicheng Guo, Xiuying Chen, Yaqi Wang, Ruidi Chang, Shichao Pei, Nitesh V. Chawla, Olaf Wiest, and Xiangliang Zhang. Large language model based multi-agents: A survey of progress and challenges. In Kate Larson (ed.), *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, pp. 8048–8057. International Joint Conferences on Artificial Intelligence Organization, August 2024. doi: 10.24963/ijcai.2024/890. URL https://doi.org/10.24963/ijcai.2024/890. Survey Track.

Sirui Hong, Mingchen Zhuge, Jiaqi Chen, Xiawu Zheng, Yuheng Cheng, Ceyao Zhang, Jinlin Wang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, Chenyu Ran, Lingfeng Xiao, Chenglin Wu, and Jürgen Schmidhuber. MetaGPT: Meta programming for a multi-agent collaborative framework. In *International Conference on Learning Representations*, pp. 23247–23275, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/6507b115562bb0a305f1958ccc87355a-Abstract-Conference.html.

Zhipeng Hou et al. Halo: Hierarchical autonomous logic-oriented orchestration for multi-agent llm systems, 2025. URL https://arxiv.org/abs/2505.13516.

HuggingFaceH4. AIME 2024. Hugging Face dataset, 2025. URL https://huggingface.co/datasets/HuggingFaceH4/aime_2024. 30 transcribed problems from 2024 AIME I and II.

Prathyusha Jwalapuram, Hehai Lin, Chuyuan Li, Fangkai Jiao, Sudong Wang, Yifei Ming, Zixuan Ke, Chengwei Qin, Giuseppe Carenini, and Shafiq Joty. The illusion of multi-agent advantage, 2026. URL https://arxiv.org/abs/2606.13003.

Guohao Li, Hasan Abed Al Kader Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. CAMEL: Communicative agents for "mind" exploration of large language model society. In *Advances in Neural Information Processing Systems*, volume 36, pp. 51991–52008. Curran Associates, Inc., 2023. doi: 10.52202/075280-2264. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/a3621ee907def47c1b952ade25c67698-Abstract-Conference.html.

Yunxuan Li, Yibing Du, Jiageng Zhang, Le Hou, Peter Grabowski, Yeqing Li, and Eugene Ie. Improving multi-agent debate with sparse communication topology. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 7281–7294, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.427. URL https://aclanthology.org/2024.findings-emnlp.427/.

Hehai Lin, Yu Yan, Zixuan Wang, Bo Xu, Sudong Wang, Weiquan Huang, Ruochen Zhao, Minzhi Li, and Chengwei Qin. Unified-MAS: Universally generating domain-specific nodes for empowering automatic multi-agent systems, 2026a. URL https://arxiv.org/abs/2603.21475.

Hehai Lin, Qi Yang, and Chengwei Qin. Skill-MAS: Evolving meta-skill for automatic multi-agent systems, 2026b. URL https://arxiv.org/abs/2606.18837.

Jianhua Lin. Divergence measures based on the shannon entropy. *IEEE Transactions on Information Theory*, 37(1):145–151, January 1991. doi: 10.1109/18.61115.

Yang Liu, Jinxuan Cai, Yishen Li, Qi Meng, Zedi Liu, Xin Li, Chen Qian, Chuan Shi, and Cheng Yang. MASFactory: A graph-centric framework for orchestrating LLM-based multi-agent systems with vibe graphing. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pp. 357–366, San Diego, California, United States, 2026. Association for Computational Linguistics. doi: 10.18653/v1/2026.acl-demo.35. URL https://aclanthology.org/2026.acl-demo.35/.

Zijun Liu, Yanzhe Zhang, Peng Li, Yang Liu, and Diyi Yang. A dynamic LLM-powered agent network for task-oriented agent collaboration. In *Proceedings of the First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=TYIfkvitPt.

Yuzhen Mao et al. Decentralized multi-agent systems with shared context, 2026. URL https://arxiv.org/abs/2606.10662.

OpenCompass. AIME 2025 Dataset. Hugging Face dataset, 2025. URL https://huggingface.co/datasets/opencompass/AIME2025. AIME 2025-I and II, 15 problems each.

Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, pp. 1–22. Association for Computing Machinery, 2023. doi: 10.1145/3586183.3606763. URL https://doi.org/10.1145/3586183.3606763.

Archiki Prasad, Alexander Koller, Mareike Hartmann, Peter Clark, Ashish Sabharwal, Mohit Bansal, and Tushar Khot. ADaPT: As-needed decomposition and planning with language models. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pp. 4226–4252, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-naacl.264. URL https://aclanthology.org/2024.findings-naacl.264/.

Qwen Team. Qwen3-Next-80B-A3B-Instruct model card. Hugging Face model repository, 2025. URL https://huggingface.co/Qwen/Qwen3-Next-80B-A3B-Instruct. Apache-2.0; revision 9c7f2fbe84465e40164a94cc16cd30b6999b0cc7; accessed 2026-09-19.

Rakibul Hasan Rajib, Mengxin Zheng, and Qian Lou. Learning what to retain: Gated-memory routing for efficient collaboration in multi-agent LLM systems, 2026. URL https://arxiv.org/abs/2609.00237.

Linus Sander, Fengjunjie Pan, Vahid Zolfaghari, Andre Schamschurko, Nenad Petrovic, and Alois Knoll. Scaling LLM-driven multi-agent systems: Design principles and architectural scalability analysis, 2026. URL https://arxiv.org/abs/2607.27942.

Pedro Santos et al. Centralized training with hybrid execution in multi-agent reinforcement learning, 2023. URL https://arxiv.org/abs/2210.06274.

Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. HuggingGPT: Solving AI tasks with ChatGPT and its friends in Hugging Face. In *Advances in Neural Information Processing Systems*, volume 36, pp. 38154–38180. Curran Associates, Inc., 2023. doi: 10.52202/075280-1657. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/77c33e6a367922d003ff102ffb92b658-Abstract-Conference.html.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36, pp. 8634–8652. Curran Associates, Inc., 2023. doi: 10.52202/075280-0377. URL https://doi.org/10.52202/075280-0377.

Andries Petrus Smit, Nathan Grinsztajn, Paul Duckworth, Thomas D. Barrett, and Arnu Pretorius. Should we be going MAD? a look at multi-agent debate strategies for LLMs. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 45883–45905. PMLR, 2024. URL https://proceedings.mlr.press/v235/smit24a.html.

Toshiharu Sugawara et al. Reducing redundant computation in multi-agent coordination through locally centralized execution, 2024. URL https://arxiv.org/abs/2404.13096.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc Le, Ed Chi, Denny Zhou, and Jason Wei. Challenging BIG-bench tasks and whether chain-of-thought can solve them. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 13003–13051, 2023. doi: 10.18653/v1/2023.findings-acl.824. URL https://aclanthology.org/2023.findings-acl.824/.

Tianyu Wang et al. Megaagent: A large-scale autonomous llm-based multi-agent system without predefined sops, 2024a. URL https://arxiv.org/abs/2408.09955.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. MMLU-Pro: A more robust and challenging multi-task language understanding benchmark. In *Advances in Neural Information Processing Systems*, volume 37, pp. 95266–95290, 2024b. doi: 10.52202/079017-3018. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/ad236edc564f3e3156e1b2feafb99a24-Abstract-Datasets_and_Benchmarks_Track.html.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, Ahmed Hassan Awadallah, Ryen W. White, Doug Burger, and Chi Wang. AutoGen: Enabling next-gen LLM applications via multi-agent conversations. In *Proceedings of the First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=uAjxFFing2.

Yingxuan Yang, Huacan Chai, Shuai Shao, Yuanyi Song, Siyuan Qi, Renting Rui, and Weinan Zhang. AgentNet: Decentralized evolutionary coordination for LLM-based multi-agent systems. In *Advances in Neural Information Processing Systems*, volume 38, pp. 118758–118785. Neural Information Processing Systems Foundation, Inc., 2025. doi: 10.52202/085713-3578. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/9a379c1b05793d1c42dc832269834515-Abstract-Conference.html.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. In *Advances in Neural Information Processing Systems*, volume 36, pp. 11809–11822. Curran Associates, Inc., 2023. doi: 10.52202/075280-0517. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/271db9922b8d1f4dd7aaef84ed5ac703-Abstract-Conference.html.

