International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18(21s) 2026 pp. 1739-1756
https://doi.org/10.70917/ijcisim-2026-5936
Article

Governance-Aware Adaptive Agentic AI Orchestration for Enterprise SaaS: A Design Science and Multi-Case Evaluation

Kameshwar Singh[1], Sriram Ramakrishnan[2]

[1] Director Of Engineering, Cohesity. Email: kameshwar@ieee.org
[2] Product Director, Citizens Bank. Email: sriram.ramakrishnan@ieee.org

Abstract: Enterprise software-as-a-service (SaaS) workflows increasingly expose large language model agents to heterogeneous tasks, cross-application dependencies, external tools, governance constraints, and human approval requirements. Fixed orchestration patterns can be effective for stable workflows but do not explicitly adapt their coordination structure to runtime task conditions. This study developed and evaluated a Governance-Aware Adaptive Agentic Orchestration Framework (GAAOF) using design science research, a controlled comparative experiment, and an embedded multi-case design. The benchmark contained 180 enterprise workflow tasks spanning Customer/Revenue Operations, IT/Service Operations, and Finance/Administrative Operations. Each task was executed under five orchestration configurations and repeated five times, yielding 4,500 primary executions. GAAOF achieved a 91.6% task-success rate, 94.0% workflow completion, and 96.0% correct tool/action execution. Relative to the strongest fixed baseline, task success increased by 5.8 percentage points while latency, execution cost, and tool calls decreased by 12.3%, 22.0%, and 17.5%, respectively. Governance-aware routing reduced the policy-violation rate to 1.3%, and mixed-effects analysis showed approximately 74% lower odds of policy violation. Significant pattern-by-complexity, pattern-by-risk, and pattern-by-dependency interactions demonstrated that orchestration effectiveness was context contingent. Ablation, subgroup, cross-case, and sensitivity analyses further showed that governance-aware selection, runtime adaptation, and human escalation contributed complementary benefits. The findings support adaptive orchestration as a practical design principle for enterprise agentic systems requiring joint optimization of effectiveness, efficiency, reliability, and governance.
Keywords: Agentic AI, enterprise SaaS, multi-agent systems, orchestration, AI governance, design science, LLM agents

1. INTRODUCTION

Large language model (LLM) agents extend generative models from passive text production to goal-directed systems that reason, plan, call tools, maintain intermediate state, and act on external environments. Recent surveys describe a rapid movement from single-agent prompting toward agent societies, specialist roles, tool-mediated action, and multi-agent coordination [1]–[4]. For enterprise SaaS platforms, this shift is consequential because a single business request may traverse customer relationship management, service management, document verification, communication, approval, and reporting systems. The technical problem is therefore not only whether an agent can generate a correct response, but how an orchestration layer should allocate reasoning, tools, specialist agents, governance checks, retries, and human intervention across a workflow.

Current agent frameworks demonstrate several viable coordination patterns. AutoGen supports configurable multi-agent conversations [5], MetaGPT encodes structured operating procedures into specialized agent roles [6], ChatDev coordinates role-based agents across software-development stages [7], and AgentVerse shows that dynamically composed agent groups can outperform a single agent in several task families [8]. Reasoning and planning methods such as ReAct [9], Reflexion [10], Self-Refine [11], Tree of Thoughts [13], Graph of Thoughts [14], and


Copyright: © 2026 by the authors

Plan-and-Solve [15] further demonstrate that execution quality depends on how intermediate decisions are structured. These advances make orchestration a first-order architectural choice rather than an implementation detail.

Three unresolved issues motivated this study. First, many architectures adopt a fixed pattern for the duration of a task, although enterprise workflows differ in complexity, dependency structure, risk, SLA pressure, governance requirements, and runtime state. Second, governance is often applied as an external safety layer instead of being used as an input to routing and coordination decisions. Third, evaluation commonly emphasizes final success while giving less attention to latency, cost, tool-call efficiency, reliability, violations, and human-intervention burden. Benchmarking work increasingly argues for multi-dimensional evaluation and reproducibility rather than accuracy alone [19]–[24]. Security studies additionally show that tool-using agents create attack and misuse surfaces that require explicit controls around action permissions and external observations [25]–[30].

