

When Does Hierarchy Help? Benchmarking Agent Coordination in Event-Driven Industrial Scheduling

Ziqi Wang, Yuhao Yang*, Zhiwei Ling, Wenzhuo Qian, Hailiang Zhao[†]
Zhejiang University

Abstract

Recent advances in agent and multi-agent systems have shown strong performance on tool use, reasoning, and collaborative tasks. However, existing benchmarks mostly evaluate task completion in weakly coupled environments, and provide limited support for studying coordination in shared, dynamically evolving systems with hierarchy and coupled constraints. This leaves an important question underexplored: when do different coordination paradigms succeed or fail? We introduce Distributed Event-driven Scheduling Benchmark (DESBench), a benchmark for evaluating agent coordination in hierarchical event-driven scheduling. Built on a shared discrete-event driven environment in industrial scheduling, our benchmark captures multi-timescale decision making, partial observability, and dynamically coupled constraints. We define tasks and metrics that evaluate effectiveness, constraint alignment, coordination efficiency, and robustness, and focus on four representative coordination paradigms: centralized, hierarchical, heterarchical, and holonic. These paradigms correspond to distinct mechanisms of information flow, decision authority, and conflict resolution. Our controlled evaluations reveal clear coordination trade-offs: centralized coordination is robust and communication-efficient but scales poorly with difficulty; hierarchical coordination improves efficiency through decomposition but suffers from cross-level misalignment; heterarchical coordination is flexible but communication-heavy; and holonic coordination satisfies constraints well but loses global robustness. These findings demonstrate that coordination design fundamentally shapes agent system behavior in complex environments, revealing structural trade-offs that cannot be captured by outcome metrics alone and underscoring the imperative for more adaptive, principled, and dynamic coordination mechanisms in future MAS research. DESBench is available at: <https://github.com/Mr-lander/DESBenchmark>.

1 Introduction

The rapid progress of agent and multi-agent systems (MAS) has expanded their use beyond single-step reasoning and tool invocation toward more complex forms of coordination and collaborative problem solving [1]. A growing body of work explores how multiple agents can coordinate to solve complex problems, often leveraging large language models (LLM) or learned policies [2]. However, despite these advances, current evaluation benchmarks remain largely focused on task completion in weakly coupled environments, where interactions are short-horizon, state transitions are loosely coupled, and coordination plays a limited role in shaping long-term system behavior. In contrast, real-world decision-making often takes place in shared, dynamically changing environments where interaction through a common system state yields delayed and widespread effects [3, 4]. Industrial scheduling problems provide an excellent testing ground for these scenarios, where coordination is not merely a

*Project lead.

[†]Corresponding author. Email: hliangzhao@zju.edu.cn

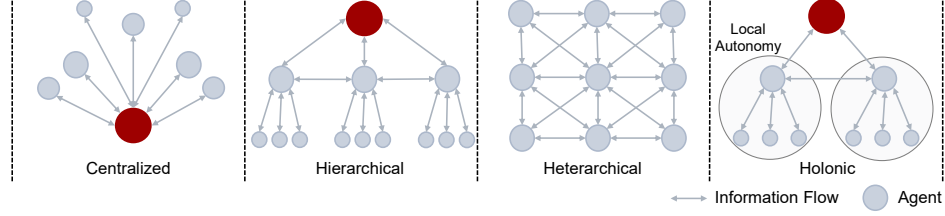


Figure 1: Four typical coordination paradigms: centralized, hierarchical, heterarchical, and holonic. Each represents a distinct mechanism of information flow, decision authority, and conflict resolution.

mechanism for dividing work, but a central factor that determines system performance, constraint satisfaction, and robustness. Existing benchmarks are insufficient for studying coordination for three key reasons:

- Traditional scheduling and optimization benchmarks primarily evaluate solution quality under implicit centralized or fixed control assumptions, providing limited insight into how alternative coordination structures influence system behavior.
- Many existing agent benchmarks lack a fully shared and dynamically evolving environment, and therefore only partially capture coordination settings where actions influence future dynamics through resource coupling and delayed feedback.
- Current environments rarely integrate hierarchical organization with dynamically coupled constraints in a unified setting, making it difficult to systematically analyze how different coordination paradigms operate under multi-level decision processes.

As a result, a fundamental question remains underexplored: *When do different coordination paradigms succeed or fail in complex and dynamic environments?* In the spirit of Aristotle’s insight that *the whole is greater than the sum of its parts*, coordination mechanisms can give rise to emergent collective behavior that cannot be inferred from isolated component performance alone.

To address this gap, we introduce Distributed Event-driven Scheduling Benchmark (DESBench), a benchmark for evaluating agent coordination in hierarchical, event-driven industrial scheduling, using it as a structured testbed for studying coordination in dynamic MAS. Our benchmark is built on a shared discrete-event simulation (DES) environment, where agents operate under partial observability, multi-timescale decision processes, and dynamically coupled constraints such as resource budgets and system capacity. Unlike prior environments [5, 6], agent decisions directly influence the evolution of the underlying system, affecting future availability, resource usage, and constraint pressure.

We focus on four representative coordination paradigms: centralized, hierarchical, heterarchical, and holonic, each corresponding to distinct mechanisms of information flow, decision authority, and conflict resolution (Figure 1). These paradigms are instantiated as executable agent systems using orchestration frameworks, enabling systematic comparison within a unified environment. We design evaluation indices that measure multiple dimensions of agent performance, including effectiveness, constraint alignment, coordination efficiency, and robustness under disruptions.

Our empirical results show that each paradigm exposes a distinct bottleneck: centralized coordination is robust but difficulty-sensitive, hierarchical coordination improves scheduling efficiency but risks level misalignment, heterarchical coordination gains adaptability at the price of heavy interaction, and holonic coordination preserves constraints but weakens global progress. These observations underscore that coordination design fundamentally governs system behavior in complex environments and point toward the need for adaptive and context-aware coordination mechanisms in future research. Our work presents three key innovations:

1. We introduce DESBench for evaluating agent coordination in event-driven industrial scheduling, capturing multi-timescale decision making and dynamically coupled constraints in shared and evolving environments.
2. We implement four representative coordination paradigms, including centralized, hierarchical, heterarchical, and holonic, each characterized by distinct mechanisms of information

flow, decision authority, and conflict resolution. We also provide a scaffold that enables the design and evaluation of new coordination strategies within our benchmark setting.

3. We establish a unified evaluation framework with four metric dimensions, including effectiveness, constraint alignment, coordination efficiency, and robustness, and use it to reveal key trade-offs across coordination paradigms, providing insights for designing more adaptive and scalable coordination mechanisms in future MAS.

2 Related Work

Agent Coordination Mechanisms. Agent coordination is fundamental in MAS, enabling distributed decision makers to share information and resolve conflicts toward collective goals [7]. Centralized coordination assigns a global controller to aggregate system-wide information and compute joint decisions. For example, EvoAgent [8] employs a centralized puppeteer to dynamically schedule and reprioritize agent actions during collaboration. While such designs can optimize global objectives under full observability, they scale poorly under uncertainty and partial observability [9, 10]. To address these limitations, recent work explores structured and decentralized coordination paradigms that balance global coherence with local autonomy [11]. Hierarchical coordination decomposes tasks into layered subproblems, enabling modular decision making and controlled cross-level interaction [12–14]. For instance, SeqComm [11] combines inter-agent negotiation with top-down execution ordering across levels. Recent decentralized algorithms advance adaptive coordination through evolving interaction topologies and interaction-based reward design, as exemplified by AgentNet and CoMAS [15, 16]. Extensions incorporate adaptive clustering [17] and privacy-aware knowledge sharing [18] to improve scalability while maintaining global consistency. Heterarchical coordination relies on peer-to-peer interaction and decentralized negotiation to reach consensus under local information constraints [19], enhancing robustness in large-scale systems. Hybrid holonic coordination [20] integrates hierarchical structure with decentralized autonomy, treating agents as self-contained units within cooperative collectives to enable dynamic reconfiguration and multi-level interaction.

Benchmarking Coordination in MAS. Existing MAS benchmarks predominantly emphasize performance on predefined scenarios and aggregate outcome measures, offering limited support for analyzing how coordination structures shape collective behavior. MultiAgentBench [5] extends evaluation to collaboration and competition through milestone-based KPIs, while BattleAgentBench [6] provides fine-grained assessment of cooperative and competitive capabilities. However, these benchmarks remain largely outcome-centric, focusing on task success or milestone achievement, and provide limited insight into coordination dynamics, communication efficiency, or sensitivity to environmental variation. Similarly, AgentsNet [21] evaluates reasoning and topology use in MAS but does not explicitly isolate the impact of coordination mechanisms on system behavior. More generally, existing evaluation frameworks rely on final-output metrics, overlooking process-level signals such as interaction patterns or redundant collaboration paths [22], despite evidence that similar outcome scores can mask substantial differences in internal coordination efficiency. These limitations align with recent surveys highlighting the lack of standardized protocols for evaluating dynamic coordination and emergent behavior in MAS [23, 7].

3 Distributed Event-driven Scheduling Benchmark (DESBench)

3.1 Problem Setting

DESBench studies industrial scheduling in a shared discrete-event environment, coordinating the execution of concurrent jobs. We consider a hierarchical MAS organized as a rooted coordination graph $\mathcal{H} = (\mathcal{V}, \mathcal{E})$, where the vertex set $\mathcal{V}$ is partitioned into three decision levels, $\mathcal{V} = \mathcal{V}^P \cup \mathcal{V}^A \cup \mathcal{V}^C$, corresponding to the *Plant*, *Area*, and *Cell* layers in a factory, respectively. A task instance specifies a fixed hierarchy $\mathcal{H}$ together with a shared discrete-event environment in which all agents interact through the evolving system state. Figure 2 depicts the DESBench architecture, where a shared event-driven environment core drives state evolution and triggers agent activations under a specified coordination protocol. At each event epoch, activated agents observe local state and protocol context, submit decisions via runtime interfaces.

At time t , the environment is in state $x_t \in \mathcal{X}$, which summarizes the current job set, machine status, queue contents, setup configuration, transport status, buffer occupancy, and resource-accounting

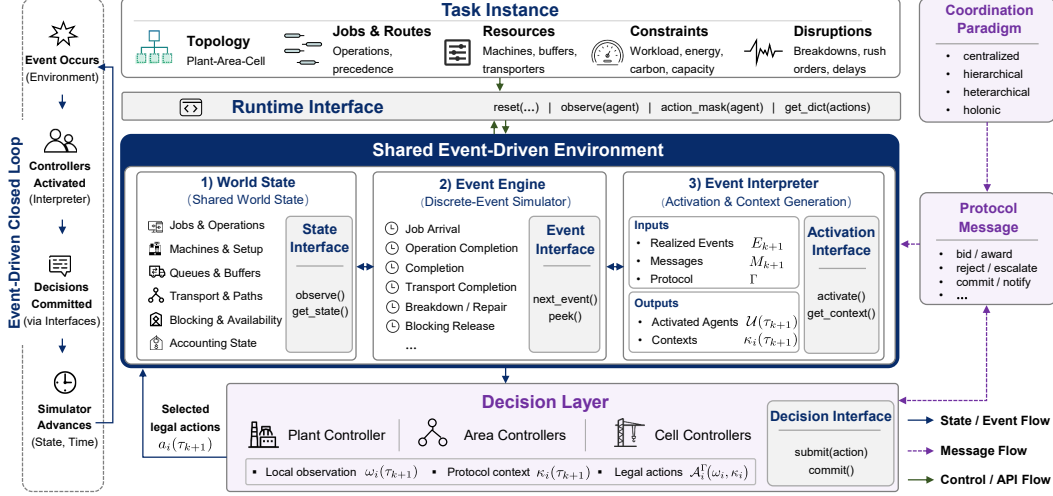


Figure 2: Overview of the DESBench. Agents at Plant, Area, and Cell levels receive local observations and protocol contexts, submit decisions via the interface, and advance the event engine. This unified architecture supports evaluation of multiple coordination paradigms under identical dynamics.

variables. State transitions are not driven by synchronous time steps but by a stream of realized environment events. We define $E_k \subseteq \mathcal{E}_{\text{env}}$ as the event set realized at time τ_k , including job arrivals, operation completions, machine breakdowns, repairs, setup changes, transport completions, and blocking releases. These event sets induce a sequence of decision epochs $0 < \tau_1 < \tau_2 < \dots < \tau_T$, at which one or more agents are activated to make scheduling or coordination decisions. Between two consecutive decision epochs, the system evolves autonomously according to the underlying discrete-event dynamics.

A task contains a set of jobs $\mathcal{J}$, where each job $j \in \mathcal{J}$ is associated with an ordered processing route $r(j) = (o_{j1}, o_{j2}, \dots, o_{jm_j})$, release conditions, and optionally due-date information. Each operation must be assigned to a feasible cell and then dispatched on a local machine subject to routing constraints, setup-dependent processing effects, transport delays across cells, finite inbound admission, and downstream blocking. Therefore, the scheduling process is jointly shaped by both local execution choices and higher-level coordination decisions.

At a decision epoch τ_k , the realized event set E_k is first incorporated into the event engine (see Section 3.2), yielding a pre-decision state $x_{\tau_k}^-$. One or more activated agents then issue coordination decisions, producing a joint decision $\mathbf{a}(\tau_k)$. Depending on the activated level, these decisions may correspond to backlog selection, area assignment, cell selection, local dispatch, rejection, rerouting, or escalation. After the activated agents commit their decisions, the event engine reaches a post-decision state $x_{\tau_k}^+$, from which the discrete-event dynamics advance to the next decision epoch. A joint execution over an episode thus yields an event-indexed trajectory

$$\xi = (x_0, E_1, \tau_1, x_{\tau_1}^-, \mathbf{a}(\tau_1), x_{\tau_1}^+, \dots, E_T, \tau_T, x_{\tau_T}^-, \mathbf{a}(\tau_T), x_{\tau_T}^+). \quad (1)$$

A key feature of DESBench is that coordination is mediated through a shared system state rather than isolated subproblems. An action taken by one agent can change the future feasibility region of other agents by altering queue congestion, setup exposure, route availability, transport load, or budget pressure. Let $c(x_t) \in \mathbb{R}^d$ denote the vector of coupled operational quantities, including workload, energy use, carbon accumulation, and capacity utilization. Feasible control must respect resource and operational constraints $c(x_t) \preceq b$, where $b \in \mathbb{R}^d$ is the task-specific budget. Violations need not occur immediately after an action, but may emerge later through delayed interactions in the event-driven dynamics, making coordination intrinsically long-horizon and cross-level. Formally, a DESBench instance is defined as

$$\mathcal{M} = (\mathcal{H}, \mathcal{X}, \mathcal{E}_{\text{env}}, \{O_i\}_{i \in \mathcal{V}}, \{A_i\}_{i \in \mathcal{V}}, P, \mathcal{J}, b, \Gamma), \quad (2)$$

where $\{O_i\}_{i \in \mathcal{V}}$ denotes the agent-specific observation operators over the shared state, $\{A_i\}_{i \in \mathcal{V}}$ denotes the corresponding base action domains, P denotes the event-driven transition law of the

shared environment, and Γ denotes the coordination protocol. In DESBench, different coordination paradigms correspond to different instantiations of Γ over the same underlying task environment.

3.2 Shared Event-Driven Environment

Given an instance $\mathcal{M}$, DESBench evolves as a shared asynchronous discrete-event system over the common state space $\mathcal{X}$. As shown in Figure 2, we distinguish three runtime components: the shared world state x_t , a shared event engine G that advances the simulator and realizes world events, and an event interpreter I that maps newly realized world objects to agent activations under Γ . All agents interact through the same underlying world state x_t , and the environment advances only when discrete events are realized. In contrast to fixed-step control benchmarks [24, 25], DESBench does not expose a universal synchronous decision loop. Instead, both the timing of decisions and the agents allowed to act are determined by the joint evolution of the physical environment and Γ .

At a decision epoch τ_k , activated agents act on the pre-decision world state $x_{\tau_k}^-$ and produce a post-decision state $x_{\tau_k}^+$. The event engine then advances the shared simulator autonomously until the next decision-relevant stopping point. Let $E_{k+1} \subseteq \mathcal{E}_{\text{env}}$ denote the set of environment events realized after decisions at epoch τ_k and incorporated at the next decision epoch τ_{k+1} , and let M_{k+1} denote the protocol-level objects generated during the same transition, including messages, commitments, and settlements. We write

$$G(x_{\tau_k}^+) = (x_{\tau_{k+1}}^-, E_{k+1}, \tau_{k+1}). \quad (3)$$

Thus, the event-engine G is represented by the autonomous mapping that rolls the simulator forward and exposes the realized world-event set E_{k+1} . Over the same interval, the coordination runtime also produces protocol-level objects M_{k+1} , including messages, commitments, and settlements.

A decision epoch is not created at every state transition. Instead, the event interpreter I consumes the newly realized E_{k+1} and M_{k+1} and determines both who acts next and what protocol context they receive. Let $\mathcal{U}(\tau_{k+1}) \subseteq \mathcal{V}$ denote the set of agents activated at epoch τ_{k+1} . We write

$$I(E_{k+1}, M_{k+1}, x_{\tau_{k+1}}^-; \Gamma) = (\mathcal{U}(\tau_{k+1}), \{\kappa_i(\tau_{k+1})\}_{i \in \mathcal{U}(\tau_{k+1})}), \quad (4)$$

where $\kappa_i(\cdot)$ denotes the protocol context. I captures the key event-driven property of the benchmark: agents do not act continuously, but only when newly realized world events or protocol objects make a new decision legally necessary.

The resulting control process is inherently asynchronous and multi-level. Different agents may be activated for different reasons, including newly opened release opportunities, local execution events, budget alarms, breakdowns, repairs, parent-to-child allocations, or escalation messages. Consequently, DESBench couples physical evolution and coordination evolution within a unified runtime: *environment events influence protocol activations, while protocol decisions in turn alter future environment events.*

At a decision epoch τ_k , the realized event set E_k has already been incorporated into the event engine, yielding the pre-decision state $x_{\tau_k}^-$. Each activated agent $i \in \mathcal{U}(\tau_k)$ receives a world-state-derived local observation $\omega_i(\tau_k) = O_i(x_{\tau_k}^-)$, together with the protocol context $\kappa_i(\tau_k)$ returned by the event interpreter. The protocol context may encode the triggering event, a parent message, local commitment status, settlement outcome, or authority-conditioned decision semantics relevant to agent i at τ_k . Agent i therefore selects an action from the protocol-conditioned feasible interface

$$a_i(\tau_k) \in \mathcal{A}_i^\Gamma(\omega_i(\tau_k), \kappa_i(\tau_k)). \quad (5)$$

Depending on the activated level and protocol instantiation, actions may correspond to backlog selection, area assignment, cell selection, local dispatch, rejection, rerouting, bid submission, or escalation. This representation makes explicit that DESBench does not expose a fixed, paradigm-independent decision menu: even under the same physical state, different instantiations of Γ can induce different feasible actions because they alter who is activated, what information is revealed, and what authority or recovery options are available at that stage.

Finally, for each task instance, $\mathcal{X}$, $\mathcal{E}_{\text{env}}$, P , $\mathcal{J}$, and b are held constant, while only the coordination protocol Γ varies. This isolates coordination as the sole experimental variable under a common dynamic substrate. Details of the shared event-driven runtime, activation logic, and state/accounting substrate are provided in Appendix A. The event trajectories are driven by a set of real-world operational datasets [26–28].

3.3 Typical Coordination Paradigms

DESBench uses the coordination protocol Γ as the primary experimental variable. For a fixed task instance, different coordination paradigms are represented as different protocol instantiations. We choose $\Gamma \in \{\Gamma^{\text{cen}}, \Gamma^{\text{hie}}, \Gamma^{\text{het}}, \Gamma^{\text{hol}}\}$ as typical examples. Each protocol Γ specifies how authority is distributed across the Plant-Area-Cell hierarchy and how control flows after commitment, rejection, or failure. These dimensions determine the legal action sets $\mathcal{A}_i^\Gamma(\omega_i(\tau_k), \kappa_i(\tau_k))$, the activation structure $\mathcal{U}(\tau_k)$, and the protocol-level objects M_k generated along a trajectory. Thus, the same local observation can lead to different feasible decisions depending on the coordination paradigm.

Centralized coordination. Under Γ^{cen} , public routing authority is concentrated at the Plant: after backlog selection, the Plant directly selects the execution Cell for the currently considered work item. Cell commitment is effectively accept-only, and rerouting is handled by Plant reselection rather than by Area-local recovery. Although the runtime still preserves the Plant-Area-Cell boundary internally, an Area commitment decision is not exposed as a separate public agent step in this mode.

Hierarchical coordination. Under Γ^{hie} , decisions follow the fixed Plant-Area-Cell chain. The Plant selects an Area, the selected Area chooses a child Cell, and the Cell commits to execution under an accept-only contract interface. If a commitment cannot be settled, the failure propagates upward rather than being resolved through local sibling rerouting. This paradigm preserves a strict command hierarchy and tests whether structured decomposition improves coordination without granting strong local autonomy.

Heterarchical coordination. Under Γ^{het} , allocation is performed through a mediated contract-net-style process. Candidate Cells may participate through a bid round and may reject contracts, while the runtime still preserves the Plant-Area-Cell object boundary. Thus, in the DESBench implementation, heterarchical coordination should be understood as a mediated heterarchical protocol rather than an unconstrained peer-to-peer system. It relaxes fixed top-down assignment by introducing competitive local participation, while retaining a bounded mediation structure for conflict resolution.

Holonic coordination. Under Γ^{hol} , the system preserves the stable Plant-Area-Cell holarchy while granting bounded local autonomy. Cells may accept or reject contracts, and Areas may attempt local rerouting among eligible sibling Cells before escalating control upward. Plant-level fallback is used only after local recovery is exhausted. This paradigm combines hierarchical organization with local self-regulation, enabling DESBench to study whether bounded autonomy can improve robustness without losing the global coordination structure.

Table 1 summarizes the high-level distinction among the four paradigms. The exact implementation-level authority boundaries and prompts are provided in Appendix B.

Table 1: High-level coordination paradigms in DESBench. All paradigms share the same event-driven scheduling environment and differ only in the coordination protocol Γ .