Geunbin Yu. AdaptOrch: Task-adaptive multi-agent orchestration in the era of LLM performance convergence, 2026. URL https://arxiv.org/abs/2602.16873.

Yanwei Yue, Guibin Zhang, Boyang Liu, Guancheng Wan, Kun Wang, Dawei Cheng, and Yiyan Qi. MasRouter: Learning to route LLMs for multi-agent systems. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15549–15572, Vienna, Austria, July 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.acl-long.757. URL https://aclanthology.org/2025.acl-long.757/.

Guibin Zhang, Yanwei Yue, Zhixun Li, Sukwon Yun, Guancheng Wan, Kun Wang, Dawei Cheng, Jeffrey Xu Yu, and Tianlong Chen. Cut the crap: An economical communication pipeline for LLM-based multi-agent systems. In *International Conference on Learning Representations*, 2025a. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/bbc461518c59a2a8d64e70e2c38c4a0e-Abstract-Conference.html.

Guibin Zhang, Yanwei Yue, Xiangguo Sun, Guancheng Wan, Miao Yu, Junfeng Fang, Kun Wang, Tianlong Chen, and Dawei Cheng. G-Designer: Architecting multi-agent communication topologies via graph neural networks. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 76678–76692. PMLR, 2025b. URL https://proceedings.mlr.press/v267/zhang25cu.html.

Wentao Zhang et al. Agentorchestra: A hierarchical multi-agent framework for general-purpose task solving, 2025c. URL https://arxiv.org/abs/2506.12508.

Yifan Zhang and Math-AI Team. American invitational mathematics examination (AIME) 2026. Hugging Face dataset, 2026. URL https://huggingface.co/datasets/math-ai/aime26.

Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Gaetan Lin, Yong-Jin Liu, and Gao Huang. ExpeL: LLM agents are experiential learners. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 19632–19642, 2024. doi: 10.1609/aaai.v38i17.29936. URL https://doi.org/10.1609/aaai.v38i17.29936.

Mingchen Zhuge, Wenyi Wang, Louis Kirsch, Francesco Faccio, Dmitrii Khizbullin, and Jürgen Schmidhuber. GPTSwarm: Language agents as optimizable graphs. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 62743–62767. PMLR, 2024. URL https://proceedings.mlr.press/v235/zhuge24a.html.

A Supplementary Roadmap

The supplementary material is organized as follows. Appendix B reports the implementation settings and the stable prompt contracts used by the Coordinator, MetaAgent, and AgentNet. Appendix C documents the datasets, retained states, repeated-sweep allocation, evaluated DoF conditions, and error-handling protocol. Appendix D gives the subject-level MMLU-Pro sampling quotas. Appendix E evaluates how closely realized branch assignments follow the target DoF routing probabilities. Appendix F inventories the derived analysis artifacts and states the limits of the currently available trace archive.

B Implementation Details and Prompt Contracts

B.1 Model and agent roster

All experiments use Qwen3-Next-80B. The roster contains five agent profiles with IDs 0–4: default, reasoning, mathematical, language, and general. All profiles use the same model, AgentNet router interface, executor interface, memory mechanisms, and retrieval procedures. They differ only in their initialized capability vectors.

Each capability vector contains seven dimensions: reasoning, mathematical, language, knowledge, sequence, spatial, and inference. The default profile assigns 0.6 to every dimension. The reasoning, mathematical, and language profiles assign 1.0 to the corresponding dimension and retain 0.6 on the other dimensions. The general profile assigns 0.7 to knowledge, sequence, spatial, and inference and 0.6 to reasoning, mathematical, and language. These vectors provide capability information to branch-internal agent selection.

B.2 Agent graph and retained state

The AgentNet branch is initialized with a complete directed graph over the five agents. Each directed edge starts with weight 1.0. During retained-state training, the state can include capability values, task-specific success statistics, edge weights, AgentNet router experience, and executor experience. Capability values and success statistics inform action selection. Edge weights and router experience inform AgentNet's branch-internal routing decisions, including the selection among `execute`, `split`, and `forward`, while executor experience supports task execution.