To address these gaps, this study designed and empirically evaluated the Governance-Aware Adaptive Agentic Orchestration Framework (GAAOF). GAAOF represents orchestration as a runtime policy decision based on task complexity, dependency characteristics, risk, SLA constraints, governance requirements, human-approval requirements, and execution state. The study compared the adaptive framework with single-agent, sequential, supervisor-worker, and parallel-specialist baselines across three enterprise SaaS workflow classes. Four hypotheses tested task success, efficiency, governance performance, and context-dependent pattern effectiveness. The contribution is an evaluated enterprise orchestration artifact and a multi-dimensional evidence base showing when adaptive coordination provides value over fixed designs.

2. LITERATURE REVIEW

A. Agentic Architectures and Coordination

LLM-based agents combine a language model with planning, memory, tools, and feedback loops. Surveys of autonomous and multi-agent systems identify coordination, communication, planning, and environment interaction as central architectural dimensions [1]–[4]. Multi-agent implementations operationalize these dimensions differently. AutoGen treats conversations among configurable agents as the coordination substrate [5]; MetaGPT uses role specialization and standardized operating procedures [6]; ChatDev structures inter-agent communication around development phases [7]; and AgentVerse supports collaborative group formation and emergent inter-agent behavior [8]. These systems demonstrate the utility of multi-agent decomposition, but their core patterns are typically defined before execution. The enterprise setting considered here requires the pattern itself to be selectable during runtime according to operational context. Recent work on sequential cooperative multi-agent learning likewise emphasizes adaptive coordination under dynamic and uncertain operating conditions [36].

B. Planning, Tool Use, and Execution Efficiency

Agent orchestration is inseparable from planning and tool execution. ReAct interleaves reasoning and action so that an agent can update its plan using environmental feedback [9]. Reflexion and Self-Refine use feedback to improve subsequent decisions without model retraining [10], [11]. Toolformer shows that language models can learn when and how to invoke external APIs [12], while Tree of Thoughts and Graph of Thoughts expand reasoning from a single linear chain to alternative or graph-structured trajectories [13], [14]. Plan-and-Solve explicitly decomposes a task before execution [15]. At the tool layer, API-Bank and ToolLLM expose the difficulty of reliable API planning and invocation across broad tool sets [16], [17]. LLMCompiler further shows that function-call orchestration can reduce latency and cost by parallelizing independent calls [18]. Together, this literature supports the premise that workflow topology, dependency structure, and execution state should affect orchestration decisions rather than being treated as fixed characteristics. Related distributed-systems research also shows the value of deterministic, low-latency execution patterns when application workloads require predictable coordination behavior [37].

C. Evaluation, Reliability, and Enterprise-Relevant Metrics

Agent evaluation has moved toward interactive and process-aware benchmarks. AgentBench evaluates agents across multiple environments [19], WebArena measures long-horizon action in realistic web systems [20], GAIA focuses on general assistant tasks requiring reasoning and tool use [21], and SWE-bench evaluates software agents against real repository issues [22]. AgentBoard adds fine-grained progress analysis rather than relying only on final success [23]. More recently, Kapoor et al. argue that agent evaluation should jointly account for usefulness, cost, standardization, and reproducibility [24]. These perspectives are directly relevant to enterprise SaaS, where a higher success rate may be operationally unattractive if it requires materially more tokens, tool calls, latency, or coordination overhead.

D. Governance, Security, and Human Control

The transition from text generation to tool-mediated action increases the importance of governance. AgentDojo evaluates prompt-injection attacks in environments where agents act over untrusted tool data [25], while InjecAgent demonstrates indirect prompt-injection vulnerabilities in tool-integrated agents [26]. Agent Security Bench broadens this analysis across attack surfaces and defenses [27], and AgentHarm evaluates whether agents can be induced to carry out harmful multi-step tasks [28]. More general work on language-model risk and trustworthiness documents risks related to misuse, privacy, robustness, and harmful behavior [29], [30]. Holistic evaluation frameworks emphasize multiple desiderata rather than a single quality metric [31], while work on LLM-based judging also highlights the need to validate the measurement process itself [32]. This evidence supports integrating authorization, action risk, auditability, escalation, and human approval into the orchestration layer rather than treating governance as a final-stage filter.