Paradigm	Routing authority	Local autonomy	Conflict handling
Centralized	Plant direct award	Cells execute without effective reject right	Plant performs direct re-selection
Hierarchical	Fixed Plant-Area-Cell chain	Cells are accept-only under commitment	Reject or infeasibility escalates upward
Heterarchical	Mediated bid/award process	Cells can bid and reject within mediation	Mediated settlement before fallback
Holonic	Stable holarchy with bounded autonomy	Cells can reject; Areas can locally reroute	Area recovery before Plant fallback

3.4 Tasks and Evaluation Dimensions

To systematically characterize performance beyond single objective scores, we organize the primary metrics into four evaluation dimensions: *effectiveness*, *constraint alignment*, *coordination efficiency*, and *robustness*. Effectiveness measures schedule quality over the event trajectory; constraint alignment captures how well runs respect shared operational limits; coordination efficiency quantifies interaction cost and decision overhead induced by a protocol; and robustness distinguishes among

qualitatively different failure modes under unresolved workloads or episode truncation. Table 2 summarizes the main metrics, with the full benchmark catalog provided in Appendix C.

Table 2: Main evaluation metrics in DESBench and their interpretations.

Dimension	Metric	Interpretation
Effectiveness ($\mathcal{G}_{\text{eff}}$)	G_{mk}	Mean makespan: average final simulation time at episode termination.
	G_{td}	Tardiness: episode-level total tardiness over completed jobs with due dates.
	G_{cj}	Completed jobs: average number of jobs completed within the episode.
Constraint Alignment ($\mathcal{G}_{\text{con}}$)	G_{vc}	Violation count: recorded overshoot count from violation summaries.
	G_{eo}	Energy overshoot magnitude: average cumulative amount by which energy usage exceeds the budget.
	G_{ed}	Energy overshoot duration: cumulative time spent in energy overshoot.
	G_{en}	Energy: mean total energy usage per episode.
Coordination Efficiency ($\mathcal{G}_{\text{coord}}$)	G_{ds}	Decision steps: mean number of executed rollout decision steps per episode.
	G_{cm}	Coordination messages: mean number of protocol-level coordination messages per episode.
	G_{fb}	Fallback-contract decisions: mean number of framework decisions attributed to the fallback-contract source.
	G_{llm}	LLM-driven decision count: mean number of framework decisions attributed to the LLM source.
Robustness ($\mathcal{G}_{\text{rob}}$)	G_{tr}	Truncation rate: fraction of episodes marked as truncated.
	G_{dj}	Completion debt (jobs): average number of unfinished jobs at episode end.
	G_{ut}	Unfinished overdue tardiness: mean overdue tardiness debt from unfinished jobs at episode end.
	G_{do}	Completion debt (operations): average number of remaining unprocessed operations at episode end.

Effectiveness. $\mathcal{G}_{\text{eff}}$ quantifies schedule quality. Specifically, G_{mk} records final simulation time at episode termination, G_{td} measures episode-level total tardiness over completed jobs with due dates, and G_{cj} counts completed jobs.

Constraint Alignment. Metrics in $\mathcal{G}_{\text{con}}$ assess how consistently a coordination paradigm operates within shared operational limits induced by the environment. G_{vc} is derived from runtime violation summaries, with an excess-state fallback when structured summaries are unavailable. The remaining metrics quantify energy-specific breach severity and resource use, with G_{eo} measuring cumulative energy overshoot magnitude, G_{ed} measuring cumulative time spent in energy overshoot, and G_{en} reporting mean total energy usage per episode.

Coordination Efficiency. $\mathcal{G}_{\text{coord}}$ targets the cost of coordination itself under a given protocol Γ , complementing outcome quality with indicators of control activity, communication burden, and audited decision provenance. G_{ds} counts executed rollout decision steps, and G_{cm} counts protocol-level coordination messages exposed by the runtime. G_{fb} and G_{llm} are framework-audited decision-source counts, recording how often a decision is attributed to fallback and LLM, respectively.

Robustness. $\mathcal{G}_{\text{rob}}$ characterizes early termination and residual work debt at episode end. Truncation rate (G_{tr}) records episodes marked as truncated, which may arise from deadlock termination or no-progress stopping. The remaining metrics quantify unresolved workload through unfinished jobs (G_{dj}), overdue debt from unfinished jobs (G_{ut}), and remaining operations (G_{do}).

4 Experimental Result and Discussion

Experimental Setting. We evaluate four coordination paradigms, each implemented in two orchestration frameworks, LangGraph [29] and AgentScope [30], and paired with three representative LLMs: GPT-5.4, Gemini-3-Flash, and Qwen-3.5. The experiments adopt A5C12 topology, consisting of five Areas and twelve Cells, partitioned as 4-3-2-2-1. Three scenario profiles are tested, each designed to emphasize a different coordination difficulty: branch pressure (high task branching and routing decisions), strong cluster pull (tight coupling of resource usage across areas), and late task commitments (delayed task scheduling with potential resource conflicts). Each paradigm is evaluated with a total of 30 test episodes, with 10 test seeds per framework and LLM pair for each

of the three scenarios. The final results for each paradigm are averaged across the three scenario profiles. Experiments are conducted on a local Windows workstation equipped with an Intel Core i7-12700K CPU and 32 GB RAM. The evaluation does not require specialized hardware and has modest computational requirements since it focuses on coordination behavior rather than large-scale model training. Implementation details are provided in Appendix D.

Experimental Result. We begin by examining the numerical results across different LLMs and framework implementations. As shown in Table 3, coordination structure remains the dominant source of variation, while LLM and framework differences are usually secondary. The table shows that switching coordination paradigms changes system behavior more substantially.

Table 3: Main benchmark comparison matrix across coordination paradigms.

Paradigm	Framework	LLM	Evaluation Metrics														
			$\mathcal{G}_{\text{eff}}$			$\mathcal{G}_{\text{con}}$			$\mathcal{G}_{\text{coord}}$				$\mathcal{G}_{\text{rob}}$				
			$G_{\text{mk}} \downarrow$	$G_{\text{td}} \downarrow$	$G_{\text{ej}} \uparrow$	$G_{\text{vc}} \downarrow$	$G_{\text{eo}} \downarrow$	$G_{\text{ed}} \downarrow$	$G_{\text{en}} \downarrow$	$G_{\text{ds}} \downarrow$	$G_{\text{cm}} \downarrow$	$G_{\text{fb}} \downarrow$	$G_{\text{llm}} \downarrow$	$G_{\text{tr}} \downarrow$	$G_{\text{dj}} \downarrow$	$G_{\text{ut}} \downarrow$	$G_{\text{do}} \downarrow$
Γ^{cen}	LangGraph	GPT-5.4	78.2	255.6	9.8	32.4	14471.5	9.7	12245.0	118.0	118.0	0.1	117.9	0.03	0.2	1.7	0.8
		Gemini-3-Flash	75.4	247.9	10.0	32.7	11947.0	7.0	11828.9	120.0	120.0	1.1	118.9	0.00	0.0	0.0	0.0
		Qwen-3.5	74.3	247.7	9.8	31.4	12129.5	7.1	11675.8	118.3	118.3	0.0	118.3	0.03	0.2	2.0	0.7
	AgentScope	GPT-5.4	76.3	252.1	9.4	31.8	14128.4	8.9	11943.2	115.4	115.3	0.1	115.3	0.07	0.6	3.4	1.8
		Gemini-3-Flash	75.4	247.9	10.0	32.7	11946.9	7.0	11828.8	120.0	120.0	1.1	118.9	0.00	0.0	0.0	0.0
		Qwen-3.5	76.2	251.0	9.8	31.6	12869.8	8.9	11953.4	118.3	118.3	0.0	118.3	0.03	0.2	2.0	0.7
Γ^{hie}	LangGraph	GPT-5.4	66.5	156.9	9.2	57.2	15292.6	36.0	10669.2	190.7	115.1	0.2	191.5	0.13	0.8	7.9	2.1
		Gemini-3-Flash	62.0	143.9	9.7	54.7	12461.3	29.9	10018.2	196.4	118.4	0.5	196.8	0.07	0.3	4.0	0.8
		Qwen-3.5	58.2	122.9	7.7	44.9	12318.6	28.6	9344.7	172.5	104.5	0.0	174.0	0.37	2.3	15.1	6.5
	AgentScope	GPT-5.4	64.3	151.8	8.9	55.2	13644.9	33.8	10317.0	186.4	112.5	0.1	187.4	0.20	1.1	12.3	3.2
		Gemini-3-Flash	62.0	143.9	9.7	54.8	12461.2	29.8	10012.2	196.4	118.4	0.6	196.8	0.07	0.3	4.0	0.8
		Qwen-3.5	57.8	124.5	7.9	46.6	12596.8	28.7	9285.6	172.7	104.6	0.0	174.2	0.33	2.1	10.3	6.4
Γ^{het}	LangGraph	GPT-5.4	64.4	154.0	10.0	57.7	10540.8	31.1	10329.6	331.1	471.2	1.8	330.3	0.00	0.0	0.0	0.0
		Gemini-3-Flash	62.5	148.8	9.8	55.7	12653.7	30.2	10099.6	201.3	244.9	0.3	202.0	0.03	0.2	1.4	0.4
		Qwen-3.5	62.7	138.3	8.6	51.8	13341.8	31.9	10070.0	195.6	243.0	0.0	197.1	0.27	1.4	13.8	3.7
	AgentScope	GPT-5.4	61.7	145.5	9.6	50.9	8974.6	27.8	9912.2	324.0	461.3	1.8	323.2	0.10	0.4	6.1	1.2
		Gemini-3-Flash	62.3	148.9	9.8	56.1	12695.0	30.1	10077.0	201.5	245.1	0.3	202.2	0.03	0.2	1.4	0.4
		Qwen-3.5	69.0	150.5	9.3	58.0	16163.4	38.4	11026.8	200.3	247.4	0.0	201.8	0.13	0.7	5.7	2.0
Γ^{hol}	LangGraph	GPT-5.4	25.9	30.7	2.8	5.3	738.8	2.8	4200.3	128.8	90.3	0.0	129.8	0.93	7.2	17.7	23.3
		Gemini-3-Flash	58.3	125.3	9.0	51.0	10660.1	26.4	9434.1	194.3	118.8	0.6	194.7	0.27	1.0	13.7	2.4
		Qwen-3.5	27.4	38.9	3.1	12.9	3092.7	7.5	4417.0	101.0	63.8	0.0	102.3	0.83	6.9	8.9	23.4
	AgentScope	GPT-5.4	28.7	38.1	3.4	5.2	407.9	1.8	4672.5	141.6	97.9	0.0	142.7	1.00	6.6	30.4	20.8
		Gemini-3-Flash	58.3	125.3	9.0	51.0	10660.1	26.4	9434.1	194.3	118.8	0.6	194.7	0.27	1.0	13.7	2.4
		Qwen-3.5	30.9	47.4	3.8	16.2	4400.8	9.8	4991.1	110.8	70.0	0.0	112.2	0.77	6.2	6.0	21.3

Centralized coordination excels in simple, well-structured environments. Due to central decision-making, it maintains strong robustness in this benchmark setting, as reflected by its favorable performance in the robustness metrics ($\mathcal{G}_{\text{rob}}$). Additionally, its message passing is relatively sparse, as evidenced by the lower values of G_{ds} and G_{cm} . However, its task processing efficiency is relatively low, as indicated by higher values of G_{mk} and G_{td} . Moreover, as the complexity of the environment increases, the performance of centralized coordination degrades significantly (see Appendix E).

Hierarchical coordination simplifies problems by decomposing tasks and assigning them to different levels of hierarchy. Its hierarchical strategy ensures stable robustness (a solid performance in $\mathcal{G}_{\text{rob}}$), though higher violations are observed in certain scenarios due to potential delays or misalignments between different hierarchical levels, which is a key issue of this paradigm. The improved task decomposition significantly enhances efficiency ($\mathcal{G}_{\text{eff}}$) compared to centralized coordination. Coordination costs ($\mathcal{G}_{\text{coord}}$) in hierarchical coordination are higher than in centralized coordination but lower than in heterarchical coordination. This makes hierarchical coordination a balanced approach for environments of moderate complexity.

Heterarchical coordination offers greater flexibility and adaptability, dynamically adjusting to different task requirements. However, its violation rate ($\mathcal{G}_{\text{con}}$) is not low, which is due to the increased complexity and the need for constant adjustments between levels. Efficiency ($\mathcal{G}_{\text{eff}}$) reaches a moderate level, and robustness is strong because the system can respond to changes and failures more effectively through its flexible structure. However, it incurs the highest coordination and message counts ($\mathcal{G}_{\text{coord}}$). For instance, the decision steps (G_{ds}) are 331.1, and the coordination message count (G_{cm}) is 471.2, reflecting an increased communication overhead compared to both centralized and hierarchical models. While heterarchical coordination provides flexibility and robustness, its high coordination cost reduces its overall efficiency. Excessive communication can lead to system confusion, hindering the effectiveness of the coordination process.

Holonic coordination emphasizes autonomy by allowing individual components to make local decisions. It achieves low delay-related effectiveness. The low makespan (G_{mk}) and tardiness (G_{td}) partly result from limited completed workload (G_{cj}) rather than uniformly superior scheduling quality. Holonic coordination attains the best constraint satisfaction. This is because autonomous local control tends to act conservatively and avoids aggressive global resource contention. Its coordination efficiency is also competitive, with moderate decision steps (G_{ds}) and coordination messages (G_{cm}), which are far lower than those of heterarchical coordination. However, holonic coordination shows the weakest robustness, as reflected by a high truncation rate (G_{tr}), large job completion debt (G_{dj}), and high remaining operation debt (G_{do}). This weakness arises from the fragmentation introduced by autonomy: local units can avoid violations, but they may fail to maintain global progress and recovery under dynamic disruptions, making the paradigm less effective in dynamic environments. It is worth noting that performance varies across different LLMs, as this paradigm relies on local autonomy, which amplifies model-specific behaviors.

Additional experiments on instances, scalability, and task difficulty are provided in Appendix E.

5 Discussion and Future Directions

The above results expose a fundamental and non-trivial trade-off landscape. Only four typical static coordination paradigms already reveal substantial behavioral diversity. No coordination paradigm dominates across all dimensions and each paradigm occupies a distinct region in the space. This highlights that coordination is not merely an implementation choice, but a first-order design variable that directly shapes system behavior.

A key insight is the inherent tension between *global consistency* and *local responsiveness*. Centralized coordination maintains strong robustness and low coordination overhead, yet suffers from scalability and efficiency limitations as system complexity grows. In contrast, heterarchical coordination achieves adaptability and resilience at the cost of excessive communication, revealing that unstructured flexibility can introduce coordination noise that degrades system-level efficiency. Hierarchical coordination partially mitigates this tension by introducing structure, but its reliance on cross-level alignment creates systematic vulnerabilities. Holonic coordination, while excelling in constraint satisfaction, exposes a critical failure mode: local autonomy can fragment global progress, leading to poor robustness despite favorable local metrics.

In addition, these results also show that single-metric comparisons are inadequate for evaluating coordination. Metrics such as makespan or tardiness can be misleading when completion and truncation are ignored, as illustrated by the holonic results in Table 3. Looking forward, several research directions emerge. First, hybrid coordination mechanisms [7, 31] that combine the strengths of multiple paradigms (e.g., hierarchical structures with adaptive heterarchical interactions) represent a promising avenue for balancing efficiency and robustness. Second, learning-based approaches that adapt coordination strategies online based on system state and workload characteristics could mitigate the rigidity observed in fixed protocols [8, 32]. Third, improving communication efficiency [33, 34], for example through selective activation or information compression, is essential to address the scalability bottlenecks observed in heterarchical systems.

Advancing MAS toward real-world complex tasks such as industrial scheduling requires rethinking coordination not as a static design, but as a dynamic and context-dependent process. A systematic understanding of these principles can help guide the development of next-generation coordination frameworks and stimulate deeper investigation into the interplay between control, communication, and system dynamics. Finally, a guidance on how to use our benchmark is provided in Appendix F.

6 Conclusion

This paper introduces DESBench, a unified benchmark for evaluating agent coordination in distributed event-driven scheduling. Our results reveal that coordination design fundamentally shapes system behavior, with clear trade-offs across effectiveness, constraint alignment, coordination efficiency, and robustness. No single paradigm dominates and each exposes distinct strengths and limitations under shared dynamics. We expect DESBench to provide a principled foundation for studying agent coordination mechanisms and to facilitate the development of more adaptive, efficient, and robust coordination strategies in multi-agent systems.

References

- [1] Shuiguang Deng, Hailiang Zhao, Ziqi Wang, Guanjie Cheng, Peng Chen, Wenzhuo Qian, Zhiwei Ling, Jianwei Yin, Albert Y Zomaya, and Schahram Dustdar. Agentic services computing. *arXiv preprint arXiv:2509.24380*, 2025.
- [2] Xinyi Li, Sai Wang, Siqi Zeng, Yu Wu, and Yi Yang. A survey on llm-based multi-agent systems: workflow, infrastructure, and challenges. *Vicinagearth*, 1(1):9, 2024.
- [3] Andrew Wagenmaker, Kevin Huang, Liyiming Ke, Kevin Jamieson, and Abhishek Gupta. Overcoming the sim-to-real gap: Leveraging simulation to learn to explore for real-world rl. *Advances in Neural Information Processing Systems*, 37:78715–78765, 2024.
- [4] Martin Rohbeck, Edward De Brouwer, Charlotte Bunne, Jan-Christian Huetter, Anne Biton, Kelvin Y Chen, Aviv Regev, and Romain Lopez. Modeling complex system dynamics with flow matching across time and conditions. In *The Thirteenth International Conference on Learning Representations*, 2025.
- [5] Kunlun Zhu, Hongyi Du, Zhaochen Hong, Xiaocheng Yang, Shuyi Guo, Daisy Zhe Wang, Zhenhailong Wang, Cheng Qian, Robert Tang, Heng Ji, et al. Multiagentbench: Evaluating the collaboration and competition of llm agents. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8580–8622, 2025.
- [6] Wei Wang, Dan Zhang, Tao Feng, Boyan Wang, and Jie Tang. Battleagentbench: A benchmark for evaluating cooperation and competition capabilities of language models in multi-agent systems, 2024. URL <https://arxiv.org/abs/2408.15971>.
- [7] Lijun Sun, Yijun Yang, Qiqi Duan, Yuhui Shi, Chao Lyu, Yu-Cheng Chang, Chin-Teng Lin, and Yang Shen. Multi-agent coordination across diverse applications: A survey. *arXiv preprint arXiv:2502.14743*, 2025.
- [8] Yufan Dang, Chen Qian, Xueheng Luo, Jingru Fan, Zihao Xie, Ruijie Shi, Weize Chen, Cheng Yang, Xiaoyin Che, Ye Tian, Xuantang Xiong, Lei Han, Zhiyuan Liu, and Maosong Sun. Multi-agent collaboration via evolving orchestration. *Advances in Neural Information Processing Systems*, 2025.
- [9] Yueheng Li, Guangming Xie, and Zongqing Lu. Multi-agent guided policy optimization. *arXiv preprint arXiv:2507.18059*, 2025.
- [10] Zhuofan Xu, Benedikt Bollig, Matthias Függer, Thomas Nowak, and Vincent Le Dréau. Centralized permutation equivariant policy for cooperative multi-agent reinforcement learning. *arXiv preprint arXiv:2508.11706*, 2025.
- [11] Ziluo Ding, Zeyuan Liu, Zhirui Fang, Kefan Su, Liwen Zhu, and Zongqing Lu. Multi-agent coordination via multi-level communication. *Advances in Neural Information Processing Systems*, 37:118513–118539, 2024.
- [12] Pei Liu, Xin Liu, Ruoyu Yao, Junming Liu, Siyuan Meng, Ding Wang, and Jun Ma. Hm-rag: Hierarchical multi-agent multimodal retrieval augmented generation. In *Proceedings of the 33rd ACM international conference on multimedia*, pages 2781–2790, 2025.
- [13] Jinzhou Tang, Jusheng Zhang, Qinhan Lv, Sidi Liu, Jing Yang, Chengpei Tang, and Keze Wang. Hiva: Self-organized hierarchical variable agent via goal-driven semantic-topological evolution. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 2083–2091, 2026.
- [14] Xianghua Zeng, Hao Peng, Dingli Su, and Angsheng Li. Hierarchical decision making based on structural information principles. *Journal of Machine Learning Research*, 26(182):1–55, 2025.
- [15] 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. *Advances in Neural Information Processing Systems*, 2025.

- [16] Xiangyuan Xue, Yifan Zhou, Guibin Zhang, Zaibin Zhang, Yijiang Li, Chen Zhang, Zhenfei Yin, Philip Torr, Wanli Ouyang, and Lei Bai. Comas: Co-evolving multi-agent systems via interaction rewards. In *The Fourteenth International Conference on Learning Representations*, 2026.
- [17] Chiqiang Liu and Dazi Li. Hygma: Hypergraph coordination networks with dynamic grouping for multi-agent reinforcement learning. *arXiv preprint arXiv:2505.07207*, 2025.
- [18] Goutham Nalagatla. Hierarchical decentralized multi-agent coordination with privacy-preserving knowledge sharing: Extending agentnet for scalable autonomous systems. *arXiv preprint arXiv:2512.00614*, 2025.
- [19] Hung Du, Srikanth Thudumu, Hy Nguyen, Rajesh Vasa, and Kon Mouzakis. Contextual knowledge sharing in multi-agent reinforcement learning with decentralized communication and coordination. *arXiv preprint arXiv:2501.15695*, 2025.
- [20] Marjan Keramati, Sauleh Etemedi, and Nasser Mozayani. Hmlb: Holonic multi-agent approach for preventive controllers load-balancing in sdn-enabled smart grid. *Computer Communications*, 228:107984, 2024.
- [21] Florian Grötschla, Luis Müller, Jan Tönshoff, Mikhail Galkin, and Bryan Perozzi. Agentsnet: Coordination and collaborative reasoning in multi-agent llms. *arXiv preprint arXiv:2507.08616*, 2025.
- [22] Jisoo Lee, Raeyoung Chang, Dongwook Kwon, Harmanpreet Singh, and Nikhil Verma. Gemmas: Graph-based evaluation metrics for multi agent systems. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 1522–1532, 2025.
- [23] Soharab Hossain Shaikh. Llm-based multi-agent systems: Frameworks, evaluation, open challenges, and research frontiers. In *International Joint Conference on Computational Intelligence*, pages 149–170. Springer, 2025.
- [24] Kunal Menda, Yi-Chun Chen, Justin Grana, James W. Bono, Brendan D. Tracey, Mykel J. Kochenderfer, and David Wolpert. Deep reinforcement learning for event-driven multi-agent decision processes. *IEEE Transactions on Intelligent Transportation Systems*, 20(4):1259–1268, 2019. doi: 10.1109/TITS.2018.2848264.
- [25] Dong Wang and Giovanni Beltrame. Moseac: Streamlined variable time step reinforcement learning. *arXiv preprint arXiv:2406.01521*, 2024.
- [26] Damien Trentesaux, Cyrille Pach, Abdelghani Bekrar, Yves Sallez, Thierry Berger, Thérèse Bonte, Paulo Leitão, and José Barbosa. Benchmarking flexible job-shop scheduling and control systems. *Control Engineering Practice*, 21(9):1204–1225, 2013. ISSN 0967-0661. doi: <https://doi.org/10.1016/j.conengprac.2013.05.004>. URL <https://www.sciencedirect.com/science/article/pii/S09670661133000889>.
- [27] Olivier Cardin and Anne L’Anton. Proposition of an implementation framework enabling benchmarking of holonic manufacturing systems. *CoRR*, abs/1901.05669, 2019. URL <http://arxiv.org/abs/1901.05669>.
- [28] Jonathan Hoss, Felix Schelling, and Noah Klarmann. A production scheduling framework for reinforcement learning under real-world constraints. In *2025 IEEE 21st International Conference on Automation Science and Engineering (CASE)*, pages 1736–1743. IEEE, 2025.
- [29] Jialin Wang and Zhihua Duan. Agent ai with langgraph: A modular framework for enhancing machine translation using large language models. *arXiv preprint arXiv:2412.03801*, 2024.
- [30] Dawei Gao, Zitao Li, Xuchen Pan, Weirui Kuang, Zhijian Ma, Bingchen Qian, Fei Wei, Wenhao Zhang, Yuexiang Xie, Daoyuan Chen, et al. Agentscope: A flexible yet robust multi-agent platform. *arXiv preprint arXiv:2402.14034*, 2024.
- [31] David J Moore. A taxonomy of hierarchical multi-agent systems: Design patterns, coordination mechanisms, and industrial applications. *arXiv preprint arXiv:2508.12683*, 2025.