Edge weights are updated from forwarding outcomes and execution time. Edges whose weights fall below the active threshold are removed from subsequent neighbor information. The primary evaluation sweeps load a retained state before evaluation. The loaded state remains fixed during each sweep.

B.3 Execution and model-call settings

The AgentNet configuration allows up to three forward hops and five executions per task. These limits bound internal forwarding and execution within the decentralized branch. The MetaAgent configuration allows up to twelve planning rounds and one redo for a step. These limits bound planning and revision within the centralized branch. The AgentNet router and executor retrieve three experiences by default, and the executor memory limit is 40.

The Coordinator and MetaAgent share an LLM client configured with a maximum of 2,048 output tokens and temperature 0.3. AgentNet calls the same underlying model with a maximum of 2,048 output tokens, temperature 0, and top-p of 1.0. Thus, the model identity is shared across the framework, while the branch implementations retain their original decoding interfaces.

B.4 Prompt contracts

The prompts use structured interfaces so that decomposition, branch-internal decisions, and aggregation can be recorded consistently. The descriptions below report the stable instructions and output schemas. Dynamic task text, retrieved experiences, agent states, and prior results are inserted at runtime.

Coordinator decomposition. The decomposition prompt receives the current task description, the requested granularity (subtask or atomic operation), and the dataset dependency mode. For a sequential profile, it states that later units may use earlier results; for a parallel profile, it requires independent units. The prompt explicitly prohibits solving the task, emitting derivations, or returning intermediate or final numeric answers. It requests only short, instruction-like unit descriptions in a JSON list with fields `id` and `description`. If parsing fails, the Coordinator retries once with a JSON-only reminder. A second failure falls back to one unit equal to the current task description.

Coordinator aggregation. For LLM-based synthesis, the aggregation prompt receives the original task, the ordered unit descriptions, their completed outputs, and any dataset-specific answer-format constraint. It instructs the model to assemble or extract the final answer only from the supplied results and prohibits new computation. Profiles using last-leaf aggregation return the final leaf result directly and do not invoke this prompt.

MetaAgent planning and selection. The planning prompt asks for the fewest necessary steps and prefers a single step unless the request contains separable reasoning stages. Its JSON response contains `needs_decomposition`, `estimated_difficulty`, and an ordered `plan`. For each plan step, the selection prompt supplies the task type, full plan, current progress, and the agents' task-relevant abilities, success summaries, and current loads. It returns `agent_id` and `reason`; invalid selections fall back to capability-based selection.

MetaAgent execution and review. The selected worker first produces a scoped draft and then performs a separate verification and refinement pass. Intermediate units are instructed not to answer the original task, whereas the final unit receives the dataset-specific output constraint. The review prompt checks reasoning, correctness, completeness, and scope. It returns one of three JSON decisions: `completed` with the final answer, `continue` with a confirmed conclusion and revised remaining plan, or `redo_current` with a revised step and failure feedback.

AgentNet routing and execution. The AgentNet router receives the major task, current unit, completed progress, active-agent state, neighboring-agent information, and retrieved experiences. Its structured response contains `DECISION`, `REASON`, `EXECUTABLE`, `DELEGATE`, `DESCRIPTION`, and `NEXT_AGENT_ID`. The decision is normalized to `execute`, `split`, or `forward`; invalid decisions default to local execution. The executor first generates a structured `RESULT` thought and

Retained state	Evaluation collection	Evaluation subset	Items	Summary runs	Dependencies
BBH-trained	BBH	Held-out BBH	100	3	Parallel
BBH-trained	AIME	AIME 2024–2026 snapshots	90	3	Sequential
BBH-trained	MMLU-Pro	Validation sample	168	3	Parallel
MMLU-Pro-trained	MMLU-Pro	Validation sample	168	2	Parallel
BBH-trained	R-Bench-T	English, text-only	168	3	Sequential
BBH-trained	SMFR	Two-investor subset	96	3	Sequential

Table 2: Training-state and evaluation allocation for the repeated DoF sweeps, in the canonical analysis order. The summary-run cohort is $3/3/3/2/3/3$. Each sweep targets the same fifteen DoF conditions; failed or missing conditions remain explicit.

then executes either the current request or an internally split unit under the dataset answer constraints. A separate completion check returns whether the request is complete and, if not, which agent should continue.