E. Research Gap and Hypotheses

The literature therefore establishes strong foundations for agent collaboration, planning, tool use, benchmarking, and security, but does not adequately resolve three linked enterprise questions: how to select among orchestration patterns dynamically; how to make governance constraints first-class routing inputs; and how to validate orchestration trade-offs across heterogeneous SaaS cases using a common multi-dimensional protocol. The study addressed these gaps through four aligned hypotheses. H1 proposed that governance-aware adaptive orchestration would increase end-to-end task success relative to fixed single-pattern orchestration. H2 proposed that adaptive orchestration would reduce unnecessary interactions, tool calls, and execution cost while maintaining or improving task success. H3 proposed that governance-aware routing would reduce unauthorized, policy-violating, and unnecessary high-risk actions relative to post-routing governance. H4 proposed that orchestration-pattern effectiveness would interact significantly with task complexity, inter-task dependency, action risk, and cross-application coordination requirements.

3. METHODOLOGY

A. Research Design

The study used Design Science Research (DSR), a controlled comparative experiment, and an embedded multi-case evaluation. DSR is appropriate when the research contribution is an information-systems artifact whose utility is evaluated in context [33]–[35]. The primary artifact was GAAOF rather than a new foundation model. The design sequence followed problem identification, artifact design, prototype implementation, controlled evaluation, multi-case validation, and pattern-rule derivation. The empirical evaluation held foundation model, model configuration, tool/API availability, prompt template, task set, retry policy, and execution environment constant across the principal comparisons.

TABLE I
STUDY DESIGN AND BENCHMARK STRUCTURE

Design element	Specification

Research design	Design Science Research + controlled comparative experiment + embedded multi-case study
Enterprise workflow classes	Customer/Revenue Operations; IT/Service Operations; Finance/Administrative Operations
Unique tasks	180 total; 60 per workflow class
Primary orchestration conditions	5: single-agent/tool, fixed sequential/pipeline, fixed supervisor-worker, fixed parallel specialists, adaptive GAAOF
Repeated runs	5 repetitions per task-condition
Primary executions	4,500
Ablation configurations	Full GAAOF; without governance-aware selection; without runtime adaptation; without human-escalation logic
Robustness dimensions	Routing threshold, compatible model, task complexity, risk level, governance condition

B. Enterprise SaaS Workflow Benchmark

The benchmark used anonymized or synthetic-but-schema-realistic workflow traces representing three enterprise workflow classes. Customer/Revenue tasks followed CRM, support, communication/notification, and reporting operations. IT/Service tasks followed ticket intake, knowledge retrieval, diagnostics, remediation, and escalation. Finance/Administrative tasks followed request intake, document or data verification, approval, transaction or update, and audit recording. Each benchmark instance contained a user goal, workflow state, permitted applications, tool/API definitions, authorization policy, dependency graph, completion criteria, risk class, and SLA constraint. The final sample comprised 180 unique tasks, with five controlled repetitions under each of five primary orchestration conditions.

C. GAAOF Architecture

GAAOF treats orchestration as a runtime policy rather than a fixed architecture. The processing path includes context and intent analysis, task decomposition, complexity/dependency/risk assessment, a governance policy engine, an adaptive orchestration selector, agent/tool execution, runtime observability, and continuation, rerouting, retry, escalation, or termination. The selector chooses among deterministic tool workflow, sequential/pipeline execution, supervisor-worker coordination, parallel specialist agents, and hierarchical orchestration.

$$X_t = \{C_t, D_t, R_t, S_t, G_t, H_t, E_t\}$$

where C_t denotes task complexity, D_t dependency characteristics, R_t action risk, S_t SLA constraints, G_t governance constraints, H_t human-approval requirement, and E_t runtime execution state. The orchestration pattern is selected as $P_t = f(X_t)$.