- [32] Limengxi Yue, Duo Xu, Dong Qiu, Yanpei Shi, Shuyang Xu, and Manish Shah. Sequential cooperative multi-agent online learning and adaptive coordination control in dynamic and uncertain environments. In *2025 5th International Conference on Electronic Information Engineering and Computer Communication (EIECC)*, pages 692–697. IEEE, 2025.
- [33] Dae Cheol Kwon and Xinyu Zhang. Cp-agentnet: Autonomous and explainable communication protocol design using generative agents. In *2025 IEEE 33rd International Conference on Network Protocols (ICNP)*, pages 1–12. IEEE, 2025.
- [34] Hui Yi Leong, Yuheng Li, Yuqing Wu, Wenwen Ouyang, Wei Zhu, Jiechao Gao, and Wei Han. Amas: Adaptively determining communication topology for llm-based multi-agent system. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 2061–2070, 2025.

A Additional Environment and Runtime Details

One decision epoch unfolds in four stages:

1. Starting from the post-decision state $x_{\tau_k}^+$, the event engine advances autonomously until the next decision-relevant stopping point is reached, yielding a new pre-decision state $x_{\tau_{k+1}}^-$ together with realized world events E_{k+1} .
2. The event interpreter combines these world events with any newly materialized protocol-level objects M_{k+1} and computes the next activated set $\mathcal{U}(\tau_{k+1})$.
3. For each activated agent $i \in \mathcal{U}(\tau_{k+1})$, the runtime constructs the benchmark-visible decision boundary (local observation), a protocol context, and a protocol-conditioned legal action set.
4. Returned actions are validated, committed to the event engine, and written back as updated state and protocol artifacts.

A.1 Event Engine

The event engine is the shared discrete-event scheduling substrate used by all coordination protocols. It advances physical production time, realizes environment events, updates job, machine, buffer, transport, blocking, and accounting states, and exposes the next decision-relevant stopping point. All coordination paradigms share this same substrate: the event queue, job/machine/buffer/transport/blocking semantics, setup and failure processes, and resource-accounting logic are held fixed. Different paradigms alter only the protocol layer Γ , which changes agent activation, information exposure, authority boundaries, and legal actions. The substrate follows the production-scheduling and discrete-event simulation in which job release, operation completion, setup, transport, buffer admission, blocking, and disruptions are represented as event-driven state changes [26].

DES Bench is therefore not exposed to controllers (agents) as a fully observed fixed-step Markov decision process. The full internal simulator state includes the pending event queue, machine status, job progress, ready jobs, blocked backlog, buffer occupancy, transport timers, residual processing and setup times, protocol artifacts, random generator state, and accumulated accounting variables. This complete state is internal to the engine. At benchmark level, agents are activated only at irregular event epochs and receive local observations, protocol contexts, and protocol-conditioned legal action sets.

In this case, Equation 3 is implemented by the event engine’s autonomous advancement routine. G is the shared rollout operator that advances the physical DES from the validated post-decision state $x_{\tau_k}^+$ to the next exposed pre-decision state while accumulating the realized environment-event set E_{k+1} and the corresponding timestamp τ_{k+1} .

Autonomous event-advance loop. Starting from $x_{\tau_k}^+$, the engine (i) selects the earliest pending environment event from the deterministic queue; (ii) advances the simulator clock to that event time; (iii) applies the event-specific physical transition, which can be written as $x \leftarrow \Phi_e(x)$; (iv) updates the affected machine, job, buffer, transport, blocking, and accounting variables and schedules any induced follow-on events; (v) appends the realized event via $E_{k+1} \leftarrow E_{k+1} \cup \{e\}$; and (vi) continues

until a decision-relevant stopping point is reached. The stopping point is not simply the next queue pop, but the first runtime boundary at which a new agent intervention is legally required or the episode terminates. Thus, a physical event is not identical to a decision epoch: several world events may be processed before any agent is activated.

The engine maintains a deterministic event queue. Events are primarily ordered by event time, with deterministic tie-breaking for events realized at the same timestamp. The event alphabet shared across paradigms includes both basic production events and inter-cell transport/buffering events. The base simulator handles the following events:

- arrival, setup_start, setup_complete, finish, breakdown, and repair.

The inter-cell scheduling layer extends this alphabet with the following events

- shared_backlog_arrival, transport_start, transport_complete, stage_release_ready, buffer_wait_start, blocking_start, buffer_admit, and blocking_end.

These labels describe realized world events in the shared substrate. They are distinct from protocol-level objects such as messages, commitments, settlements, rejection records, or escalation artifacts generated by the coordination layer. These event classes mirror common real-world scheduling constraints such as transport logistics, buffer management, sequence/setup effects, machine breakdowns, and stochastic processing conditions [28].

The main scheduling transitions follow standard discrete-event shop-floor semantics. A job-arrival event inserts work into the ready set when it is eligible and not already processing or completed. A dispatch decision binds a ready job to a feasible idle machine. Once committed, dispatch proceeds as `ready job → feasible idle machine → setup if needed → processing → finish`.

If the selected machine requires a setup change, the engine first realizes setup-related events before processing begins; otherwise, processing starts immediately and a finish event is scheduled. A finish event frees the machine, advances the job’s operation index, and then either marks the job as completed or releases it to the next route stage. Completion therefore updates both physical availability and accounting state, while nonterminal completion creates the next routing, transport, or stage-release opportunity.

The inter-cell layer introduces transport, finite inbound admission, and downstream blocking. A routing or assignment decision does not immediately make a job available for processing. Instead, the job enters a transport process governed by transport-related events. Only upon transport completion does it enter the destination cell’s inbound state and become eligible for local setup and dispatch. Similarly, a `stage_release_ready` event indicates that a downstream release has become physically admissible, but not yet processed. When a downstream inbound buffer is full, the engine neither drops the job nor treats the assignment as completed. Instead, it records buffer waiting or blocking and moves the item into a blocked backlog. Once downstream capacity becomes available, admission or blocking-release events are triggered, allowing the job to re-enter the appropriate backlog or inbound state and resume normal routing and dispatch.

Failure and repair are also treated as world events. A breakdown changes the availability of the affected machine and prevents the corresponding processing slice from completing normally under the simulator’s failure semantics. The machine remains unavailable until the matching repair event is realized. After repair, the engine either resumes eligible interrupted work or returns the machine to the idle state. This is why routing decisions in DESBench do not immediately convert into processing availability: transport, finite inbound capacity, and failure recovery are modeled as separate physical events. During these transitions, the engine records the accounting and trace substrate used by the benchmark metrics. It accumulates simulation time, energy usage, carbon emissions, setup time, transport time, blocking time, buffer waiting time, completed-job counts, remaining operations, and episode-status indicators such as truncation, deadlock, and done flags. These traces make the benchmark auditable: the reported metrics are derived from the same shared event trajectory.

A.2 Event Interpreter

Once the event engine has produced the next pre-decision state, the runtime switches from physical advancement to activation logic. The interpreter takes as input the realized world-event set E_{k+1} ,

the newly materialized protocol-level object set M_{k+1} , the pre-decision state $x_{\tau_{k+1}}^-$, and the current coordination protocol Γ , and maps them to the next activated set and controller-facing protocol contexts. As described in Equation 4 in Section 3.2, DESBench distinguishes between *environment events* and *protocol-level coordination objects*. The former are generated by the shared event engine and capture changes in the underlying operational system, while the latter arise from the execution of the coordination protocol itself. We write

$$E_{k+1} \subseteq \mathcal{E}_{\text{env}}, \quad M_{k+1} \subseteq \mathcal{M}_{\text{proto}}, \quad (6)$$

where $\mathcal{M}_{\text{proto}}$ denotes the space of materialized protocol-level objects, including messages, commitments, allocations, settlements, rejection records, and escalation artifacts.

Operationally, I scans the realized event set E_{k+1} and protocol objects M_{k+1} for decision triggers, and resolves them under the authority, rerouting, escalation, and commitment semantics of Γ . The output is not merely a set of agent identifiers: $\mathcal{U}(\tau_{k+1})$ specifies which agents are obligated to act, while each $\kappa_i(\tau_{k+1})$ provides a role-conditioned context constructed from triggering events, relevant protocol objects, and necessary local state summaries. The interpreter maps these world events and protocol-level objects to activation records. We denote the activation record for agent i at epoch τ_k by $\alpha_i(\tau_k)$ and write it as

$$\alpha_i(\tau_k) = (i, \text{cause}_i(\tau_k), \text{loop}_i(\tau_k), \text{time}_i(\tau_k), \text{ctx}_i(\tau_k)). \quad (7)$$

where $\text{cause}_i(\tau_k)$ records whether the activation is induced by world events or protocol-level objects, $\text{loop}_i(\tau_k)$ identifies the current functional loop of the runtime, and $\text{ctx}_i(\tau_k)$ stores the structured context attached to that activation. This activation record is an implementation-level representation used to build $\kappa_i(\tau_k)$. In the current runtime, activations arise primarily from two sources:

1. **World-event activations**, induced by newly realized environment events such as release opportunities, completion-triggered routing, queue-state changes, budget alarms, breakdowns, or repairs.
2. **Protocol-object activations**, induced by coordination objects addressed to a specific agent, such as assignment requests, parent-to-child allocations, child feedback, settlement notices, rejection records, or escalation requests.

Only agents in $\mathcal{U}(\tau_k)$ are required to act at epoch τ_k , and the activated set itself is an output of the event interpreter.

Controller-facing payload. Conditional on activation, agent i does not observe the full hidden simulator state. The exposed decision boundary consists of $(\omega_i(\tau_k), \kappa_i(\tau_k), m_i^\Gamma(\cdot \mid \omega_i(\tau_k), \kappa_i(\tau_k)))$, where $\omega_i(\tau_k) = O_i(x_{\tau_k}^-)$ is the local observation derived from the pre-decision world state, $\kappa_i(\tau_k)$ is the protocol context returned by the interpreter, and m_i^Γ is the legality mask induced by the coordination protocol. The feasible decision set can be written as

$$\mathcal{A}_i^\Gamma(\omega_i(\tau_k), \kappa_i(\tau_k)) = \{a \in \mathcal{A}_i : m_i^\Gamma(a \mid \omega_i(\tau_k), \kappa_i(\tau_k)) = 1\}. \quad (8)$$

Therefore, the same physical state can induce different feasible actions under different Γ because the protocol changes who is activated, what information is exposed, and which authority or recovery options are legally available.

Hidden state, validation, and traces. The hidden simulator state remains internal to the event engine and interpreter. Agents never read the full queueing, machine, buffer, transport, blocking, or accounting state directly. These variables influence decisions only through the derived local observation ω_i , protocol context κ_i , and legal action mask. Returned actions are validated against $\mathcal{A}_i^\Gamma(\omega_i(\tau_k), \kappa_i(\tau_k))$ and then committed back to the event engine, which resumes autonomous advancement from the resulting post-decision state. The accounting traces accumulated by the engine support the metric families in Appendix C.

B Coordination Protocol Details

This section specifies the implementation-level protocol boundaries corresponding to the four coordination paradigms in Section 3.3. In DESBench, each paradigm is implemented as a concrete

authority mode that instantiates the coordination protocol Γ . These authority modes only change how routing authority, local autonomy, rerouting, escalation, and mediated bidding are distributed across the Plant-Area-Cell hierarchy. Let $\rho(\Gamma) = (q_\Gamma, u_\Gamma, p_\Gamma, h_\Gamma)$ denote the implementation-level authority tuple associated with protocol Γ , where

- q_Γ denotes the cell-level contract action set;
- u_Γ indicates whether Area-local rerouting is enabled;
- p_Γ indicates whether a failed lower-level commitment is escalated to a mandatory Plant-level reroute path;
- h_Γ indicates whether a mediated bid/award round is enabled.

The four DESBench coordination paradigms are shown in Table 4.

Centralized. It is implemented by authority mode `centralized`. The Plant acts as the direct routing authority and selects the execution cell without requiring an Area-level commitment stage. Cells expose an accept-only contract boundary, so rejection is not part of the effective cell-level action interface. Area-local rerouting and mediated bid rounds are disabled. Because routing authority remains at the Plant, rerouting is handled as direct Plant reselection rather than as upward escalation from a failed local decision.

Hierarchical. It is implemented by authority mode `hierarchical`. Routing follows the fixed Plant-Area-Cell chain. The Plant selects an Area, the Area selects a child Cell, and the Cell accepts the assigned commitment under an accept-only contract interface. Area-local rerouting is disabled. If a commitment cannot be settled, the failure propagates upward, and Plant-level reroute is required. This creates a strict layered control flow in which recovery is handled by escalation rather than local sibling substitution.

Heterarchical. It is implemented by authority mode `heterarchical_cnp`. Allocation is mediated through a contract-net-style bid/award mechanism. Cells may participate in a bid round and expose an accept/reject contract boundary. Area-local rerouting is disabled, because conflict resolution is represented through mediated competition.

Holonic. It is implemented by authority mode `holonic_hybrid`. The benchmark preserves the stable Plant-Area-Cell holarchy while granting bounded autonomy to lower levels. Cells expose an accept/reject contract boundary. When a Cell rejects a contract or local infeasibility occurs, the Area may first attempt local rerouting among eligible sibling Cells. Plant fallback is used only after local recovery is exhausted. Mediated bid rounds are disabled, so the protocol emphasizes local self-regulation within a stable holarchy.

Table 4: Implementation-level protocol switches for the four coordination paradigms in DESBench.

Paradigm	mode_id	q_Γ	u_Γ	p_Γ	h_Γ
Centralized	<code>centralized</code>	<code>accept_only</code>	<code>False</code>	<code>False</code>	<code>False</code>
Hierarchical	<code>hierarchical</code>	<code>accept_only</code>	<code>False</code>	<code>True</code>	<code>False</code>
Heterarchical	<code>heterarchical_cnp</code>	<code>accept_reject</code>	<code>False</code>	<code>False</code>	<code>True</code>
Holonic	<code>holonic_hybrid</code>	<code>accept_reject</code>	<code>True</code>	<code>False</code>	<code>False</code>

Action-boundary effects. The authority tuple $\rho(\Gamma)$ affects the benchmark interface through the protocol-conditioned legal action set $\mathcal{A}_i^\Gamma(\omega_i(\tau_k), \kappa_i(\tau_k))$. For example, when $q_\Gamma = \text{accept_only}$, a Cell activation under commitment management exposes only acceptance-compatible contract behavior. When $q_\Gamma = \text{accept_reject}$, rejection becomes a legal Cell action and may generate a rejection object in the next materialized protocol set M_{k+1} . Similarly, when $u_\Gamma = \text{True}$, an Area receiving a child rejection may attempt a local reroute before escalation; when $u_\Gamma = \text{False}$, the same rejection must be settled through the configured escalation or fallback path. When $h_\Gamma = \text{True}$, the Area-level commitment context includes mediated bid/award semantics, enabling candidate Cells to participate competitively before an allocation is committed.

B.1 Prompt and Single-Decision Context

All framework adapters use the same OpenAI-compatible two-message interaction, so prompt wording is controlled across frameworks and model families. The system instruction only enforces

the benchmark contract: the controller must use the current agent-local payload and return exactly one legal action index. The same prompt contract is used for all test cases, and only the runtime payload values change with the current state, activated role, legal actions, and visible context.

System instruction.

You are a benchmark controller agent. CRITICAL OUTPUT CONTRACT. Select exactly one legal integer action index from legal_actions. Use only the provided compact agent-local payload, including current context, visible feedback, decision_factors, and bounded working_memory when present. Reply with exactly one JSON object and nothing else. Valid schema: {"action": <index>}. No reasoning. No markdown. No prose.

Runtime payload construction. At each activation, the benchmark first constructs a role-conditioned controller payload from the activated agent’s local observation $\omega_i(\tau_k)$ and protocol context $\kappa_i(\tau_k)$. The adapter then compacts this payload into a JSON object with the following recurring fields:

- agent, role, and kind: identify the activated controller and decision type;
- legal_actions: the only action indices that can be returned;
- decision_factors: short benchmark-provided guidance for the current decision kind;
- options: compact summaries of the currently legal candidates;
- context: visible feedback, recent event types, selected job/stage information, and bounded working memory when available.

Representative single-decision payload from A5C12.

```
{
  "agent": "plant",
  "role": "plant",
  "kind": "area_selection",
  "legal_actions": [0, 1, 2],
  "decision_factors": [
    "Use current context, visible feedback, and working_memory only.",
    "Prefer feasible actions that reduce congestion and future reroute pressure."
  ],
  "options": [
    {"action": 0, "area_id": 0,
     "summary": {"active_jobs": 0, "backlog_jobs": 10, "energy_kwh": 0.0}},
    {"action": 1, "area_id": 1,
     "summary": {"active_jobs": 0, "backlog_jobs": 10, "energy_kwh": 0.0}},
    {"action": 2, "area_id": 2,
     "summary": {"active_jobs": 0, "backlog_jobs": 10, "energy_kwh": 0.0}}
  ],
  "context": {
    "feedback": {
      "progress": {
        "released_jobs_total": 0,
        "completed_jobs_total": 0,
        "backlog_jobs_total": 10
      },
      "capacity": {
        "admissible_for_new_work": true,
        "busy_machines_total": 0,
        "down_machines_total": 0
      }
    },
    "recent_event_types": ["release_area_selection_window"],
    "selected_job_id": 0,
    "selected_stage_id": 0
  }
}
```

Generation uses deterministic decoding with a short output budget, as the benchmark evaluates coordination decisions rather than free-form reasoning. A response is accepted only if it yields a legal integer action from `legal_actions`. Otherwise, the runtime discards it, falls back to the contract-safe controller path, and records the decision source for auditing.

C Full Metric Catalog

DESBench records a broad set of metrics to support both outcome evaluation and deeper analysis of coordination behavior. In Section 3.4, we report a minimal reading set chosen for interpretability. This section provides the complete metric catalog.

We group metrics into five categories: *effectiveness*, *constraint alignment*, *coordination efficiency*, *robustness*, and *process-oriented metrics*. The first four correspond to the primary evaluation dimensions used in the main text, capturing schedule quality, adherence to shared operational limits, coordination cost and interaction burden, and resilience under unresolved workload or execution pressure, respectively. The fifth category is reserved for mechanism analysis, trace inspection, and implementation diagnostics, enabling finer-grained investigation of protocol behavior.

Effectiveness. Table 5 describes service quality and task completion outcomes.

Table 5: Effectiveness metrics in DESBench.

Metric	Meaning	Note
G_{sr}	Success rate: fraction of episodes that terminate successfully.	Strict evaluator success definition: terminated and neither deadlocked nor truncated.
G_{mk}	Makespan mean: mean episode completion time.	Measures realized simulation horizon and is a core effectiveness indicator.
G_{mk}^σ	Makespan std: standard deviation of episode makespan.	Variability / stability measure across rollouts.
G_{td}	Tardiness: mean total tardiness over completed jobs with due dates.	Counts only completed jobs.
G_{tcd}	Tardiness with completion debt: mean tardiness of completed jobs plus overdue debt from unfinished jobs.	More suitable than G_{td} when truncation is non-negligible.
G_{cj}	Completed jobs: mean number of completed jobs per episode.	Raw completion volume.
G_{th}	Throughput: mean value of completed jobs per unit makespan.	Approximate jobs completed per simulation unit time.

Constraint Alignment. Table 6 quantifies whether the agent completes work while respecting shared resource and budget constraints.

Coordination Efficiency. Table 7 quantifies coordination burden, branching complexity, and decision cost.

Robustness. Table 8 describes whether an episode converges stably, completely, and without residual work debt.

Process-oriented Metrics. These metrics are not used as the four primary main-text evaluation dimensions. They are more appropriate for trace interpretation, mechanism explanation, and implementation diagnostics. Process-oriented supplementary metrics are shown in Table 9. Diagnostic and compatibility metrics are shown in Table 10.

D Implementation Approach

This section explains the implementation boundary by which DESBench is instantiated in concrete orchestration frameworks in Section 4.

Shared execution contract. At each decision epoch, the event interpreter identifies the activated set $\mathcal{U}(\tau_k)$ and constructs one role-conditioned local controller payload for each activated agent. The framework layer receives only these benchmark-visible payloads and must return one legal action

Table 6: Constraint-alignment metrics in DESBench.