Disclosure scope. The appendix reports the reusable contracts rather than task-populated prompt dumps. Runtime prompts can contain benchmark questions, retained experiences, intermediate generations, and agent-state values; configuration files can also contain service endpoints and checkpoint locations. These values are excluded because they are not necessary to specify the method and require separate license, privacy, and credential review before artifact release.

B.5 Dataset profiles

Dataset profiles define the decomposition mode, final-answer format, answer extraction rule, and aggregation strategy. BBH and MMLU-Pro use parallel task-unit dependencies. AIME, RBench-T, and SMFR use sequential dependencies. Dataset-specific answer extraction and correctness evaluation are applied after the final output is produced.

C Dataset and Protocol Details

C.1 Evaluation collections and sweep allocation

The primary BBH-trained retained state uses a disjoint BBH training collection of 634 items. It is evaluated on BBH, AIME, MMLU-Pro, RBench-T, and SMFR. The additional MMLU-Pro-trained retained state uses a disjoint subject-proportional training sample of 600 items and is evaluated on a separate MMLU-Pro sample. Table 2 summarizes the evaluation allocation used by the paper-facing analyses.

The subject quotas for the MMLU-Pro adaptation are listed in Appendix D. Training and evaluation items are disjoint within each retained-state setting.

C.2 Run and error handling

The planned valid-run cohort sizes are the summary-run counts in Table 2. A condition with an incomplete planned cohort is retained in the condition-level record and excluded from the variance summaries. Failed or null conditions retain their error state, and missing accuracy is not imputed. Accuracy means use every valid observed run for a condition. The variance comparison includes a DoF condition only when its complete planned cohort is valid.

C.3 Trace instrumentation and available records

The execution runner supports a full per-task trace schema containing the task, DoF configuration, decomposition log, unit dependencies, branch assignments, branch-internal events, unit outputs, aggregation log, final result, and correctness. The analysis checkout used for the present paper retains condition summaries and derived task-level cooperation paths, but not the raw per-task trace JSON files, original question text, generated answer text, dependency graphs, or branch communication

content. Consequently, the reported quantitative analyses use only fields preserved in the summaries and path records and do not reconstruct unavailable answer-level traces.

The derived records preserve the dataset/state setting, run, task identifier, DoF value, correctness, condition accuracy, encoded cooperation path, and path length. Condition summaries additionally preserve the task count, average number of routed units, realized centralized assignment fraction, and any condition-level error. These fields support task alignment, repeated-run analysis, matched-routing comparisons, branch-profile summaries, routing-fidelity checks, and retained-state comparisons.

D MMLU-Pro Sampling Quotas

The 600-item training and 168-item evaluation samples preserve the original fourteen-category proportions using largest-remainder allocation (Table 3). The two samples are disjoint.

Subject	Train	Eval.	Subject	Train	Eval.
Math	67	19	Physics	65	18
Chemistry	57	16	Law	55	15
Engineering	48	14	Other	46	13
Economics	42	12	Health	41	11
Psychology	40	11	Business	39	11
Biology	36	10	Philosophy	25	7
Computer Science	20	6	History	19	5

Table 3: Subject-proportional MMLU-Pro sampling quotas.

E Supplemental Routing-Control Validation

The DoF schedule specifies a target probability $q(d)$ that an eligible task unit is assigned to the decentralized branch. Because each task produces a finite number of routable units and assignments are sampled independently, the realized fraction can deviate from this target within a run. Figure 5 compares the target probability with the realized decentralized fraction, computed as one minus the condition-level centralized assignment ratio, for every valid intermediate condition.

The comparison evaluates control fidelity rather than task accuracy. Across 216 valid intermediate run–condition points, the mean absolute deviation from the routing target is 0.0098. Deviations are expected when a condition contains a finite number of units, especially for coarser subtask decompositions that create fewer assignment draws. The machine-readable sidecar retains the complete point set and all 39 exclusions from the 255-condition planned matrix (34 endpoints, four failed intermediate conditions, and one missing intermediate condition), together with run identities and absolute-deviation summaries.