Metric	Meaning	Note
G_{en}	Energy: total energy consumption over an episode, in kWh.	Computed from final cell summaries; core constraint usage indicator.
G_{co}	Carbon: mean total carbon emissions, in kg CO ₂ .	Computed from final cell summaries; reflects sustainability burden.
G_{vc}	Violation count: mean number of recorded overshoot events.	On current structured surfaces, it largely mirrors the frequency of energy and carbon overshoot intervals.
G_{eo}	Energy overshoot magnitude: mean cumulative amount by which energy usage exceeds the cap.	Stronger than pure count metrics, captures overshoot severity.
G_{com}	Carbon overshoot magnitude: mean cumulative carbon overshoot above the cap.	Analogous to energy overshoot but for carbon.
G_{ed}	Energy overshoot duration: mean cumulative time spent above the energy cap.	Time-above-cap metric for energy violations.
G_{cod}	Carbon overshoot duration: mean cumulative time spent above the carbon cap.	Time-above-cap metric for carbon violations.
G_{eof}	Energy overshoot frequency: mean number of distinct energy-overshoot intervals.	Frequency-based diagnostic for overshoot episodes.
G_{cof}	Carbon overshoot frequency: mean number of distinct carbon-overshoot intervals.	Frequency-based diagnostic for carbon overshoots.

per activated controller. Unactivated agents remain idle, and no framework is allowed to read hidden simulator channels outside the controller-facing interface. The shared benchmark runtime therefore remains the single source of truth for world evolution, while the framework layer operates only as a controller orchestrator above it.

Framework realization boundary. Both LangGraph [29] and AgentScope [30] are implemented as thin orchestration layers over the same shared event-driven scheduling substrate and controller-facing contract. LangGraph realizes coordination through graph-structured state updates that can carry compact summaries across activations, whereas AgentScope realizes coordination through relay-style message passing with framework-local handoff and observation state. These differences affect how controller-side state is organized inside the framework, but they do not change the benchmark-visible payloads, the legality interface, the event interpreter, or the underlying DES trajectory.

Controller-side statefulness. Framework runtimes may maintain lightweight internal state across activations, such as short-horizon summaries of backlog pressure, recent handoff outcomes, or observation histories. This statefulness is intentionally treated as controller-side memory rather than as part of the shared benchmark state. In other words, physical evolution, task progress, and evaluation traces remain in the benchmark runtime, while framework-specific memory affects only how a controller interprets future benchmark-visible payloads. This distinction is important because it permits implementation-side adaptation without allowing frameworks to bypass the common substrate.

E Additional Experiments

The benchmark defines four reported instances covering different topology scales and difficulty regimes. All experimental results are reported over these instances. Their configurations are summarized in Table 11. Each primary table follows the same 15-metric order as the main text. Each logical full-metric matrix is reported as three continued landscape panels: Panel A foregrounds the primary metrics together with closely related outcome and robustness anchors, Panel B reports the broader constraint/resource/process surface, and Panel C reports coordination, audit, and process-diagnostic metrics. Detailed results provided in Tables 12–27.

Table 7: Coordination efficiency metrics in DESBench.

Metric	Meaning	Note
G_{vb}	Backlog branching width: mean number of legal backlog candidates at Plant backlog-selection epochs.	Plant-level branching width.
G_{bc}	Backlog decision frequency: mean number of backlog-selection decision epochs.	Frequency of Plant-level selection events.
G_{pa}	Plant routing width: mean number of legal Areas at Plant routing epochs.	Plant-to-Area routing width.
G_{ac}	Area routing width: mean number of legal Cells at Area decision epochs.	Area-to-Cell branching width.
G_{dw}	Local dispatch width: mean number of legal dispatch options at Cell execution epochs.	Local dispatch branching factor.
G_{ba}	Backlog ambiguity rate: fraction of backlog-selection epochs with multiple legal choices.	Backlog-level ambiguity.
G_{pa}^{amb}	Plant ambiguity rate: fraction of Plant routing epochs with multiple Areas.	Plant-level ambiguity.
G_{aa}	Area ambiguity rate: fraction of Area decision epochs with multiple Cells.	Area-level ambiguity.
G_{ca}	Cell ambiguity rate: fraction of Cell dispatch epochs with multiple options.	Cell-level ambiguity.
G_{ds}	Decision steps: mean number of decision steps per episode.	High-level coordination burden.
G_{cm}	Coordination messages: mean number of protocol messages exchanged.	Communication cost.
G_{as}	Active agent load: mean cumulative number of agents exposed for decision at each step.	Parallel decision pressure.
G_{wc}	runtime_mean_seconds: mean wall-clock runtime, in seconds.	Real execution cost, not simulation time.
G_{llm}	Model-driven decision count: mean number of LLM-driven decisions.	Model-driven control load.
G_{fb}	Fallback-contract decisions: mean number of fallback decisions.	Reliance on rule-based fallback.

Table 8: Robustness metrics in DESBench.

Metric	Meaning	Note
G_{dl}	Deadlock rate: fraction of episodes marked as deadlocked.	Hard failure indicator.
G_{tr}	Truncation rate: fraction of episodes marked as truncated.	May arise from horizon limits, deadlock termination, or no-progress stopping.
G_{ut}	Unfinished overdue tardiness: mean overdue debt induced by unfinished jobs at episode end.	Service loss from unfinished workload.
G_{dj}	Completion debt (jobs): mean number of unfinished jobs at episode end.	Residual job-level debt.
G_{do}	Completion debt (operations): mean number of remaining operations at episode end.	Residual workload-level debt.

F Using the Benchmark

DESBench is designed for researchers with existing MAS implementations or those developing new coordination methods, who wish to evaluate coordination effectiveness under shared event-driven scheduling dynamics. The benchmark provides the instance configurations, simulator, legal-action validation, execution traces, and evaluation metrics, while the user supplies a MAS runner that consumes the public decision payload and returns valid actions. The recommended usage follows a simple pipeline: select a reported instance, connect the runner, optionally register a new coordination paradigm, execute the provided entry points, and inspect the resulting summaries and trace artifacts.

Table 9: Process-oriented metrics in DESBench.

Metric	Meaning	Note
G_{st}	Setup time: mean cumulative setup time within an episode.	Process overhead.
G_{su}	Setup energy: mean extra energy attributable to setup.	Setup cost diagnostic.
G_{tt}	Transport time: mean total transport time.	Logistics burden.
G_{te}	Transport energy: mean transport energy usage.	Logistics cost.
G_{tm}	Transport moves: mean number of transport moves.	Raw logistics activity count.
G_{bt}	Blocking time: mean blocked time induced by downstream or buffer constraints.	Congestion / blocking indicator.
G_{bw}	Buffer wait time: mean waiting time in buffers.	Queueing / waiting indicator.
G_{ae}	Arrival events: mean number of realized arrival events.	Dynamic workload injection count.
G_{aj}	Arrived jobs: mean number of jobs introduced by arrivals.	Dynamic workload scale.
G_{ab}	Arrival batch size: mean jobs per arrival event.	Arrival-shape diagnostic.
G_{bd}	Breakdown count: mean number of machine breakdown events.	Derived from world-event traces.
G_{dt}	Downtime: mean total machine downtime.	Aggregate post-failure unavailability.
G_{rt}	Repair time: mean total repair duration.	Disturbance burden.

Table 10: Diagnostic and compatibility metrics in DESBench.

Metric	Meaning	Note
G_{vm}	Mixed violation magnitude: legacy mixed violation-magnitude metric.	Aggregates energy and carbon overshoot with inconsistent units.
G_{ic}	Inter-cell moves: mean number of cross-cell moves.	Effectively aliased to transport moves.
G_{ar}	Activation records: mean number of activation-trace records.	Useful for trace volume analysis.
G_{sv}	Settlement events: mean number of settlement-trace events.	Protocol-closure trace indicator.
G_{sc}	Settled contracts: mean number of contracts in settled state at episode end.	Measures contract closure completeness.
G_{oc}	Open contracts: mean number of contracts remaining open at episode end.	Indicates coordination leakage.
G_{uc}	Unsettled completed contracts: mean number of completed but unsettled contracts.	Captures settlement lag.
G_{p1}	framework_pass_at_1_mean: mean first-pass contract acceptance rate across LLM-backed decisions.	Only available for result bundles that explicitly record pass@1 auditing.
G_{na}	No-action events: mean number of decision points with no legal action available.	Debug/system health indicator.

F.1 Choose a Reported Instance

A valid comparison begins by fixing one reported instance in Table 11. The reported instance already fixes topology, route grammar, arrival process, disturbance profile, budgets, and seed split. The four paper instances cover distinct coordination stresses: A3C9-1 for smaller-topology transfer with cross-area coupling, A5C12-1 for branch pressure and routing ambiguity, A5C12-2 for cluster pull and greedy-routing lure, and A5C12-3 for late commitment and long-horizon regret. Once the instance is chosen, it should remain fixed for all methods in the same comparison.

F.2 Connect a MAS Runner

The main integration point is the MAS runner. At each decision step, DESBench provides a decision payload containing the active agents and their local views. Each view includes the agent role, observation, protocol-visible context, and an action mask. The runner must return exactly one valid action for each active agent, consistent with the provided action mask.

Table 11: Benchmark instances. Each reported instance is a stable suite configuration with a fixed topology and route surface.

Instance	Source ID	Topology and route surface	Profile and purpose	Failure and budget
A3C9-1	intercell_a3c9_cross_area_tradeoff	tree_a3_c9; three Areas and nine Cells; three-stage a3c9_canonical_non_degenerate_v1 route grammar	Cross-area tradeoff; max in-flight jobs 4; backlog top-3; transport multiplier 1.15; inbound cap 1; smaller-topology transfer and cross-area routing-regret test.	Weibull-aging failures from the provided profile metadata; standard R0 WIP, energy, and carbon case budgets
A5C12-1	intercell_a5c12_branch_pressure	tree_a5_c12; five Areas and twelve Cells with a 4-3-2-2-1 asymmetric partition; four-stage a5c12_asymmetric_long_horizon_v1 route grammar	Branch-pressure anchor; max in-flight jobs 4; backlog top-4; transport multiplier 1.15; default inbound cap 1; later weak cells (3, 6, 8, 10, 11) are slowed.	Exponential nominal failures; A5C12 R0 WIP, energy, and carbon budgets fixed across protocols.
A5C12-2	intercell_a5c12_cluster_pull	tree_a5_c12; five Areas and twelve Cells with a 4-3-2-2-1 asymmetric partition; four-stage a5c12_asymmetric_long_horizon_v1 route grammar	Strong-cluster-pull / greedy routing lure; max in-flight jobs 4; backlog top-4; transport multiplier 1.20; default inbound cap 1; cells (0, 1, 4) are faster while (7, 9, 11) are slower.	Weibull-aging failures; same A5C12 R0 budget-accounting surface as A5C12-1.
A5C12-3	intercell_a5c12_late_commit	tree_a5_c12; five Areas and twelve Cells with a 4-3-2-2-1 asymmetric partition; four-stage a5c12_asymmetric_long_horizon_v1 route grammar	Late-commitment / long-horizon commitment regret; max in-flight jobs 4; backlog top-4; transport multiplier 1.25; default inbound cap 1; late-stage weak cells (2, 6, 8, 10, 11) are slowed.	Load-dependent failures; same A5C12 R0 budget-accounting surface, emphasizing long-horizon commitment cost.

Table 12: Primary 15-metric matrix for A3C9-1.

Paradigm	Framework	LLM	Evaluation Metrics															
			$\mathcal{G}_{\text{eff}}$				$\mathcal{G}_{\text{con}}$				$\mathcal{G}_{\text{coord}}$				$\mathcal{G}_{\text{rob}}$			
			$G_{\text{mk}} \downarrow$	$G_{\text{td}} \downarrow$	$G_{\text{ej}} \uparrow$	$G_{\text{vc}} \downarrow$	$G_{\text{eo}} \downarrow$	$G_{\text{ed}} \downarrow$	$G_{\text{en}} \downarrow$	$G_{\text{ds}} \downarrow$	$G_{\text{cm}} \downarrow$	$G_{\text{fb}} \downarrow$	$G_{\text{ilm}} \downarrow$	$G_{\text{tr}} \downarrow$	$G_{\text{dj}} \downarrow$	$G_{\text{ut}} \downarrow$	$G_{\text{do}} \downarrow$	
Γ^{cen}	LangGraph	GPT-5.4	54.7	116.2	10.0	22.2	3277.2	5.0	6562.7	90.0	90.0	0.00	90.0	0.00	0.00	0.00	0.00	
		Gemini-3-Flash	55.2	107.5	10.0	20.6	3542.1	6.1	6620.9	90.0	90.0	1.6	88.4	0.00	0.00	0.00	0.00	
		Qwen-3.5	54.7	115.1	10.0	23.2	2962.8	6.0	6554.1	90.0	90.0	0.00	90.0	0.00	0.00	0.00	0.00	
	AgentScope	GPT-5.4	55.2	121.0	10.0	21.8	3363.8	5.5	6622.2	90.0	90.0	0.00	90.0	0.00	0.00	0.00	0.00	
		Gemini-3-Flash	55.0	106.7	10.0	20.6	3564.7	6.0	6602.4	90.0	90.0	1.8	88.2	0.00	0.00	0.00	0.00	
		Qwen-3.5	53.8	113.6	10.0	22.4	2841.7	5.2	6458.8	90.0	90.0	0.00	90.0	0.00	0.00	0.00	0.00	
Γ^{hie}	LangGraph	GPT-5.4	41.2	37.2	10.0	13.0	554.8	11.2	5117.1	149.0	90.0	0.00	150.0	0.00	0.00	0.00	0.00	
		Gemini-3-Flash	43.1	37.2	10.0	13.2	636.4	12.9	5326.6	149.0	90.0	2.3	147.7	0.00	0.00	0.00	0.00	
		Qwen-3.5	41.9	32.4	10.0	11.8	710.0	11.3	5220.2	149.0	90.0	0.00	150.0	0.00	0.00	0.00	0.00	
	AgentScope	GPT-5.4	40.4	36.4	10.0	14.8	709.5	11.0	5040.9	149.0	90.0	0.00	150.0	0.00	0.00	0.00	0.00	
		Gemini-3-Flash	43.2	37.1	10.0	13.2	651.1	12.9	5331.1	149.0	90.0	2.3	147.7	0.00	0.00	0.00	0.00	
		Qwen-3.5	42.4	32.1	10.0	10.6	725.4	11.1	5272.8	149.0	90.0	0.00	150.0	0.00	0.00	0.00	0.00	
Γ^{het}	LangGraph	GPT-5.4	40.9	34.7	10.0	11.6	588.9	10.8	5104.1	181.9	236.2	1.0	181.9	0.00	0.00	0.00	0.00	
		Gemini-3-Flash	42.9	37.0	10.0	13.4	653.0	12.9	5313.3	149.0	180.0	2.3	147.7	0.00	0.00	0.00	0.00	
		Qwen-3.5	42.8	36.7	10.0	12.2	737.1	11.7	5319.7	151.8	184.9	0.00	152.8	0.00	0.00	0.00	0.00	
	AgentScope	GPT-5.4	41.6	32.9	10.0	12.6	759.4	11.9	5184.1	186.5	244.8	0.00	187.5	0.00	0.00	0.00	0.00	
		Gemini-3-Flash	42.9	37.0	10.0	13.4	653.0	12.9	5313.3	149.0	180.0	2.5	147.5	0.00	0.00	0.00	0.00	
		Qwen-3.5	41.7	32.6	10.0	9.4	608.0	10.1	5192.7	151.4	184.2	0.00	152.4	0.00	0.00	0.00	0.00	
Γ^{hol}	LangGraph	GPT-5.4	91.5	18.8	6.8	9.0	5528.1	61.3	10430.3	191.1	136.7	0.40	191.9	1.00	3.2	201.4	5.3	
		Gemini-3-Flash	43.1	37.2	10.0	13.2	636.4	12.9	5326.6	149.0	90.0	2.3	147.7	0.00	0.00	0.00	0.00	
		Qwen-3.5	108.3	103.9	9.3	20.0	25287.8	77.7	12379.6	145.8	89.1	0.00	147.6	0.60	0.70	97.0	1.4	
	AgentScope	GPT-5.4	93.5	22.9	7.6	11.8	7997.2	64.8	10674.6	167.2	112.6	0.20	168.7	0.90	2.4	169.3	4.1	
		Gemini-3-Flash	43.1	37.2	10.0	13.2	636.4	12.9	5326.6	149.0	90.0	2.3	147.7	0.00	0.00	0.00	0.00	
		Qwen-3.5	94.1	107.3	9.4	16.4	30112.4	62.2	10838.2	146.7	89.8	0.00	148.5	0.40	0.60	114.1	1.1	

```

class MyMASRunner:
    def select_actions(self, decision):
        actions = {}
        for agent_id, view in decision["active_agents"].items():
            mask = view["action_mask"]
            actions[agent_id] = choose_legal_action(view, mask)
        return actions

```

Table 13: Logical full-metric matrix for A3C9-1 (Panel A)

Paradigm Framework LLM			Panel A: Primary Outcomes + Robustness Anchors																		
			G^{mk}	G^{mk}_{σ}	G^{td}	G^{tcd}	G^{cf}	G^{th}	G^{sr}	G^{lp}	G^{tr}	G^{4n}	G^{fp}	G^{op}	G^{vo}	G^{en}	G^{do}	G^{ad}	G^{sp}	G^{cm}	
Γ^{cen}	LangGraph	GPT-5.4	54.7	1.57	116.2	116.2	10.0	0.183	1.00	0.000	0.000	0.000	0.000	0.000	0.000	6562.7	22.2	3277.2	5.00	90.0	90.0
	LangGraph	Gemini-3-Flash	55.2	1.08	107.5	107.5	10.0	0.181	1.00	0.000	0.000	0.000	0.000	0.000	0.000	6620.9	20.6	3542.1	6.13	90.0	90.0
	LangGraph	Qwen-3.5	54.7	2.05	115.1	115.1	10.0	0.183	1.00	0.000	0.000	0.000	0.000	0.000	0.000	6554.1	23.2	2962.8	6.01	90.0	90.0
	AgentScope	GPT-5.4	55.2	2.55	121.0	121.0	10.0	0.182	1.00	0.000	0.000	0.000	0.000	0.000	0.000	6622.2	21.8	3363.8	5.49	90.0	90.0
	AgentScope	Gemini-3-Flash	55.0	0.996	106.7	106.7	10.0	0.182	1.00	0.000	0.000	0.000	0.000	0.000	0.000	6602.4	20.6	3564.7	5.95	90.0	90.0
	AgentScope	Qwen-3.5	53.8	1.35	113.6	113.6	10.0	0.186	1.00	0.000	0.000	0.000	0.000	0.000	0.000	6458.8	22.4	2841.7	5.24	90.0	90.0
Γ^{hie}	LangGraph	GPT-5.4	41.2	2.69	37.2	37.2	10.0	0.244	1.00	0.000	0.000	0.000	0.000	0.000	0.000	5117.1	13.0	554.8	11.2	149.0	90.0
	LangGraph	Gemini-3-Flash	43.1	4.54	37.2	37.2	10.0	0.234	1.00	0.000	0.000	0.000	0.000	0.000	0.000	5326.6	13.2	636.4	12.9	149.0	90.0
	LangGraph	Qwen-3.5	41.9	3.02	32.4	32.4	10.0	0.240	1.00	0.000	0.000	0.000	0.000	0.000	0.000	5220.2	11.8	710.0	11.3	149.0	90.0
	AgentScope	GPT-5.4	40.4	2.75	36.4	36.4	10.0	0.249	1.00	0.000	0.000	0.000	0.000	0.000	0.000	5040.9	14.8	709.5	11.0	149.0	90.0
	AgentScope	Gemini-3-Flash	43.2	4.53	37.1	37.1	10.0	0.234	1.00	0.000	0.000	0.000	0.000	0.000	0.000	5331.1	13.2	651.1	12.9	149.0	90.0
	AgentScope	Qwen-3.5	42.4	2.40	32.1	32.1	10.0	0.236	1.00	0.000	0.000	0.000	0.000	0.000	0.000	5272.8	10.6	725.4	11.1	149.0	90.0
Γ^{het}	LangGraph	GPT-5.4	40.9	2.14	34.7	34.7	10.0	0.245	1.00	0.000	0.000	0.000	0.000	0.000	0.000	5104.1	11.6	588.9	10.8	181.9	236.2
	LangGraph	Gemini-3-Flash	42.9	1.08	37.0	37.0	10.0	0.233	1.00	0.000	0.000	0.000	0.000	0.000	0.000	5313.3	13.4	653.0	12.9	149.0	180.0
	LangGraph	Qwen-3.5	42.8	4.48	36.7	36.7	10.0	0.236	1.00	0.000	0.000	0.000	0.000	0.000	0.000	5319.7	12.2	737.1	11.7	151.8	184.9
	AgentScope	GPT-5.4	41.6	2.69	32.9	32.9	10.0	0.241	1.00	0.000	0.000	0.000	0.000	0.000	0.000	5184.1	12.6	759.4	11.9	186.5	244.8
	AgentScope	Gemini-3-Flash	42.9	1.08	37.0	37.0	10.0	0.233	1.00	0.000	0.000	0.000	0.000	0.000	0.000	5313.3	13.4	653.0	12.9	149.0	180.0
	AgentScope	Qwen-3.5	41.7	1.84	32.6	32.6	10.0	0.240	1.00	0.000	0.000	0.000	0.000	0.000	0.000	5192.7	9.40	608.0	10.1	151.4	184.2
Γ^{hol}	LangGraph	GPT-5.4	91.5	11.8	18.8	220.2	6.80	0.076	0.000	0.000	1.00	201.4	3.20	5.30	10430.3	9.0	5528.1	61.3	191.1	136.7	
	LangGraph	Gemini-3-Flash	43.1	4.54	37.2	37.2	10.0	0.234	1.00	0.000	0.000	0.000	0.000	0.000	0.000	5326.6	13.2	636.4	12.9	149.0	90.0
	LangGraph	Qwen-3.5	108.3	66.2	103.9	200.9	9.30	0.134	0.400	0.000	0.600	97.0	0.700	1.40	12379.6	20.0	25287.8	77.7	145.8	89.1	
	AgentScope	GPT-5.4	93.5	23.1	22.9	192.2	7.60	0.092	0.100	0.000	0.900	169.3	2.40	4.10	10674.6	11.8	7997.2	64.8	167.2	112.6	
	AgentScope	Gemini-3-Flash	43.1	4.54	37.2	37.2	10.0	0.234	1.00	0.000	0.000	0.000	0.000	0.000	0.000	5326.6	13.2	636.4	12.9	149.0	90.0
	AgentScope	Qwen-3.5	94.1	80.7	107.3	221.5	9.40	0.173	0.600	0.000	0.000	0.400	114.1	0.600	1.10	10838.2	16.4	30112.4	62.2	146.7	89.8

Table 14: Logical full-metric matrix for A3C9-1 (Panel B).