F Analysis Artifacts and Availability

The paper-facing analysis is derived from two surviving record types. Condition summaries provide accuracy, task count, average routed-unit count, realized centralized assignment fraction, and explicit condition errors. Task-level cooperation-path records provide dataset/state setting, run, task identifier, DoF value, correctness, encoded branch/profile path, and path length. The canonical task-condition collection contains 33,030 unique path records. Derived comparison tables contain 1,785 same-run DoF pairs and 328 matched-routing granularity pairs.

The validation report records 32,348 path records joined to valid condition summaries, 144 records associated with null or error summaries, 442 empty cooperation paths, and 96 records without matching summaries. These categories account for all 33,030 unique path records. Error and missing states remain explicit, and no missing accuracy is imputed. The analysis runner produces the canonical accuracy table, branch-specific profile tables, granularity and routing comparisons, repeated-run comparisons, routing-control validation, and machine-readable provenance sidecars.

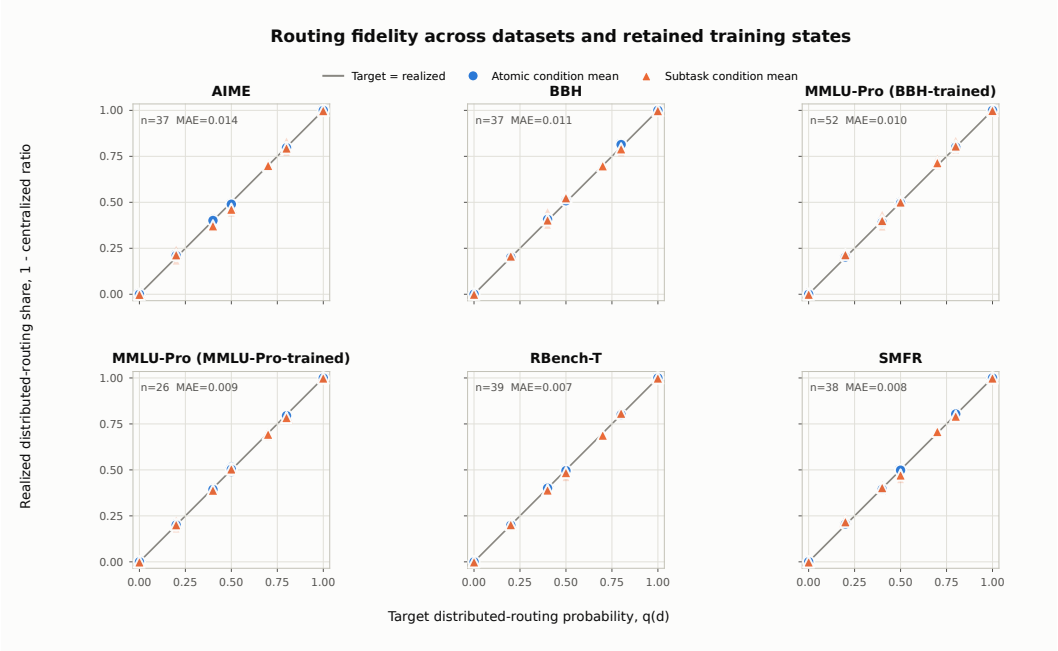

Figure 5: Target and realized decentralized task-unit assignment fractions across repeated DoF sweeps. Each panel represents one evaluation collection and retained-state setting. Small points show individual runs, large points show condition means, and the diagonal denotes exact agreement with the target $q(d)$. Marker shape distinguishes atomic-target and subtask-target conditions. Endpoint conditions are excluded because they bypass probabilistic per-unit routing.

The current analysis checkout does not contain the original per-task trace JSON files. It therefore cannot recover task text, decomposition responses, dependency graphs, model answers, review decisions, or communication content. The experiment runner defines and can write those fields, but any future release of raw traces must be inventoried separately and reviewed for benchmark redistribution terms, retained-memory content, service credentials, and user-specific paths. All claims in the present manuscript are restricted to the condition summaries and derived cooperation-path records available to the analysis pipeline.