Paradigm Framework LLM			Panel B: Constraint / Resource / Process																			
			$\odot$	$\vee$	G_{com}	G_{cod}	G_{foa}	G_{cof}	G_{st}	G_{ns}	G_{tt}	G_{et}	G_{tm}	G_{ic}	G_{p}	G_{bw}	G_{ae}	G_{f}	G_{a}	G_{p}	G_{d}	G_{t}
Γ^{cen}	LangGraph	GPT-5.4	3937.6	5243.5	1966.3	5.00	11.1	11.1	5.32	31.9	49.2	9.84	30.0	30.0	0.000	0.000	4.00	7.00	1.75	0.100	0.011	0.011
	LangGraph	Gemini-3-Flash	3972.5	5667.4	2125.3	6.13	10.3	10.3	5.04	30.3	48.9	9.78	30.0	30.0	0.000	0.000	4.00	7.00	1.75	0.000	0.000	0.000
	LangGraph	Qwen-3.5	3932.4	4740.5	1777.7	6.01	11.6	11.6	4.15	24.9	48.6	9.73	30.0	30.0	0.000	0.000	4.00	7.00	1.75	0.000	0.000	0.000
	AgentScope	GPT-5.4	3973.3	5382.1	2018.3	5.49	10.9	10.9	4.76	28.6	49.5	9.89	30.0	30.0	0.000	0.000	4.00	7.00	1.75	0.100	0.925	0.925
	AgentScope	Gemini-3-Flash	3961.5	5703.5	2138.8	5.95	10.3	10.3	4.82	28.9	48.9	9.78	30.0	30.0	0.000	0.000	4.00	7.00	1.75	0.000	0.000	0.000
	AgentScope	Qwen-3.5	3875.3	4546.7	1705.0	5.24	11.2	11.2	4.47	26.8	48.5	9.71	30.0	30.0	0.000	0.000	4.00	7.00	1.75	0.100	0.011	0.011
Γ^{hie}	LangGraph	GPT-5.4	3070.3	887.7	332.9	11.2	6.50	6.50	3.42	20.5	49.3	9.87	30.0	30.0	0.000	0.000	4.00	7.00	1.75	0.600	2.34	2.34
	LangGraph	Gemini-3-Flash	3195.9	1018.3	381.8	12.9	6.60	6.60	3.79	22.8	49.5	9.89	30.0	30.0	0.000	0.000	4.00	7.00	1.75	0.600	4.25	4.25
	LangGraph	Qwen-3.5	3132.1	1135.9	426.0	11.3	5.90	5.90	3.30	19.8	49.7	9.94	30.0	30.0	0.000	0.000	4.00	7.00	1.75	0.700	1.37	1.37
	AgentScope	GPT-5.4	3024.5	1135.3	425.7	11.0	7.40	7.40	3.53	21.2	49.6	9.92	30.0	30.0	0.000	0.000	4.00	7.00	1.75	0.400	1.39	1.39
	AgentScope	Gemini-3-Flash	3198.7	1041.7	390.6	12.9	6.60	6.60	3.75	22.5	49.4	9.88	30.0	30.0	0.000	0.000	4.00	7.00	1.75	0.600	4.25	4.25
	AgentScope	Qwen-3.5	3163.7	1160.7	435.2	11.1	5.30	5.30	3.22	19.3	49.7	9.94	30.0	30.0	0.000	0.000	4.00	7.00	1.75	0.600	1.72	1.72
Γ^{het}	LangGraph	GPT-5.4	3062.5	942.3	353.4	10.8	5.80	5.80	3.64	21.8	49.3	9.87	30.0	30.0	0.000	0.000	4.00	7.00	1.75	0.500	2.19	2.19
	LangGraph	Gemini-3-Flash	3188.0	1044.8	391.8	12.9	6.70	6.70	4.16	25.0	49.6	9.91	30.0	30.0	0.000	0.000	4.00	7.00	1.75	0.600	3.06	3.06
	LangGraph	Qwen-3.5	3191.8	1179.4	442.3	11.7	6.10	6.10	3.65	21.9	49.9	9.98	30.0	30.0	0.000	0.000	4.00	7.00	1.75	0.800	3.42	3.42
	AgentScope	GPT-5.4	3110.5	1215.1	455.7	11.9	6.30	6.30	4.08	24.5	49.8	9.96	30.0	30.0	0.000	0.000	4.00	7.00	1.75	0.300	0.562	0.562
	AgentScope	Gemini-3-Flash	3188.0	1044.8	391.8	12.9	6.70	6.70	4.16	25.0	49.6	9.91	30.0	30.0	0.000	0.000	4.00	7.00	1.75	0.600	3.06	3.06
	AgentScope	Qwen-3.5	3115.6	972.7	364.8	10.1	4.70	4.70	3.36	20.1	49.6	9.92	30.0	30.0	0.000	0.000	4.00	7.00	1.75	0.400	1.52	1.52
Γ^{hol}	LangGraph	GPT-5.4	6258.2	8845.0	3316.9	61.3	4.50	4.50	2.36	14.2	44.4	8.88	26.8	26.8	0.000	0.000	4.00	7.00	1.75	0.600	2.26	2.26
	LangGraph	Gemini-3-Flash	3195.9	1018.3	381.8	12.9	6.60	6.60	3.79	22.8	49.5	9.89	30.0	30.0	0.000	0.000	4.00	7.00	1.75	0.600	4.25	4.25
	LangGraph	Qwen-3.5	7427.8	40460.4	15172.7	77.7	10.0	10.0	4.55	27.3	48.2	9.65	29.3	29.3	0.000	0.000	4.00	7.00	1.75	0.600	2.52	2.52
	AgentScope	GPT-5.4	6404.8	12795.6	4798.3	64.8	5.90	5.90	2.41	14.4	46.5	9.29	28.0	28.0	0.000	0.000	4.00	7.00	1.75	0.200	1.10	1.10
	AgentScope	Gemini-3-Flash	3195.9	1018.3	381.8	12.9	6.60	6.60	3.79	22.8	49.5	9.89	30.0	30.0	0.000	0.000	4.00	7.00	1.75	0.600	4.25	4.25
	AgentScope	Qwen-3.5	6502.9	48179.8	18067.4	62.2	8.20	8.20	3.39	20.3	48.5	9.69	29.5	29.5	0.000	0.000	4.00	7.00	1.75	0.600	2.08	2.08

Table 15: Logical full-metric matrix for A3C9-1 (Panel C).

Paradigm Framework LLM		Panel C: Coordination / Audit / Diagnostics																			
		G^{v_b}	G^{v_c}	G^{p_a}	G^{a_c}	G^{d_w}	G^{b_a}	G^{a_m}	G^{a_a}	G^{c_a}	G^{a_s}	G^{w_c}	G^{a_r}	G^{s_v}	G^{s_c}	G^{o_c}	G^{u_c}	G^{l_m}	G^{f_b}	G^{u_a}	
Γ^{cen}	LangGraph	GPT-5.4	2.10	30.0	0.000	0.000	2.10	0.710	0.000	0.000	0.733	90.0	1095.9	90.0	120.0	10.0	20.0	0.000	90.0	0.000	0.000
	LangGraph	Gemini-3-Flash	2.17	30.0	0.000	0.000	2.10	0.677	0.000	0.000	0.733	90.0	417.0	90.0	120.0	10.0	20.0	0.000	88.4	1.60	0.000
	LangGraph	Qwen-3.5	2.17	30.0	0.000	0.000	2.10	0.777	0.000	0.000	0.733	90.0	108.3	90.0	120.0	10.0	20.0	0.000	90.0	0.000	0.000
	AgentScope	GPT-5.4	2.09	30.0	0.000	0.000	2.10	0.703	0.000	0.000	0.733	90.0	1094.3	90.0	120.0	10.0	20.0	0.000	90.0	0.000	0.000
	AgentScope	Gemini-3-Flash	2.17	30.0	0.000	0.000	2.10	0.673	0.000	0.000	0.733	90.0	426.2	90.0	120.0	10.0	20.0	0.000	88.2	1.80	0.000
	AgentScope	Qwen-3.5	2.17	30.0	0.000	0.000	2.10	0.783	0.000	0.000	0.733	90.0	109.7	90.0	120.0	10.0	20.0	0.000	90.0	0.000	0.000
Γ^{hie}	LangGraph	GPT-5.4	1.43	30.0	2.96	2.00	2.10	0.267	1.00	1.00	0.733	150.0	674.4	150.0	120.0	10.0	20.0	0.000	150.0	0.000	0.000
	LangGraph	Gemini-3-Flash	1.43	30.0	2.94	2.00	2.10	0.267	1.00	1.00	0.733	150.0	631.6	150.0	120.0	10.0	20.0	0.000	147.7	2.30	0.000
	LangGraph	Qwen-3.5	1.43	30.0	2.86	2.00	2.07	0.267	1.00	1.00	0.720	150.0	205.0	150.0	120.0	10.0	20.0	0.000	150.0	0.000	0.000
	AgentScope	GPT-5.4	1.43	30.0	2.97	2.00	2.10	0.267	1.00	1.00	0.733	150.0	673.5	150.0	120.0	10.0	20.0	0.000	150.0	0.000	0.000
	AgentScope	Gemini-3-Flash	1.43	30.0	2.94	2.00	2.10	0.267	1.00	1.00	0.733	150.0	637.9	150.0	120.0	10.0	20.0	0.000	147.7	2.30	0.000
	AgentScope	Qwen-3.5	1.43	30.0	2.87	2.00	2.07	0.267	1.00	1.00	0.720	150.0	209.5	150.0	120.0	10.0	20.0	0.000	150.0	0.000	0.000
Γ^{het}	LangGraph	GPT-5.4	1.45	30.5	2.94	1.96	2.09	0.272	1.00	0.964	0.730	182.9	1228.1	182.9	136.2	10.0	20.0	0.000	181.9	1.00	0.000
	LangGraph	Gemini-3-Flash	1.43	30.0	2.94	2.00	2.10	0.267	1.00	1.00	0.733	150.0	667.1	150.0	120.0	10.0	20.0	0.000	147.7	2.30	0.000
	LangGraph	Qwen-3.5	1.43	30.0	2.90	2.00	2.09	0.267	1.00	0.997	0.730	152.8	193.3	152.8	121.4	10.0	20.0	0.000	152.8	0.000	0.000
	AgentScope	GPT-5.4	1.43	30.3	2.95	1.98	2.10	0.267	1.00	0.981	0.733	187.5	810.4	187.5	138.6	10.0	20.0	0.000	187.5	0.000	0.000
	AgentScope	Gemini-3-Flash	1.43	30.0	2.94	2.00	2.10	0.267	1.00	1.00	0.733	150.0	704.3	150.0	120.0	10.0	20.0	0.000	147.5	2.50	0.000
	AgentScope	Qwen-3.5	1.43	30.0	2.88	2.00	2.09	0.267	1.00	1.00	0.730	152.4	192.5	152.4	121.2	10.0	20.0	0.000	152.4	0.000	0.000
Γ^{hol}	LangGraph	GPT-5.4	1.52	28.9	2.95	1.95	2.07	0.340	1.00	0.948	0.697	192.3	845.1	193.0	148.2	6.8	20.0	0.000	191.9	0.400	0.000
	LangGraph	Gemini-3-Flash	1.43	30.0	2.94	2.00	2.10	0.267	1.00	1.00	0.733	150.0	703.3	150.0	120.0	10.0	20.0	0.000	147.7	2.30	0.000
	LangGraph	Qwen-3.5	1.50	29.3	2.88	2.00	2.07	0.322	1.00	1.00	0.716	147.6	186.9	147.6	117.6	9.30	20.0	0.000	147.6	0.000	0.000
	AgentScope	GPT-5.4	1.50	28.9	2.94	1.98	2.09	0.316	1.00	0.981	0.722	168.9	995.7	169.2	132.5	7.6	20.4	0.000	168.7	0.200	0.000
	AgentScope	Gemini-3-Flash	1.43	30.0	2.94	2.00	2.10	0.267	1.00	1.00	0.733	150.0	704.2	150.0	120.0	10.0	20.0	0.000	147.7	2.30	0.000
	AgentScope	Qwen-3.5	1.49	29.5	2.86	2.00	2.07	0.310	1.00	1.00	0.719	148.5	189.1	148.5	118.4	9.40	20.1	0.000	148.5	0.000	0.000

Table 16: Primary 15-metric matrix for A5C12-1.

Paradigm	Framework	LLM	Evaluation Metrics														
			$\mathcal{G}_{\text{eff}}$			$\mathcal{G}_{\text{con}}$				$\mathcal{G}_{\text{coord}}$				$\mathcal{G}_{\text{rob}}$			
			$G_{mk} \downarrow$	$G_{td} \downarrow$	$G_{ej} \uparrow$	$G_{vc} \downarrow$	$G_{eo} \downarrow$	$G_{ed} \downarrow$	$G_{en} \downarrow$	$G_{ds} \downarrow$	$G_{cm} \downarrow$	$G_{fb} \downarrow$	$G_{llm} \downarrow$	$G_{tr} \downarrow$	$G_{dj} \downarrow$	$G_{ut} \downarrow$	$G_{do} \downarrow$
Γ^{cen}	LangGraph	GPT-5.4	78.2	242.3	10.0	32.4	12636.8	10.7	12255.4	120.0	120.0	0.2	119.8	0.00	0.0	0.0	0.0
		Gemini-3-Flash	73.9	230.2	10.0	31.2	10389.3	8.0	11621.4	120.0	120.0	2.2	117.8	0.00	0.0	0.0	0.0
		Qwen-3.5	68.8	211.1	9.4	27.6	8862.4	6.2	10851.9	114.9	114.9	0.0	114.9	0.10	0.6	6.1	2.0
	AgentScope	GPT-5.4	77.4	253.2	10.0	33.2	13662.8	9.3	12135.3	120.0	120.0	0.1	119.9	0.00	0.0	0.0	0.0
		Gemini-3-Flash	73.9	230.2	10.0	31.2	10388.9	8.0	11621.1	120.0	120.0	2.2	117.8	0.00	0.0	0.0	0.0
		Qwen-3.5	72.4	217.4	9.4	28.0	10366.9	9.3	11371.9	114.9	114.9	0.0	114.9	0.10	0.6	6.1	2.0
Γ^{hie}	LangGraph	GPT-5.4	65.7	142.4	9.0	59.6	15253.6	35.7	10535.0	190.6	115.0	0.2	191.4	0.20	1.0	14.4	2.4
		Gemini-3-Flash	60.7	132.9	9.5	50.0	10981.7	28.0	9812.6	194.8	117.5	0.9	194.9	0.10	0.5	4.1	1.2
		Qwen-3.5	56.2	101.2	7.6	45.0	11969.5	27.7	9036.7	170.1	103.2	0.0	171.9	0.40	2.4	16.8	7.0
	AgentScope	GPT-5.4	66.0	135.2	9.3	59.0	12418.3	35.4	10576.0	194.1	117.1	0.1	195.0	0.20	0.7	17.0	1.6
		Gemini-3-Flash	60.7	132.9	9.5	50.0	10981.7	28.0	9812.6	194.8	117.5	0.9	194.9	0.10	0.5	4.1	1.2
		Qwen-3.5	51.7	98.6	7.2	42.3	10724.3	24.5	8322.2	160.6	97.6	0.0	162.4	0.40	2.8	7.9	9.0
Γ^{het}	LangGraph	GPT-5.4	64.8	147.7	10.0	56.0	10710.7	31.6	10379.3	319.0	450.0	5.5	314.5	0.00	0.0	0.0	0.0
		Gemini-3-Flash	60.9	134.7	9.5	51.6	11320.0	28.4	9846.3	197.6	240.0	0.5	198.1	0.10	0.5	4.1	1.2
		Qwen-3.5	71.1	155.9	10.0	62.8	16361.2	40.1	11367.2	212.6	265.2	0.0	214.4	0.00	0.0	0.0	0.0
	AgentScope	GPT-5.4	60.1	136.1	9.5	49.4	8166.8	26.1	9677.2	322.9	460.3	0.0	323.9	0.10	0.5	6.6	1.6
		Gemini-3-Flash	60.9	135.4	9.5	51.4	11344.0	28.3	9839.8	198.0	240.7	0.5	198.5	0.10	0.5	4.1	1.2
		Qwen-3.5	69.2	141.8	9.4	58.2	15574.7	38.3	11038.3	204.9	255.2	0.0	206.7	0.10	0.6	4.5	1.6
Γ^{hol}	LangGraph	GPT-5.4	25.0	13.6	2.7	6.0	975.5	3.3	4047.4	123.8	87.1	0.0	124.8	0.90	7.3	12.8	24.0
		Gemini-3-Flash	56.4	103.7	8.5	47.4	9151.7	23.8	9137.0	190.5	115.4	0.9	190.6	0.50	1.5	26.3	3.2
		Qwen-3.5	40.7	66.4	5.5	26.8	7088.2	15.9	6560.0	138.5	86.9	0.0	140.2	0.60	4.5	6.2	15.3
	AgentScope	GPT-5.4	29.0	11.3	3.6	7.9	1009.2	3.1	4718.2	140.6	96.8	0.0	141.6	1.00	6.4	24.0	20.9
		Gemini-3-Flash	56.4	103.7	8.5	47.4	9151.7	23.8	9137.0	190.5	115.4	1.0	190.5	0.50	1.5	26.3	3.2
		Qwen-3.5	32.5	50.2	4.3	19.4	5179.6	11.4	5246.4	115.8	73.5	0.0	117.5	0.70	5.7	3.2	20.1

This interface is framework-agnostic. The runner can be implemented using LLMs, learned policies, rule-based logic, or custom orchestration. It operates solely on the public payload and must respect the active-agent set and action constraints.

F.3 Register a New Paradigm

If the contribution is a new coordination paradigm, it should be assigned a public id and registered before running the suite. A paradigm registration specifies the benchmark-visible coordination semantics, including routing authority, rejection rights, local rerouting, escalation conditions, and negotiation mechanisms. The repository provides `custom_paradigm.py` as a minimal template.

```
from fms_bench.paradigms import AuthorityMode, register_authority_mode

register_authority_mode(
    AuthorityMode(
        mode_id="my_hybrid",
        routing_authority_mode="holonic_hybrid",
        cell_contract_actions="accept_reject",
        area_local_reroute_enabled=True,
        plant_reroute_required=False,
        mediated_bid_round_enabled=True,
    )
)
```

This registration defines the coordination semantics exposed by DESBench, while the MAS runner determines how agents act under these semantics. The coordination structure may be static or dynamic. For example, authority and interaction patterns may shift over time between local and global coordination regimes depending on system state or event context.

If the paradigm can be expressed through existing authority switches, a lightweight registration is sufficient. More substantial extensions (e.g., new activation types, message objects, legal actions, or settlement semantics) require corresponding runtime modifications and validation.

For paradigm-level comparison, the reported instance, seed split, runner interface, simulator dynamics, and metrics should remain fixed. The paradigm may change authority, information flow, negotiation, rerouting, or escalation, but should not alter topology, route grammar, hidden simulator state, or metric definitions.

Table 17: Logical full-metric matrix for A5C12-1 (Panel A).

Paradigm Framework LLM		Panel A: Primary Outcomes + Robustness Anchors														
		$\pi_{\text{ur}}^{\text{ur}}$	$\pi_{\text{ur}}^{\text{p}}$	$\rho_{\text{p}}^{\text{p}}$	$\rho_{\text{p}}^{\text{ur}}$	$\rho_{\text{p}}^{\text{ur}}$	$\rho_{\text{p}}^{\text{ur}}$	$\rho_{\text{p}}^{\text{ur}}$	$\rho_{\text{p}}^{\text{ur}}$	$\rho_{\text{p}}^{\text{ur}}$	$\rho_{\text{p}}^{\text{ur}}$	$\rho_{\text{p}}^{\text{ur}}$	$\rho_{\text{p}}^{\text{ur}}$	$\rho_{\text{p}}^{\text{ur}}$	$\rho_{\text{p}}^{\text{ur}}$	$\rho_{\text{p}}^{\text{ur}}$
Γ^{cen}	LangGraph	GPT-5.4	78.2	4.39	242.3	242.3	10.0	0.128	1.00	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	LangGraph	Gemini-3-Flash	73.9	2.18	230.2	230.2	10.0	0.135	1.00	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	LangGraph	Qwen-3.5	68.8	10.6	211.1	217.2	9.40	0.135	0.900	0.000	0.100	6.10	0.600	2.00	10851.9	27.6
	AgentScope	GPT-5.4	77.4	4.41	253.2	253.2	10.0	0.130	1.00	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	AgentScope	Gemini-3-Flash	73.9	2.18	230.2	230.2	10.0	0.135	1.00	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	AgentScope	Qwen-3.5	72.4	11.9	217.4	223.5	9.40	0.129	0.900	0.000	0.100	6.10	0.600	2.00	11371.9	28.0
Γ^{hie}	LangGraph	GPT-5.4	65.7	14.2	142.4	156.8	9.00	0.137	0.800	0.000	0.200	14.4	1.00	2.40	10535.0	59.6
	LangGraph	Gemini-3-Flash	60.7	8.22	132.9	137.0	9.50	0.156	0.900	0.000	0.100	4.12	0.500	1.20	9812.6	50.0
	LangGraph	Qwen-3.5	56.2	19.7	101.2	118.0	7.60	0.131	0.600	0.000	0.400	16.8	2.40	7.00	9036.7	45.0
	AgentScope	GPT-5.4	66.0	8.85	135.2	152.2	9.30	0.141	0.800	0.000	0.200	17.0	0.700	1.60	10576.0	59.0
	AgentScope	Gemini-3-Flash	60.7	8.22	132.9	137.0	9.50	0.156	0.900	0.000	0.100	4.12	0.500	1.20	9812.6	50.0
	AgentScope	Qwen-3.5	51.7	21.4	98.6	106.5	7.20	0.132	0.600	0.000	0.400	7.86	2.80	9.00	8322.2	42.3
Γ^{het}	LangGraph	GPT-5.4	64.8	4.15	147.7	147.7	10.0	0.155	1.00	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	LangGraph	Gemini-3-Flash	60.9	8.18	134.7	138.8	9.50	0.155	0.900	0.000	0.100	4.12	0.500	1.20	9846.3	51.6
	LangGraph	Qwen-3.5	71.1	4.45	155.9	155.9	10.0	0.141	1.00	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	AgentScope	GPT-5.4	60.1	9.08	136.1	142.6	9.50	0.158	0.900	0.000	0.100	6.55	0.500	1.60	9677.2	49.4
	AgentScope	Gemini-3-Flash	60.9	8.18	135.4	139.6	9.50	0.155	0.900	0.000	0.100	4.12	0.500	1.20	9839.8	51.4
	AgentScope	Qwen-3.5	69.2	12.6	141.8	146.3	9.40	0.136	0.900	0.000	0.100	4.48	0.600	1.60	11038.3	58.2
Γ^{hol}	LangGraph	GPT-5.4	25.0	14.9	13.6	26.4	2.70	0.080	0.100	0.000	0.900	12.8	7.30	24.0	4047.4	6.00
	LangGraph	Gemini-3-Flash	56.4	9.35	103.7	130.0	8.50	0.150	0.500	0.000	0.500	26.3	1.50	3.20	9137.0	47.4
	LangGraph	Qwen-3.5	40.7	24.5	66.4	72.6	5.50	0.120	0.400	0.000	0.600	6.21	4.50	15.3	6560.0	26.8
	AgentScope	GPT-5.4	29.0	10.2	11.3	35.4	3.60	0.119	0.000	0.000	1.00	24.0	6.40	20.9	4718.2	7.90
	AgentScope	Gemini-3-Flash	56.4	9.35	103.7	130.0	8.50	0.150	0.500	0.000	0.500	26.3	1.50	3.20	9137.0	47.4
	AgentScope	Qwen-3.5	32.5	24.6	50.2	53.4	4.30	0.101	0.300	0.000	0.700	3.22	5.70	20.1	5246.4	19.4

Table 18: Logical full-metric matrix for A5C12-1 (Panel B).

Paradigm Framework LLM			Panel B: Constraint / Resource / Process																			
			G^{co}	G^{vm}	G^{com}	G^{pod}	G^{eof}	G^{cof}	G^{ss}	G^{tt}	G^{te}	G^{tm}	G^{ic}	G^{pbt}	G^{bw}	G^{ae}	G^{af}	G^{ab}	G^{pd}	G^{pt}	G^{tr}	
Γ^{cen}	LangGraph	GPT-5.4	7353.2	20218.8	7582.1	10.7	16.2	16.2	9.45	56.7	66.0	13.2	40.0	40.0	0.000	0.000	4.00	7.00	1.75	0.500	1.68	1.68
	LangGraph	Gemini-3-Flash	6972.9	16622.8	6233.6	8.04	15.6	15.6	9.50	57.0	65.0	13.0	40.0	40.0	0.000	0.000	4.00	7.00	1.75	0.500	3.16	3.16
	LangGraph	Qwen-3.5	6511.1	14179.9	5317.4	6.20	13.8	13.8	9.27	55.6	62.2	12.4	38.4	38.4	0.000	0.000	4.00	7.00	1.75	0.300	0.727	0.727
	AgentScope	GPT-5.4	7281.2	21860.4	8197.7	9.28	16.6	16.6	8.88	53.3	66.2	13.2	40.0	40.0	0.000	0.000	4.00	7.00	1.75	0.500	3.69	3.69
	AgentScope	Gemini-3-Flash	6972.7	16622.3	6233.4	8.04	15.6	15.6	9.58	57.5	65.0	13.0	40.0	40.0	0.000	0.000	4.00	7.00	1.75	0.500	3.16	3.16
	AgentScope	Qwen-3.5	6823.1	16587.0	6220.1	9.26	14.0	14.0	9.99	60.0	62.5	12.5	38.4	38.4	0.000	0.000	4.00	7.00	1.75	0.200	1.80	1.80
Γ^{hie}	LangGraph	GPT-5.4	6321.0	24405.7	9152.1	35.7	29.8	29.8	12.5	74.9	65.7	13.1	38.4	38.4	0.000	0.000	4.00	7.00	1.75	4.20	13.4	13.4
	LangGraph	Gemini-3-Flash	5887.6	17570.7	6589.0	28.0	25.0	25.0	9.32	55.9	67.0	13.4	39.2	39.2	0.000	0.000	4.00	7.00	1.75	4.30	12.1	12.1
	LangGraph	Qwen-3.5	5422.0	19151.2	7181.7	27.7	22.5	22.5	10.3	61.6	58.9	11.8	34.5	34.5	0.000	0.000	4.00	7.00	1.75	4.20	13.0	13.0
	AgentScope	GPT-5.4	6345.6	19869.3	7451.0	35.4	29.5	29.5	11.8	71.0	66.9	13.4	39.1	39.1	0.000	0.000	4.00	7.00	1.75	4.30	12.4	12.4
	AgentScope	Gemini-3-Flash	5887.6	17570.7	6589.0	28.0	25.0	25.0	9.32	55.9	67.0	13.4	39.2	39.2	0.000	0.000	4.00	7.00	1.75	4.30	12.1	12.1
	AgentScope	Qwen-3.5	4993.3	17159.1	6434.6	24.5	21.1	21.1	8.24	49.4	55.5	11.1	32.6	32.6	0.000	0.000	4.00	7.00	1.75	3.20	11.9	11.9
Γ^{het}	LangGraph	GPT-5.4	6227.6	17137.0	6426.4	31.6	28.0	28.0	8.27	49.6	67.8	13.6	40.0	40.0	0.000	0.000	4.00	7.00	1.75	3.80	12.7	12.7
	LangGraph	Gemini-3-Flash	5907.8	18111.9	6792.0	28.4	25.8	25.8	9.77	58.6	67.0	13.4	39.2	39.2	0.000	0.000	4.00	7.00	1.75	4.30	12.2	12.2
	LangGraph	Qwen-3.5	6820.3	26177.9	9816.7	40.1	31.4	31.4	13.5	81.1	68.4	13.7	40.0	40.0	0.000	0.000	4.00	7.00	1.75	4.10	14.3	14.3
	AgentScope	GPT-5.4	5806.3	13066.9	4900.1	26.1	24.7	24.7	8.44	50.7	66.0	13.2	38.8	38.8	0.000	0.000	4.00	7.00	1.75	4.40	14.9	14.9
	AgentScope	Gemini-3-Flash	5903.9	18150.5	6806.4	28.3	25.7	25.7	10.3	61.5	67.0	13.4	39.2	39.2	0.000	0.000	4.00	7.00	1.75	4.30	12.2	12.2
	AgentScope	Qwen-3.5	6623.0	24919.5	9344.8	38.3	29.1	29.1	12.0	72.2	66.2	13.2	38.8	38.8	0.000	0.000	4.00	7.00	1.75	4.10	14.4	14.4
Γ^{hol}	LangGraph	GPT-5.4	2428.4	1560.9	585.3	3.33	3.00	3.00	2.15	12.9	33.2	6.64	19.6	19.6	0.000	0.000	4.00	7.00	1.75	1.90	4.48	4.48
	LangGraph	Gemini-3-Flash	5482.2	14642.7	5491.0	23.8	23.7	23.7	9.42	56.5	65.3	13.1	38.2	38.2	0.000	0.000	4.00	7.00	1.75	4.10	11.7	11.7
	LangGraph	Qwen-3.5	3936.0	11341.2	4252.9	15.9	13.4	13.4	5.16	31.0	46.2	9.23	27.1	27.1	0.000	0.000	4.00	7.00	1.75	2.70	7.88	7.88
	AgentScope	GPT-5.4	2830.9	1614.8	605.5	3.11	3.90	3.90	3.05	18.3	38.9	7.79	22.9	22.9	0.000	0.000	4.00	7.00	1.75	2.00	5.52	5.52
	AgentScope	Gemini-3-Flash	5482.2	14642.7	5491.0	23.8	23.7	23.7	9.42	56.5	65.3	13.1	38.2	38.2	0.000	0.000	4.00	7.00	1.75	4.10	11.7	11.7
	AgentScope	Qwen-3.5	3147.8	8287.4	3107.8	11.4	9.70	9.70	4.73	28.4	38.5	7.70	22.7	22.7	0.000	0.000	4.00	7.00	1.75	1.70	6.42	6.42

Table 19: Logical full-metric matrix for A5C12-1 (Panel C).

Paradigm Framework LLM			Panel C: Coordination / Audit / Diagnostics																			
			G^{vb}	G^{bc}	G^{pa}	G^{ac}	G^{dw}	G^{ba}	G^{amb}	G^{aa}	G^{ca}	G^{as}	G^{wc}	G^{ar}	G^{sv}	G^{sc}	G^{oc}	G^{uc}	G^{llm}	G^{fb}	G^{na}	G^{pd}
Γ^{cen}	LangGraph	GPT-5.4	2.00	40.0	0.000	0.000	2.00	0.555	0.000	0.000	0.700	120.0	146.7	120.0	160.0	10.0	30.0	0.000	119.8	0.200	0.000	0.998
	LangGraph	Gemini-3-Flash	1.96	40.0	0.000	0.000	2.02	0.515	0.000	0.000	0.698	120.0	257.3	120.0	160.0	10.0	30.0	0.000	117.8	2.20	0.000	-
	LangGraph	Qwen-3.5	1.91	38.4	0.000	0.000	2.01	0.487	0.000	0.000	0.701	114.9	191.6	114.9	152.8	9.40	29.0	0.000	114.9	0.000	0.000	1.00
	AgentScope	GPT-5.4	2.00	40.0	0.000	0.000	2.02	0.545	0.000	0.000	0.700	120.0	355.8	120.0	160.0	10.0	30.0	0.000	119.9	0.100	0.000	0.999
	AgentScope	Gemini-3-Flash	1.96	40.0	0.000	0.000	2.02	0.515	0.000	0.000	0.698	120.0	335.1	120.0	160.0	10.0	30.0	0.000	117.8	2.20	0.000	-
	AgentScope	Qwen-3.5	1.92	38.4	0.000	0.000	2.01	0.482	0.000	0.000	0.701	114.9	271.0	114.9	152.8	9.40	29.0	0.000	114.9	0.000	0.000	1.00
Γ^{hie}	LangGraph	GPT-5.4	1.42	38.4	3.94	1.37	2.01	0.207	1.00	0.363	0.697	191.6	258.0	191.6	152.0	9.00	29.4	0.000	191.4	0.200	0.000	0.999
	LangGraph	Gemini-3-Flash	1.41	39.2	3.97	1.49	2.03	0.205	1.00	0.471	0.700	195.8	394.3	195.8	156.0	9.50	29.7	0.000	194.9	0.900	0.000	-
	LangGraph	Qwen-3.5	1.48	34.5	3.82	1.35	1.97	0.249	1.00	0.345	0.667	171.9	296.0	171.9	135.0	7.60	26.9	0.000	171.9	0.000	0.000	1.00
	AgentScope	GPT-5.4	1.41	39.1	3.89	1.39	2.01	0.205	1.00	0.383	0.696	195.1	588.7	195.1	155.0	9.30	29.8	0.000	195.0	0.100	0.000	0.999
	AgentScope	Gemini-3-Flash	1.41	39.2	3.97	1.49	2.03	0.205	1.00	0.471	0.700	195.8	434.6	195.8	156.0	9.50	29.7	0.000	194.9	0.900	0.000	-
	AgentScope	Qwen-3.5	1.50	32.6	3.83	1.37	1.98	0.260	1.00	0.358	0.673	162.4	387.7	162.4	127.2	7.20	25.4	0.000	162.4	0.000	0.000	1.00
Γ^{het}	LangGraph	GPT-5.4	1.40	40.0	4.01	1.42	2.01	0.200	1.00	0.388	0.695	320.0	1163.2	320.0	220.0	10.0	30.0	0.000	314.5	5.50	0.000	0.983
	LangGraph	Gemini-3-Flash	1.41	39.2	3.95	1.48	2.02	0.205	1.00	0.467	0.698	198.6	445.3	198.6	157.4	9.50	29.7	0.000	198.1	0.500	0.000	-
	LangGraph	Qwen-3.5	1.42	40.0	3.85	1.32	1.97	0.220	1.00	0.317	0.682	214.4	348.8	214.4	167.2	10.0	30.0	0.000	214.4	0.000	0.000	1.00
	AgentScope	GPT-5.4	1.42	38.8	4.05	1.38	2.01	0.209	1.00	0.348	0.698	323.9	972.5	323.9	219.4	9.50	29.3	0.000	323.9	0.000	0.000	1.00
	AgentScope	Gemini-3-Flash	1.41	39.2	3.95	1.48	2.02	0.205	1.00	0.468	0.698	199.0	595.1	199.0	157.6	9.50	29.7	0.000	198.5	0.500	0.000	-
	AgentScope	Qwen-3.5	1.43	38.8	3.81	1.30	1.98	0.225	1.00	0.299	0.682	206.7	500.7	206.7	160.8	9.40	29.4	0.000	206.7	0.000	0.000	1.00
Γ^{hol}	LangGraph	GPT-5.4	1.50	19.6	4.07	1.36	1.93	0.271	1.00	0.357	0.650	124.8	300.8	124.8	88.2	2.70	16.9	0.000	124.8	0.000	0.000	1.00
	LangGraph	Gemini-3-Flash	1.42	38.2	3.95	1.48	2.03	0.210	1.00	0.466	0.699	191.5	472.3	191.5	151.0	8.50	29.7	0.000	190.6	0.900	0.000	-
	LangGraph	Qwen-3.5	1.55	27.1	3.84	1.35	1.96	0.300	1.00	0.347	0.674	140.2	229.4	140.2	107.6	5.50	21.6	0.000	140.2	0.000	0.000	1.00
	AgentScope	GPT-5.4	1.58	22.9	4.04	1.36	2.04	0.281	1.00	0.355	0.706	141.6	413.1	141.6	101.2	3.60	19.3	0.000	141.6	0.000	0.000	1.00
	AgentScope	Gemini-3-Flash	1.42	38.2	3.95	1.48	2.03	0.210	1.00	0.466	0.699	191.5	367.5	191.5	151.0	8.50	29.7	0.000	190.5	1.00	0.000	-
	AgentScope	Qwen-3.5	1.57	22.7	3.87	1.34	1.90	0.336	1.00	0.334	0.626	117.5	258.0	117.5	89.0	4.30	18.4	0.000	117.5	0.000	0.000	1.00

Table 20: Primary 15-metric matrix for A5C12-2.

Paradigm	Framework	LLM	Evaluation Metrics														
			$\mathcal{G}_{\text{eff}}$			$\mathcal{G}_{\text{con}}$				$\mathcal{G}_{\text{coord}}$				$\mathcal{G}_{\text{rob}}$			
			$G_{\text{mk}} \downarrow$	$G_{\text{td}} \downarrow$	$G_{\text{cj}} \uparrow$	$G_{\text{vc}} \downarrow$	$G_{\text{eo}} \downarrow$	$G_{\text{ed}} \downarrow$	$G_{\text{en}} \downarrow$	$G_{\text{ds}} \downarrow$	$G_{\text{cm}} \downarrow$	$G_{\text{fb}} \downarrow$	$G_{\text{llm}} \downarrow$	$G_{\text{tr}} \downarrow$	$G_{\text{dj}} \downarrow$	$G_{\text{ut}} \downarrow$	$G_{\text{do}} \downarrow$
Γ^{cen}	LangGraph	GPT-5.4	78.3	261.2	10.0	32.6	13454.4	9.4	12238.7	120.0	120.0	0.0	120.0	0.00	0.0	0.0	0.0
		Gemini-3-Flash	72.9	248.4	10.0	32.2	10924.9	4.8	11453.1	120.0	120.0	0.0	120.0	0.00	0.0	0.0	0.0
		Qwen-3.5	74.0	253.9	10.0	32.0	11571.3	6.1	11625.5	120.0	120.0	0.0	120.0	0.00	0.0	0.0	0.0
	AgentScope	GPT-5.4	68.6	208.3	8.3	26.4	9912.3	7.6	10722.2	106.2	105.9	0.1	106.1	0.20	1.7	10.1	5.3
		Gemini-3-Flash	72.9	248.4	10.0	32.2	10924.9	4.8	11453.1	120.0	120.0	0.0	120.0	0.00	0.0	0.0	0.0
		Qwen-3.5	74.9	254.5	10.0	32.0	11686.5	7.0	11741.1	120.0	120.0	0.0	120.0	0.00	0.0	0.0	0.0
Γ^{hie}	LangGraph	GPT-5.4	66.3	144.5	9.4	53.6	12729.2	34.7	10637.5	193.3	116.6	0.1	194.2	0.10	0.6	7.9	1.5
		Gemini-3-Flash	59.8	137.6	10.0	50.2	9592.4	27.2	9689.4	199.0	120.0	0.6	199.4	0.00	0.0	0.0	0.0
		Qwen-3.5	49.5	78.1	6.8	32.2	7485.8	19.3	8007.7	163.6	99.0	0.0	164.6	0.50	3.2	19.4	8.9
	AgentScope	GPT-5.4	61.0	131.8	8.9	48.4	11310.8	30.7	9802.8	183.7	110.9	0.1	184.6	0.20	1.1	5.7	3.8
		Gemini-3-Flash	59.7	137.8	10.0	50.4	9592.0	27.0	9671.2	199.0	120.0	0.7	199.3	0.00	0.0	0.0	0.0
		Qwen-3.5	53.7	90.3	7.4	37.0	8743.4	23.9	8657.1	169.4	102.4	0.0	170.4	0.40	2.6	14.7	7.4
Γ^{het}	LangGraph	GPT-5.4	63.0	144.4	10.0	53.4	8218.5	29.9	10111.8	335.4	478.7	0.0	336.4	0.00	0.0	0.0	0.0
		Gemini-3-Flash	59.7	138.2	10.0	50.4	9544.2	27.0	9671.7	199.0	240.0	0.2	199.8	0.00	0.0	0.0	0.0
		Qwen-3.5	52.5	83.7	7.4	36.0	7582.4	21.4	8479.5	181.8	225.3	0.0	182.9	0.50	2.6	21.5	7.1
	AgentScope	GPT-5.4	59.0	119.8	9.2	41.8	7238.1	25.0	9488.2	301.7	423.9	5.5	297.4	0.20	0.8	11.6	1.9
		Gemini-3-Flash	59.7	138.2	10.0	50.4	9544.2	27.0	9671.7	199.0	240.0	0.2	199.8	0.00	0.0	0.0	0.0
		Qwen-3.5	58.8	109.0	8.4	44.6	9577.6	27.9	9469.8	192.1	237.3	0.0	193.1	0.30	1.6	12.7	4.3
Γ^{hol}	LangGraph	GPT-5.4	30.6	20.0	4.0	7.5	992.5	4.1	4988.9	152.9	106.8	0.0	153.9	0.90	6.0	22.4	19.5
		Gemini-3-Flash	59.7	137.8	10.0	50.4	9592.0	27.0	9671.2	199.0	120.0	0.8	199.2	0.00	0.0	0.0	0.0
		Qwen-3.5	17.6	1.5	1.7	1.0	17.2	0.4	2876.3	79.7	50.8	0.0	80.7	1.00	8.3	8.8	28.3
	AgentScope	GPT-5.4	29.7	7.7	3.9	4.0	105.5	1.1	4849.2	151.3	104.4	0.0	152.6	1.00	6.1	35.3	19.1
		Gemini-3-Flash	59.7	137.8	10.0	50.4	9592.0	27.0	9671.2	199.0	120.0	0.8	199.2	0.00	0.0	0.0	0.0
		Qwen-3.5	18.9	1.8	2.0	1.0	9.6	0.4	3083.0	85.9	54.8	0.0	86.9	1.00	8.0	9.3	27.2

F.4 Run and Audit

DESBench separates MAS connection from experiment execution. For an existing external MAS, `run_mas.py` is the recommended adapter because it exercises the same public decision payload without requiring the MAS to be written in the repository. For built-in or repository-native runners, `run_suite.py` executes a full suite and produces a standard `suite_summary.csv`, while `run_case.py` reruns selected reported cases for detailed diagnosis or reproduction. For quick testing, users may restrict cases and seeds and use a lightweight backend. For comparison, all debug limits should be removed, and the reported instance, paradigm id, runner, model setting, and seed split must be kept identical across methods within a comparison block. Examples are shown below.

```
python examples/run_mas.py \\  
  --suite intercell_a3c9_wide \\  
  --authority-mode heterarchical_cnp \\  
  --output-dir dist/benchmark_runs/my_mas_a3c9 \\  
  --controller-command python my_mas_runner.py
```

```
python examples/run_suite.py \\  
  --framework langgraph \\  
  --suite intercell_a3c9_wide \\  
  --authority-mode heterarchical_cnp \\  
  --llm-backend openai_compat \\  
  --llm-model qwen/qwen3.5-plus-02-15 \\  
  --output-dir dist/benchmark_runs/a3c9_cross_langgraph_heter
```

```
python examples/run_case.py \\  
  --framework agentscope \\  
  --suite intercell_a5c12_harder \\  
  --case-id <exact_case_id_from_suite_summary_or_artifacts> \\  
  --authority-mode holonic_hybrid \\  
  --llm-backend openai_compat \\  
  --llm-model openai/gpt-5.4 \\  
  --output-dir dist/benchmark_runs/a5c12_case_shards
```

Table 21: Logical full-metric matrix for A5C12-2 (Panel A).

Paradigm Framework LLM			Panel A: Primary Outcomes + Robustness Anchors																		
			G_{mk}^a	G_{mk}^s	G_{td}^a	G_{td}^s	G_{ci}^a	G_{th}^a	G_{st}^a	G_{ld}^a	G_{tr}^a	G_{un}^a	G_{vc}^a	G_{oe}^a	G_{pa}^a	G_{sp}^a	G_{uc}^a				
Γ^{cen}	LangGraph	GPT-5.4	78.3	4.59	261.2	261.2	10.0	0.128	1.00	0.000	0.000	0.000	0.000	0.000	12238.7	32.6	13454.4	9.38	120.0	120.0	
	LangGraph	Gemini-3-Flash	72.9	0.961	248.4	248.4	10.0	0.137	1.00	0.000	0.000	0.000	0.000	0.000	11453.1	32.2	10924.9	4.82	120.0	120.0	
	LangGraph	Qwen-3.5	74.0	1.85	253.9	253.9	10.0	0.135	1.00	0.000	0.000	0.000	0.000	0.000	11625.5	32.0	11571.3	6.06	120.0	120.0	
	AgentScope	GPT-5.4	68.6	19.7	208.3	218.5	8.30	0.112	0.800	0.000	0.200	10.1	1.70	5.30	10722.2	26.4	9912.3	7.62	106.2	105.9	
	AgentScope	Gemini-3-Flash	72.9	0.961	248.4	248.4	10.0	0.137	1.00	0.000	0.000	0.000	0.000	0.000	11453.1	32.2	10924.9	4.82	120.0	120.0	
	AgentScope	Qwen-3.5	74.9	2.92	254.5	254.5	10.0	0.134	1.00	0.000	0.000	0.000	0.000	0.000	11741.1	32.0	11686.5	6.95	120.0	120.0	
Γ^{hie}	LangGraph	GPT-5.4	66.3	11.5	144.5	152.4	9.40	0.141	0.900	0.000	0.100	7.87	0.600	1.50	10637.5	53.6	12729.2	34.7	193.3	116.6	
	LangGraph	Gemini-3-Flash	59.8	0.598	137.6	137.6	10.0	0.167	1.00	0.000	0.000	0.000	0.000	0.000	9689.4	50.2	9592.4	27.2	199.0	120.0	
	LangGraph	Qwen-3.5	49.5	18.8	78.1	97.5	6.80	0.131	0.500	0.000	0.500	19.4	3.20	8.90	8007.7	32.2	7485.8	19.3	163.6	99.0	
	AgentScope	GPT-5.4	61.0	16.7	131.8	137.5	8.90	0.147	0.800	0.000	0.200	5.72	1.10	3.80	9802.8	48.4	11310.8	30.7	183.7	110.9	
	AgentScope	Gemini-3-Flash	59.7	0.623	137.8	137.8	10.0	0.168	1.00	0.000	0.000	0.000	0.000	0.000	9671.2	50.4	9592.0	27.0	199.0	120.0	
	AgentScope	Qwen-3.5	53.7	19.4	90.3	105.0	7.40	0.132	0.600	0.000	0.400	14.7	2.60	7.40	8657.1	37.0	8743.4	23.9	169.4	102.4	
Γ^{het}	LangGraph	GPT-5.4	63.0	5.35	144.4	144.4	10.0	0.160	1.00	0.000	0.000	0.000	0.000	0.000	10111.8	53.4	8218.5	29.9	335.4	478.7	
	LangGraph	Gemini-3-Flash	59.7	0.585	138.2	138.2	10.0	0.168	1.00	0.000	0.000	0.000	0.000	0.000	9671.7	50.4	9544.2	27.0	199.0	240.0	
	LangGraph	Qwen-3.5	52.5	16.7	83.7	105.3	7.40	0.139	0.500	0.000	0.500	21.5	2.60	7.10	8479.5	36.0	7582.4	21.4	181.8	225.3	
	AgentScope	GPT-5.4	59.0	10.5	119.8	131.4	9.20	0.156	0.800	0.000	0.200	11.6	0.800	1.90	9488.2	41.8	7238.1	25.0	301.7	423.9	
	AgentScope	Gemini-3-Flash	59.7	0.585	138.2	138.2	10.0	0.168	1.00	0.000	0.000	0.000	0.000	0.000	9671.7	50.4	9544.2	27.0	199.0	240.0	
	AgentScope	Qwen-3.5	58.8	15.3	109.0	121.6	8.40	0.141	0.700	0.000	0.300	12.7	1.60	4.30	9469.8	44.6	9577.6	27.9	192.1	237.3	
Γ^{hol}	LangGraph	GPT-5.4	30.6	13.8	20.0	42.4	4.00	0.126	0.100	0.000	0.900	22.4	6.00	19.5	4988.9	7.50	992.5	4.06	152.9	106.8	
	LangGraph	Gemini-3-Flash	59.7	0.623	137.8	137.8	10.0	0.168	1.00	0.000	0.000	0.000	0.000	0.000	9671.2	50.4	9592.0	27.0	199.0	120.0	
	LangGraph	Qwen-3.5	17.6	8.18	1.54	10.3	1.70	0.073	0.000	0.000	1.00	8.77	8.30	28.3	2876.3	1.00	17.2	0.416	79.7	50.8	
	AgentScope	GPT-5.4	29.7	5.47	7.71	43.0	3.90	0.129	0.000	0.000	1.00	35.3	6.10	19.1	4849.2	4.00	105.5	1.14	151.3	104.4	
	AgentScope	Gemini-3-Flash	59.7	0.623	137.8	137.8	10.0	0.168	1.00	0.000	0.000	0.000	0.000	0.000	9671.2	50.4	9592.0	27.0	199.0	120.0	
	AgentScope	Qwen-3.5	18.9	7.61	1.84	11.2	2.00	0.086	0.000	0.000	1.00	9.34	8.00	27.2	3083.0	1.00	9.62	0.360	85.9	54.8	

Table 22: Logical full-metric matrix for A5C12-2 (Panel B).

Paradigm Framework LLM			Panel B: Constraint / Resource / Process																			
			G_{co}	G_{vm}	G_{com}	G_{cod}	G_{cof}	G_{cof}	G_{st}	G_{ns}	G_{tt}	G_{te}	G_{um}	G_{ic}	G_{pt}	G_{bw}	G_{ae}	G_{af}	G_{ab}	G_{pq}	G_{dp}	G_{tr}
Γ^{cen}	LangGraph	GPT-5.4	7343.2	21527.1	8072.7	9.38	16.3	16.3	8.94	53.6	68.6	13.7	40.0	40.0	0.000	0.000	4.00	7.00	1.75	0.300	0.136	0.136
Γ^{cen}	LangGraph	Gemini-3-Flash	6871.9	17479.8	6554.9	4.82	16.1	16.1	8.09	48.5	67.9	13.6	40.0	40.0	0.000	0.000	4.00	7.00	1.75	0.000	0.000	0.000
Γ^{cen}	LangGraph	Qwen-3.5	6975.3	18514.0	6942.8	6.06	16.0	16.0	10.0	60.1	67.9	13.6	40.0	40.0	0.000	0.000	4.00	7.00	1.75	0.200	0.432	0.432
Γ^{cen}	AgentScope	GPT-5.4	6433.3	15859.6	5947.4	7.62	13.2	13.2	7.41	44.4	60.8	12.2	35.5	35.5	0.000	0.000	4.00	7.00	1.75	0.100	1.06	1.06
Γ^{cen}	AgentScope	Gemini-3-Flash	6871.9	17479.8	6554.9	4.82	16.1	16.1	8.09	48.5	67.9	13.6	40.0	40.0	0.000	0.000	4.00	7.00	1.75	0.000	0.000	0.000
Γ^{cen}	AgentScope	Qwen-3.5	7044.7	18698.4	7011.9	6.95	16.0	16.0	9.63	57.8	67.8	13.6	40.0	40.0	0.000	0.000	4.00	7.00	1.75	0.200	0.157	0.157
Γ^{hie}	LangGraph	GPT-5.4	6382.5	20366.8	7637.5	34.7	26.8	26.8	13.0	77.7	69.4	13.9	38.9	38.9	0.000	0.000	4.00	7.00	1.75	0.800	2.78	2.78
Γ^{hie}	LangGraph	Gemini-3-Flash	5813.7	15347.8	5755.4	27.2	25.1	25.1	9.67	58.0	71.3	14.3	40.0	40.0	0.000	0.000	4.00	7.00	1.75	1.00	1.83	1.83
Γ^{hie}	LangGraph	Qwen-3.5	4804.6	11977.3	4491.5	19.3	16.1	16.1	8.76	52.6	59.0	11.8	33.1	33.1	0.000	0.000	4.00	7.00	1.75	0.700	2.58	2.58
Γ^{hie}	AgentScope	GPT-5.4	5881.7	18097.3	6786.5	30.7	24.2	24.2	10.5	62.7	66.0	13.2	37.0	37.0	0.000	0.000	4.00	7.00	1.75	0.800	2.00	2.00
Γ^{hie}	AgentScope	Gemini-3-Flash	5802.7	15347.1	5755.2	27.0	25.2	25.2	9.66	58.0	71.2	14.2	40.0	40.0	0.000	0.000	4.00	7.00	1.75	1.00	2.37	2.37
Γ^{hie}	AgentScope	Qwen-3.5	5194.3	13989.4	5246.0	23.9	18.5	18.5	8.68	52.1	60.9	12.2	34.2	34.2	0.000	0.000	4.00	7.00	1.75	1.00	2.84	2.84
Γ^{het}	LangGraph	GPT-5.4	6067.1	13149.6	4931.1	29.9	26.7	26.7	7.77	46.6	70.7	14.1	40.0	40.0	0.000	0.000	4.00	7.00	1.75	1.00	3.57	3.57
Γ^{het}	LangGraph	Gemini-3-Flash	5803.0	15270.8	5726.5	27.0	25.2	25.2	9.75	58.5	71.2	14.2	40.0	40.0	0.000	0.000	4.00	7.00	1.75	1.00	2.41	2.41
Γ^{het}	LangGraph	Qwen-3.5	5087.7	12131.8	4549.4	21.4	18.0	18.0	8.99	53.9	62.2	12.4	34.9	34.9	0.000	0.000	4.00	7.00	1.75	0.500	1.26	1.26
Γ^{het}	AgentScope	GPT-5.4	5692.9	11580.9	4342.8	25.0	20.9	20.9	8.37	50.2	68.5	13.7	38.7	38.7	0.000	0.000	4.00	7.00	1.75	0.800	2.72	2.72
Γ^{het}	AgentScope	Gemini-3-Flash	5803.0	15270.8	5726.5	27.0	25.2	25.2	9.75	58.5	71.2	14.2	40.0	40.0	0.000	0.000	4.00	7.00	1.75	1.00	2.41	2.41
Γ^{het}	AgentScope	Qwen-3.5	5681.9	15324.1	5746.5	27.9	22.3	22.3	10.8	64.7	65.8	13.2	36.9	36.9	0.000	0.000	4.00	7.00	1.75	0.700	2.15	2.15
Γ^{hol}	LangGraph	GPT-5.4	2993.3	1588.1	595.5	4.06	3.70	3.70	3.15	18.9	42.8	8.56	24.1	24.1	0.000	0.000	4.00	7.00	1.75	0.400	0.727	0.727
Γ^{hol}	LangGraph	Gemini-3-Flash	5802.7	15347.1	5755.2	27.0	25.2	25.2	9.66	58.0	71.2	14.2	40.0	40.0	0.000	0.000	4.00	7.00	1.75	1.00	2.37	2.37
Γ^{hol}	LangGraph	Qwen-3.5	1725.8	27.6	10.3	0.416	0.500	0.500	1.02	6.11	27.7	5.53	15.7	15.7	0.000	0.000	4.00	7.00	1.75	0.200	0.381	0.381
Γ^{hol}	AgentScope	GPT-5.4	2909.5	168.8	63.3	1.14	2.00	2.00	3.04	18.3	44.0	8.81	24.8	24.8	0.000	0.000	4.00	7.00	1.75	0.400	0.934	0.934
Γ^{hol}	AgentScope	Gemini-3-Flash	5802.7	15347.1	5755.2	27.0	25.2	25.2	9.66	58.0	71.2	14.2	40.0	40.0	0.000	0.000	4.00	7.00	1.75	1.00	2.37	2.37
Γ^{hol}	AgentScope	Qwen-3.5	1849.8	15.4	5.77	0.360	0.500	0.500	1.24	7.46	29.6	5.93	16.8	16.8	0.000	0.000	4.00	7.00	1.75	0.300	0.388	0.388

Table 23: Logical full-metric matrix for A5C12-2 (Panel C).

Paradigm Framework LLM		Panel C: Coordination / Audit / Diagnostics																				
		G^{vb}	G^{bc}	G^{pa}	G^{ac}	G^{dw}	G^{ba}	G^{am}	G^{aa}	G^{ca}	G^{as}	G^{wc}	G^{ar}	G^{sv}	G^{sc}	G^{co}	G^{uc}	G^{lm}	G^{fb}	G^{na}	G^{pd}	
Γ^{cen}	LangGraph	GPT-5.4	2.12	40.0	0.000	0.000	2.02	0.652	0.000	0.000	0.700	120.0	155.6	120.0	160.0	10.0	30.0	0.000	120.0	0.000	0.000	1.00
	LangGraph	Gemini-3-Flash	2.00	40.0	0.000	0.000	2.02	0.560	0.000	0.000	0.700	120.0	243.8	120.0	160.0	10.0	30.0	0.000	120.0	0.000	0.000	-
	LangGraph	Qwen-3.5	2.04	40.0	0.000	0.000	2.02	0.602	0.000	0.000	0.698	120.0	204.5	120.0	160.0	10.0	30.0	0.000	120.0	0.000	0.000	1.00
	AgentScope	GPT-5.4	2.15	35.5	0.000	0.000	2.04	0.654	0.000	0.000	0.710	106.2	354.1	106.2	140.4	8.30	27.2	0.000	106.1	0.100	0.000	0.999
	AgentScope	Gemini-3-Flash	2.00	40.0	0.000	0.000	2.02	0.560	0.000	0.000	0.700	120.0	342.6	120.0	160.0	10.0	30.0	0.000	120.0	0.000	0.000	-
	AgentScope	Qwen-3.5	2.08	40.0	0.000	0.000	2.02	0.622	0.000	0.000	0.698	120.0	311.6	120.0	160.0	10.0	30.0	0.000	120.0	0.000	0.000	1.00
Γ^{hie}	LangGraph	GPT-5.4	1.41	38.9	3.87	1.38	2.00	0.204	1.00	0.377	0.687	194.3	261.1	194.3	154.8	9.40	29.5	0.000	194.2	0.100	0.000	0.999
	LangGraph	Gemini-3-Flash	1.40	40.0	3.97	1.50	2.02	0.200	1.00	0.472	0.698	200.0	397.5	200.0	160.0	10.0	30.0	0.000	199.4	0.600	0.000	-
	LangGraph	Qwen-3.5	1.47	33.1	3.77	1.36	1.99	0.227	1.00	0.357	0.678	164.6	264.6	164.6	128.4	6.80	26.3	0.000	164.6	0.000	0.000	1.00
	AgentScope	GPT-5.4	1.44	37.0	3.90	1.40	2.02	0.217	1.00	0.399	0.705	184.7	525.7	184.7	146.4	8.90	28.1	0.000	184.6	0.100	0.000	0.999
	AgentScope	Gemini-3-Flash	1.40	40.0	4.00	1.51	2.02	0.200	1.00	0.477	0.698	200.0	507.7	200.0	160.0	10.0	30.0	0.000	199.3	0.700	0.000	-
	AgentScope	Qwen-3.5	1.46	34.2	3.81	1.35	1.98	0.223	1.00	0.340	0.666	170.4	431.0	170.4	133.6	7.40	26.8	0.000	170.4	0.000	0.000	1.00
Γ^{het}	LangGraph	GPT-5.4	1.40	40.0	3.98	1.41	2.01	0.200	1.00	0.382	0.693	336.4	846.2	336.4	228.2	10.0	30.0	0.000	336.4	0.000	0.000	1.00
	LangGraph	Gemini-3-Flash	1.40	40.0	4.00	1.51	2.02	0.200	1.00	0.477	0.698	200.0	444.5	200.0	160.0	10.0	30.0	0.000	199.8	0.200	0.000	-
	LangGraph	Qwen-3.5	1.47	34.9	3.80	1.32	1.98	0.231	1.00	0.318	0.672	182.9	298.2	182.9	140.2	7.40	27.5	0.000	182.9	0.000	0.000	1.00
	AgentScope	GPT-5.4	1.42	38.7	3.99	1.40	2.02	0.211	1.00	0.378	0.702	302.9	1116.7	302.9	208.4	9.20	29.5	0.000	297.4	5.50	0.000	0.982
	AgentScope	Gemini-3-Flash	1.40	40.0	4.00	1.51	2.02	0.200	1.00	0.477	0.698	200.0	388.4	200.0	160.0	10.0	30.0	0.000	199.8	0.200	0.000	-
	AgentScope	Qwen-3.5	1.44	36.9	3.79	1.34	1.99	0.217	1.00	0.343	0.672	193.1	406.0	193.1	149.8	8.40	28.5	0.000	193.1	0.000	0.000	1.00
Γ^{hol}	LangGraph	GPT-5.4	1.58	24.1	3.97	1.36	2.05	0.274	1.00	0.363	0.690	153.9	374.9	153.9	110.0	4.00	20.1	0.000	153.9	0.000	0.000	1.00
	LangGraph	Gemini-3-Flash	1.40	40.0	4.00	1.51	2.02	0.200	1.00	0.477	0.698	200.0	494.8	200.0	160.0	10.0	30.0	0.000	199.2	0.800	0.000	-
	LangGraph	Qwen-3.5	1.55	15.7	3.91	1.33	1.88	0.311	1.00	0.318	0.613	80.7	133.3	80.7	57.8	1.70	14.0	0.000	80.7	0.000	0.000	1.00
	AgentScope	GPT-5.4	1.60	24.8	3.97	1.34	2.05	0.290	1.00	0.340	0.693	152.6	450.3	152.6	109.6	3.90	20.9	0.000	152.6	0.000	0.000	1.00
	AgentScope	Gemini-3-Flash	1.40	40.0	4.00	1.51	2.02	0.200	1.00	0.477	0.698	200.0	377.5	200.0	160.0	10.0	30.0	0.000	199.2	0.800	0.000	-
	AgentScope	Qwen-3.5	1.57	16.8	3.89	1.33	1.90	0.308	1.00	0.326	0.603	86.9	173.7	86.9	62.8	2.00	14.8	0.000	86.9	0.000	0.000	1.00

Table 24: Primary 15-metric matrix for A5C12-3.

Paradigm	Framework	LLM	Evaluation Metrics														
			$\mathcal{G}_{\text{eff}}$			$\mathcal{G}_{\text{con}}$				$\mathcal{G}_{\text{coord}}$				$\mathcal{G}_{\text{rob}}$			
			$G_{\text{mk}} \downarrow$	$G_{\text{td}} \downarrow$	$G_{\text{cj}} \uparrow$	$G_{\text{vc}} \downarrow$	$G_{\text{eo}} \downarrow$	$G_{\text{ed}} \downarrow$	$G_{\text{en}} \downarrow$	$G_{\text{ds}} \downarrow$	$G_{\text{cm}} \downarrow$	$G_{\text{fb}} \downarrow$	$G_{\text{llm}} \downarrow$	$G_{\text{tr}} \downarrow$	$G_{\text{dj}} \downarrow$	$G_{\text{ut}} \downarrow$	$G_{\text{do}} \downarrow$
Γ^{cen}	LangGraph	GPT-5.4	78.2	258.0	9.3	32.2	17323.2	9.0	12241.0	114.0	114.0	0.1	113.9	0.10	0.7	5.2	2.3
		Gemini-3-Flash	79.3	265.0	10.0	34.8	14527.0	8.3	12412.2	120.0	120.0	1.1	118.9	0.00	0.0	0.0	0.0
		Qwen-3.5	80.1	272.0	10.0	34.6	15954.8	9.2	12550.1	120.0	120.0	0.0	120.0	0.00	0.0	0.0	0.0
	AgentScope	GPT-5.4	83.0	284.5	10.0	35.8	18810.1	9.9	12972.1	120.0	120.0	0.0	120.0	0.00	0.0	0.0	0.0
		Gemini-3-Flash	79.3	265.0	10.0	34.8	14527.0	8.3	12412.2	120.0	120.0	1.1	118.9	0.00	0.0	0.0	0.0
		Qwen-3.5	81.4	275.1	10.0	34.8	16556.1	10.3	12747.2	120.0	120.0	0.0	120.0	0.00	0.0	0.0	0.0
Γ^{hie}	LangGraph	GPT-5.4	67.6	160.3	9.3	58.4	17895.0	37.6	10835.1	188.2	113.6	0.3	189.0	0.10	0.7	1.3	2.5
		Gemini-3-Flash	65.4	161.1	9.5	64.0	16809.9	34.4	10552.7	195.3	117.8	0.1	196.2	0.10	0.5	7.8	1.1
		Qwen-3.5	68.8	144.1	8.8	57.4	17500.5	38.7	10989.9	183.9	111.4	0.0	185.6	0.20	1.2	9.1	3.6
	AgentScope	GPT-5.4	65.9	151.5	8.6	58.2	17205.6	35.3	10572.3	181.5	109.6	0.0	182.5	0.20	1.4	14.2	4.2
		Gemini-3-Flash	65.4	161.1	9.5	64.0	16809.9	34.4	10552.7	195.3	117.8	0.1	196.2	0.10	0.5	7.8	1.1
		Qwen-3.5	67.9	153.6	9.0	60.6	18322.7	37.7	10877.4	188.0	113.9	0.0	189.7	0.20	1.0	8.4	2.8
Γ^{het}	LangGraph	GPT-5.4	65.5	169.8	10.0	63.8	12693.3	31.6	10497.6	339.0	485.0	0.0	340.0	0.00	0.0	0.0	0.0
		Gemini-3-Flash	66.8	173.4	10.0	65.2	17097.0	35.2	10780.8	207.3	254.7	0.2	208.2	0.00	0.0	0.0	0.0
		Qwen-3.5	64.6	134.1	8.4	56.6	16081.9	34.3	10363.1	192.4	238.4	0.0	194.1	0.30	1.6	19.8	4.1
	AgentScope	GPT-5.4	66.0	162.6	10.0	61.4	11518.8	32.4	10571.3	347.4	499.7	0.0	348.4	0.00	0.0	0.0	0.0
		Gemini-3-Flash	66.4	173.2	10.0	66.6	17196.8	35.0	10719.6	207.4	254.7	0.2	208.2	0.00	0.0	0.0	0.0
		Qwen-3.5	78.9	183.5	10.0	71.2	23338.0	48.9	12572.4	203.9	249.8	0.0	205.6	0.00	0.0	0.0	0.0
Γ^{hol}	LangGraph	GPT-5.4	22.0	5.3	1.7	2.4	248.4	0.9	3564.5	109.6	76.9	0.0	110.6	1.00	8.3	18.1	26.5
		Gemini-3-Flash	58.8	134.3	8.5	55.2	13236.6	28.2	9494.1	193.5	121.1	0.1	194.4	0.30	1.5	14.9	4.1
		Qwen-3.5	23.7	22.1	2.2	10.8	2172.6	6.3	3814.8	84.7	53.7	0.0	86.0	0.90	7.8	11.6	26.7
	AgentScope	GPT-5.4	27.4	4.0	2.8	3.6	109.0	1.2	4450.2	132.8	92.5	0.0	133.8	1.00	7.2	31.9	22.5
		Gemini-3-Flash	58.8	134.3	8.5	55.2	13236.6	28.2	9494.1	193.5	121.1	0.1	194.4	0.30	1.5	14.9	4.1
		Qwen-3.5	41.4	72.2	5.2	28.2	8013.1	17.7	6644.1	130.6	81.8	0.0	132.1	0.60	4.8	5.3	16.7

F.5 Read the Comparison Surface

The resulting runs should be inspected first through the 15 primary metrics in Tables 12-27, which form the main comparison surface. These metrics cover effectiveness, constraint alignment, coordination efficiency, and robustness, and no single scalar score is sufficient to establish superiority. Each run also produces a standardized artifact set (Table 29) to support reconstruction and diagnosis. We encourage users to design adaptive coordination strategies that dynamically adjust between local and global behaviors over time, aiming to achieve strong performance across most metric families.

Table 25: Logical full-metric matrix for A5C12-3 (Panel A).

Paradigm Framework LLM		Panel A: Primary Outcomes + Robustness Anchors																		
		G_{mk}^a	G_{mk}^b	G_{td}^a	G_{td}^b	G_{cf}^a	G_{th}^a	G_{st}^a	G_{lp}^a	G_{tr}^a	G_{fp}^a	G_{op}^a	G_{ua}^a	G_{va}^a	G_{oa}^a	G_{pa}^a	G_{sp}^a	G_{uc}^a		
Γ^{cen}	LangGraph	GPT-5.4	78.2	15.0	258.0	263.2	9.30	0.117	0.900	0.000	0.100	5.21	0.700	2.30	12241.0	32.2	17323.2	9.04	114.0	114.0
	LangGraph	Gemini-3-Flash	79.3	4.78	265.0	265.0	10.0	0.127	1.00	0.000	0.000	0.000	0.000	0.000	12412.2	34.8	14527.0	8.28	120.0	120.0
	LangGraph	Qwen-3.5	80.1	4.20	272.0	272.0	10.0	0.125	1.00	0.000	0.000	0.000	0.000	0.000	12550.1	34.6	15954.8	9.16	120.0	120.0
	AgentScope	GPT-5.4	83.0	3.77	284.5	284.5	10.0	0.121	1.00	0.000	0.000	0.000	0.000	0.000	12972.1	35.8	18810.1	9.88	120.0	120.0
	AgentScope	Gemini-3-Flash	79.3	4.78	265.0	265.0	10.0	0.127	1.00	0.000	0.000	0.000	0.000	0.000	12412.2	34.8	14527.0	8.28	120.0	120.0
	AgentScope	Qwen-3.5	81.4	4.22	275.1	275.1	10.0	0.123	1.00	0.000	0.000	0.000	0.000	0.000	12747.2	34.8	16556.1	10.3	120.0	120.0
Γ^{hie}	LangGraph	GPT-5.4	67.6	16.3	160.3	161.6	9.30	0.138	0.900	0.000	0.100	1.28	0.700	2.50	10835.1	58.4	17895.0	37.6	188.2	113.6
	LangGraph	Gemini-3-Flash	65.4	7.48	161.1	169.0	9.50	0.144	0.900	0.000	0.100	7.83	0.500	1.10	10552.7	64.0	16809.9	34.4	195.3	117.8
	LangGraph	Qwen-3.5	68.8	17.8	144.1	153.2	8.80	0.127	0.800	0.000	0.200	9.07	1.20	3.60	10989.9	57.4	17500.5	38.7	183.9	111.4
	AgentScope	GPT-5.4	65.9	16.1	151.5	165.6	8.60	0.126	0.800	0.000	0.200	14.2	1.40	4.20	10572.3	58.2	17205.6	35.3	181.5	109.6
	AgentScope	Gemini-3-Flash	65.4	7.48	161.1	169.0	9.50	0.144	0.900	0.000	0.100	7.83	0.500	1.10	10552.7	64.0	16809.9	34.4	195.3	117.8
	AgentScope	Qwen-3.5	67.9	15.2	153.6	162.0	9.00	0.133	0.800	0.000	0.200	8.43	1.00	2.80	10877.4	60.6	18322.7	37.7	188.0	113.9
Γ^{het}	LangGraph	GPT-5.4	65.5	4.68	169.8	169.8	10.0	0.154	1.00	0.000	0.000	0.000	0.000	0.000	10497.6	63.8	12693.3	31.6	339.0	485.0
	LangGraph	Gemini-3-Flash	66.8	2.62	173.4	173.4	10.0	0.150	1.00	0.000	0.000	0.000	0.000	0.000	10780.8	65.2	17097.0	35.2	207.3	254.7
	LangGraph	Qwen-3.5	64.6	15.7	134.1	153.9	8.40	0.128	0.700	0.000	0.300	19.8	1.60	4.10	10363.1	56.6	16081.9	34.3	192.4	238.4
	AgentScope	GPT-5.4	66.0	5.35	162.6	162.6	10.0	0.153	1.00	0.000	0.000	0.000	0.000	0.000	10571.3	61.4	11518.8	32.4	347.4	499.7
	AgentScope	Gemini-3-Flash	66.4	3.00	173.2	173.2	10.0	0.151	1.00	0.000	0.000	0.000	0.000	0.000	10719.6	66.6	17196.8	35.0	207.4	254.7
	AgentScope	Qwen-3.5	78.9	4.70	183.5	183.5	10.0	0.127	1.00	0.000	0.000	0.000	0.000	0.000	12572.4	71.2	23338.0	48.9	203.9	249.8
Γ^{hol}	LangGraph	GPT-5.4	22.0	10.3	5.25	23.3	1.70	0.054	0.000	0.000	1.00	18.1	8.30	26.5	3564.5	2.40	248.4	0.905	109.6	76.9
	LangGraph	Gemini-3-Flash	58.8	13.8	134.3	149.2	8.50	0.140	0.700	0.000	0.300	14.9	1.50	4.10	9494.1	55.2	13236.6	28.2	193.5	121.1
	LangGraph	Qwen-3.5	23.7	22.0	22.1	33.8	2.20	0.049	0.100	0.000	0.900	11.6	7.80	26.7	3814.8	10.8	2172.6	6.27	84.7	53.7
	AgentScope	GPT-5.4	27.4	8.49	3.98	35.9	2.80	0.091	0.000	0.000	1.00	31.9	7.20	22.5	4450.2	3.60	109.0	1.15	132.8	92.5
	AgentScope	Gemini-3-Flash	58.8	13.8	134.3	149.2	8.50	0.140	0.700	0.000	0.300	14.9	1.50	4.10	9494.1	55.2	13236.6	28.2	193.5	121.1
	AgentScope	Qwen-3.5	41.4	27.7	72.2	77.5	5.20	0.103	0.400	0.000	0.600	5.31	4.80	16.7	6644.1	28.2	8013.1	17.7	130.6	81.8

Table 26: Logical full-metric matrix for A5C12-3 (Panel B).

Paradigm Framework LLM			Panel B: Constraint / Resource / Process																			
			$\mathcal{G}^{co}$	$\mathcal{G}^{vm}$	$\mathcal{G}^{com}$	$\mathcal{G}^{cod}$	$\mathcal{G}^{eof}$	$\mathcal{G}^{col}$	$\mathcal{G}^{st}$	$\mathcal{G}^{ns}$	$\mathcal{G}^{tt}$	$\mathcal{G}^{et}$	$\mathcal{G}^{wt}$	$\mathcal{G}^{ae}$	$\mathcal{G}^{af}$	$\mathcal{G}^{ab}$	$\mathcal{G}^{pd}$	$\mathcal{G}^{tp}$	$\mathcal{G}^{tr}$			
Γ^{cen}	LangGraph	GPT-5.4	7344.6	27717.1	10393.9	9.04	16.1	16.1	8.90	53.4	68.6	13.7	38.1	38.1	0.000	0.000	4.00	7.00	1.75	1.00	2.26	2.26
	LangGraph	Gemini-3-Flash	7447.3	23243.1	8716.2	8.28	17.4	17.4	8.68	52.1	70.1	14.0	40.0	40.0	0.000	0.000	4.00	7.00	1.75	0.500	2.62	2.62
	LangGraph	Qwen-3.5	7530.1	25527.6	9572.9	9.16	17.3	17.3	8.85	53.1	70.3	14.1	40.0	40.0	0.000	0.000	4.00	7.00	1.75	0.500	2.37	2.37
	AgentScope	GPT-5.4	7783.3	30096.2	11286.1	9.88	17.9	17.9	9.30	55.8	72.0	14.4	40.0	40.0	0.000	0.000	4.00	7.00	1.75	0.500	1.75	1.75
	AgentScope	Gemini-3-Flash	7447.3	23243.1	8716.2	8.28	17.4	17.4	8.68	52.1	70.1	14.0	40.0	40.0	0.000	0.000	4.00	7.00	1.75	0.500	2.62	2.62
	AgentScope	Qwen-3.5	7648.3	26489.8	9933.7	10.3	17.4	17.4	10.2	61.4	70.3	14.1	40.0	40.0	0.000	0.000	4.00	7.00	1.75	0.800	4.32	4.32
Γ^{hie}	LangGraph	GPT-5.4	6501.1	28632.0	10737.0	37.6	29.2	29.2	10.9	65.4	70.4	14.1	37.9	37.9	0.000	0.000	4.00	7.00	1.75	5.30	13.6	13.6
	LangGraph	Gemini-3-Flash	6331.6	26895.8	10085.9	34.4	32.0	32.0	9.60	57.6	72.9	14.6	39.3	39.3	0.000	0.000	4.00	7.00	1.75	6.00	17.9	17.9
	LangGraph	Qwen-3.5	6593.9	28000.8	10500.3	38.7	28.7	28.7	13.2	79.2	68.9	13.8	37.2	37.2	0.000	0.000	4.00	7.00	1.75	5.70	16.2	16.2
	AgentScope	GPT-5.4	6343.4	27529.0	10323.4	35.3	29.1	29.1	11.8	71.0	68.0	13.6	36.6	36.6	0.000	0.000	4.00	7.00	1.75	5.80	18.1	18.1
	AgentScope	Gemini-3-Flash	6331.6	26895.8	10085.9	34.4	32.0	32.0	9.60	57.6	72.9	14.6	39.3	39.3	0.000	0.000	4.00	7.00	1.75	6.00	17.9	17.9
	AgentScope	Qwen-3.5	6526.4	29316.4	10993.6	37.7	30.3	30.3	11.8	70.6	70.4	14.1	38.0	38.0	0.000	0.000	4.00	7.00	1.75	5.80	17.5	17.5
Γ^{het}	LangGraph	GPT-5.4	6298.5	20309.3	7616.0	31.6	31.9	31.9	8.71	52.2	73.6	14.7	40.0	40.0	0.000	0.000	4.00	7.00	1.75	5.30	16.6	16.6
	LangGraph	Gemini-3-Flash	6468.5	27355.1	10258.2	35.2	32.6	32.6	10.3	61.5	74.3	14.9	40.0	40.0	0.000	0.000	4.00	7.00	1.75	5.70	17.6	17.6
	LangGraph	Qwen-3.5	6217.9	25731.0	9649.1	34.3	28.3	28.3	11.6	69.7	68.7	13.7	37.1	37.1	0.000	0.000	4.00	7.00	1.75	6.00	17.1	17.1
	AgentScope	GPT-5.4	6342.8	18430.1	6911.3	32.4	30.7	30.7	8.43	50.6	73.6	14.7	40.0	40.0	0.000	0.000	4.00	7.00	1.75	5.30	13.7	13.7
	AgentScope	Gemini-3-Flash	6431.7	27514.9	10318.1	35.0	33.3	33.3	10.3	61.9	74.3	14.9	40.0	40.0	0.000	0.000	4.00	7.00	1.75	5.80	16.5	16.5
	AgentScope	Qwen-3.5	7543.5	37340.7	14002.8	48.9	35.6	35.6	13.6	81.4	74.2	14.8	40.0	40.0	0.000	0.000	4.00	7.00	1.75	6.30	18.5	18.5
Γ^{hol}	LangGraph	GPT-5.4	2138.7	397.5	149.1	0.905	1.20	1.20	1.52	9.11	32.2	6.44	17.5	17.5	0.000	0.000	4.00	7.00	1.75	2.20	4.94	4.94
	LangGraph	Gemini-3-Flash	5696.5	21178.5	7941.9	28.2	27.6	27.6	8.55	51.3	68.4	13.7	36.9	36.9	0.000	0.000	4.00	7.00	1.75	5.50	16.1	16.1
	LangGraph	Qwen-3.5	2288.9	3476.2	1303.6	6.27	5.40	5.40	2.58	15.5	31.1	6.21	16.9	16.9	0.000	0.000	4.00	7.00	1.75	2.10	4.46	4.46
	AgentScope	GPT-5.4	2670.1	174.4	65.4	1.15	1.80	1.80	2.23	13.4	39.7	7.94	21.5	21.5	0.000	0.000	4.00	7.00	1.75	2.60	5.31	5.31
	AgentScope	Gemini-3-Flash	5696.5	21178.5	7941.9	28.2	27.6	27.6	8.55	51.3	68.4	13.7	36.9	36.9	0.000	0.000	4.00	7.00	1.75	5.50	16.1	16.1
	AgentScope	Qwen-3.5	3986.5	12820.9	4807.8	17.7	14.1	14.1	5.20	31.2	47.5	9.50	25.7	25.7	0.000	0.000	4.00	7.00	1.75	3.30	9.20	9.20

Table 27: Logical full-metric matrix for A5C12-3 (Panel C)

Paradigm Framework LLM		Panel C: Coordination / Audit / Diagnostics																				
		G^{vb}	G^{bc}	G^{pa}	G^{ac}	G^{dw}	G^{ba}	G^{amb}	G^{aa}	G^{ca}	G^{as}	G^{wc}	G^{ar}	G^{sv}	G^{sc}	G^{oc}	G^{uc}	G^{lm}	G^{fb}	G^{na}	G^{pl}	
Γ^{cen}	LangGraph	GPT-5.4	2.07	38.1	0.000	0.000	2.02	0.601	0.000	0.000	0.702	114.0	158.1	114.0	151.6	9.30	28.8	0.000	113.9	0.100	0.000	0.999
Γ^{cen}	LangGraph	Gemini-3-Flash	1.97	40.0	0.000	0.000	2.02	0.530	0.000	0.000	0.700	120.0	250.7	120.0	160.0	10.0	30.0	0.000	118.9	1.10	0.000	-
Γ^{cen}	LangGraph	Qwen-3.5	1.97	40.0	0.000	0.000	2.02	0.565	0.000	0.000	0.700	120.0	201.2	120.0	160.0	10.0	30.0	0.000	120.0	0.000	0.000	1.00
Γ^{cen}	AgentScope	GPT-5.4	2.01	40.0	0.000	0.000	2.02	0.570	0.000	0.000	0.700	120.0	294.7	120.0	160.0	10.0	30.0	0.000	120.0	0.000	0.000	1.00
Γ^{cen}	AgentScope	Gemini-3-Flash	1.97	40.0	0.000	0.000	2.02	0.530	0.000	0.000	0.700	120.0	343.9	120.0	160.0	10.0	30.0	0.000	118.9	1.10	0.000	-
Γ^{cen}	AgentScope	Qwen-3.5	1.97	40.0	0.000	0.000	2.02	0.550	0.000	0.000	0.700	120.0	296.1	120.0	160.0	10.0	30.0	0.000	120.0	0.000	0.000	1.00
Γ^{hie}	LangGraph	GPT-5.4	1.43	37.9	3.90	1.41	2.01	0.214	1.00	0.410	0.701	189.3	269.3	189.3	150.8	9.30	28.6	0.000	189.0	0.300	0.000	0.998
Γ^{hie}	LangGraph	Gemini-3-Flash	1.41	39.3	3.95	1.46	2.01	0.204	1.00	0.442	0.698	196.3	388.5	196.3	156.4	9.50	29.8	0.000	196.2	0.100	0.000	-
Γ^{hie}	LangGraph	Qwen-3.5	1.45	37.2	3.82	1.36	1.99	0.233	1.00	0.347	0.693	185.6	300.1	185.6	147.2	8.80	28.4	0.000	185.6	0.000	0.000	1.00
Γ^{hie}	AgentScope	GPT-5.4	1.43	36.6	3.90	1.40	2.02	0.212	1.00	0.393	0.712	182.5	405.0	182.5	144.8	8.60	28.0	0.000	182.5	0.000	0.000	1.00
Γ^{hie}	AgentScope	Gemini-3-Flash	1.41	39.3	3.95	1.46	2.01	0.204	1.00	0.442	0.698	196.3	491.0	196.3	156.4	9.50	29.8	0.000	196.2	0.100	0.000	-
Γ^{hie}	AgentScope	Qwen-3.5	1.45	38.0	3.85	1.39	1.98	0.232	1.00	0.376	0.679	189.7	464.8	189.7	150.4	9.00	29.0	0.000	189.7	0.000	0.000	1.00
Γ^{het}	LangGraph	GPT-5.4	1.40	40.0	4.01	1.39	2.01	0.200	1.00	0.359	0.695	340.0	842.7	340.0	230.0	10.0	30.0	0.000	340.0	0.000	0.000	1.00
Γ^{het}	LangGraph	Gemini-3-Flash	1.40	40.0	3.97	1.48	2.01	0.203	1.00	0.444	0.698	208.4	464.2	208.4	164.2	10.0	30.0	0.000	208.2	0.200	0.000	-
Γ^{het}	LangGraph	Qwen-3.5	1.46	37.1	3.89	1.36	2.02	0.237	1.00	0.346	0.700	194.1	321.3	194.1	150.6	8.40	28.7	0.000	194.1	0.000	0.000	1.00
Γ^{het}	AgentScope	GPT-5.4	1.40	40.0	4.00	1.37	2.02	0.200	1.00	0.352	0.698	348.4	929.8	348.4	234.2	10.0	30.0	0.000	348.4	0.000	0.000	1.00
Γ^{het}	AgentScope	Gemini-3-Flash	1.40	40.0	3.97	1.48	2.00	0.200	1.00	0.447	0.698	208.4	403.2	208.4	164.2	10.0	30.0	0.000	208.2	0.200	0.000	-
Γ^{het}	AgentScope	Qwen-3.5	1.42	40.0	3.86	1.35	1.97	0.217	1.00	0.339	0.685	205.6	349.7	205.6	162.8	10.0	30.0	0.000	205.6	0.000	0.000	1.00
Γ^{hol}	LangGraph	GPT-5.4	1.45	17.5	4.04	1.33	1.93	0.263	1.00	0.333	0.650	110.6	259.6	110.6	77.0	1.70	15.8	0.000	110.6	0.000	0.000	1.00
Γ^{hol}	LangGraph	Gemini-3-Flash	1.42	36.9	3.99	1.47	2.02	0.212	1.00	0.452	0.703	194.5	481.2	194.5	151.6	8.50	28.4	0.000	194.4	0.100	0.000	-
Γ^{hol}	LangGraph	Qwen-3.5	1.55	16.9	3.97	1.34	1.78	0.371	1.00	0.340	0.548	86.0	140.9	86.0	63.2	2.20	14.7	0.000	86.0	0.000	0.000	1.00
Γ^{hol}	AgentScope	GPT-5.4	1.54	21.5	4.08	1.34	2.01	0.277	1.00	0.335	0.692	133.8	385.4	133.8	94.8	2.80	18.7	0.000	133.8	0.000	0.000	1.00
Γ^{hol}	AgentScope	Gemini-3-Flash	1.42	36.9	3.99	1.47	2.02	0.212	1.00	0.452	0.703	194.5	364.2	194.5	151.6	8.50	28.4	0.000	194.4	0.100	0.000	-
Γ^{hol}	AgentScope	Qwen-3.5	1.56	25.7	3.85	1.37	1.93	0.313	1.00	0.361	0.641	132.1	240.6	132.1	101.6	5.20	20.5	0.000	132.1	0.000	0.000	1.00

Table 28: Linear use checklist for connecting a MAS design to DESBench.

Step	What DESBench fixes	What the user provides
Choose instance	Reported instance, topology, route grammar, arrivals, disturbances, budgets, and metrics.	Select A3C9-1, A5C12-1, A5C12-2, or A5C12-3.
Connect runner	Active-agent payloads, local observations, protocol context, and legal-action masks.	Return one legal action for each active agent through <code>select_actions</code> or the <code>run_mas.py</code> JSON boundary.
Declare paradigm	Built-in protocol semantics for <code>centralized</code> , <code>hierarchical</code> , <code>heterarchical_cnp</code> , <code>holonic_hybrid</code> , or newly proposed coordination paradigms.	Reuse a built-in paradigm or register a new <code>mode_id</code> for the proposed coordination semantics.
Run comparison	Public runners, locked test seeds, output schema, and trace format.	Use the same instance, seed split, runner boundary, model setting, and metrics for all compared methods.
Audit result	Metric summaries, runtime traces, and decision-audit fields.	Check coverage, fallback behavior, invalid actions, no-action epochs, and decision failures.

Table 29: Expected DESBench output artifacts for user-side evaluation and diagnosis.

Artifact type	Representative files or fields	How users should use them
Metrics summary	<code>suite_summary.csv</code> and per-case <code>summary.csv</code>	Start here for the primary 15 metrics and the broader Appendix C surface. Use these files to compare methods on the same reported instance.
Artifact manifest	Per-case <code>artifacts.csv</code>	Use it to locate the exact trace files and verify that the reported instance, runner, backend, and budget metadata match the intended experiment.
World and activation traces	<code>*_world_event_trace.json</code> and <code>*_activation_trace.json</code>	Diagnose when queueing, routing, transport, breakdowns, or activation patterns cause a coordination method to slow down or deadlock.
Settlement and protocol traces	<code>*_settlement_trace.json</code> and <code>*_protocol_trace.json</code>	Diagnose contract settlement, reroute, rejection, escalation, and message-level coordination behavior induced by Γ .
Decision audit fields	<code>framework_llm_decision_count</code> , <code>framework_fallback_contract_decision_count</code> , <code>framework_no_legal_action_count</code> , and related traces	Distinguish model-backed decisions from fallback and no-legal-action cases before concluding that one method is genuinely stronger.
Failure and fallback logs	<code>progress_log.jsonl</code> , <code>progress_snapshot.json</code> , and failure traces	Check exhausted retries, invalid responses, interrupted cases, and other execution failures that can confound comparison if left unexplained.