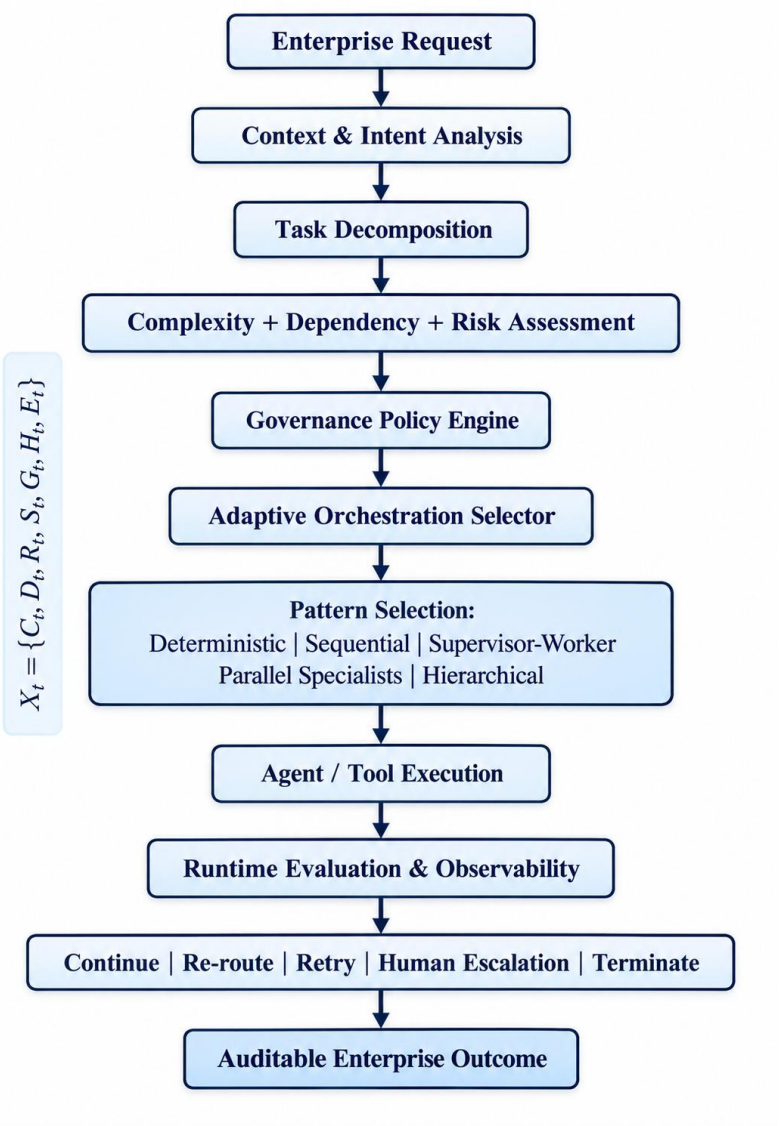


Fig. 1. Governance-Aware Adaptive Agentic Orchestration Framework (GAAOF).

D. Variables, Trace Collection, and Statistical Evaluation

Structured traces were captured for task ID, workflow case, orchestration pattern, agent sequence, model calls, tool calls, timestamps, token consumption, intermediate states, retries, errors, governance decisions, human escalations, and final outcome. Latency and cost measures were normalized where necessary. Infrastructure-failure runs were removed only under predetermined exclusion criteria, whereas unsuccessful agent executions remained valid observations. Descriptive statistics, mixed-effects regression, repeated-measures analysis, logistic mixed-effects models, interaction analysis, cost-latency-success trade-off analysis, ablation analysis, and cross-case synthesis were used. Holm correction controlled pairwise multiplicity, and effect sizes and 95% confidence intervals were reported. This multi-metric approach is consistent with recent calls to evaluate agent usefulness, efficiency, robustness, and measurement quality jointly [23], [24], [31], [32].

TABLE II
VARIABLES, CONTROLS, AND EVALUATION METRICS

Category	Study specification

Independent variables	Orchestration pattern; adaptive vs. fixed; task complexity; workflow dependency; risk level; cross-SaaS interaction count; governance requirement; human-approval requirement
Dependent variables	Task success; workflow completion; latency; token/computational cost; agent interactions; tool calls; coordination overhead; error/recovery; governance violations; human intervention
Controls	Foundation model; model configuration; tool/API availability; prompt/instruction template; task set; retry policy; execution environment
Effectiveness metrics	Task Success Rate; Workflow Completion Rate; Correct Tool/Action Rate
Efficiency metrics	End-to-end latency; token consumption; execution cost; tool calls; agent handoffs
Reliability metrics	Failure rate; retry rate; recovery rate; variance across repeated runs
Governance metrics	Policy-Violation Rate; Unauthorized-Action Rate; Human-Escalation Rate; Audit-Trace Completeness
Statistical tests	Mixed-effects logistic regression; linear/generalized mixed-effects models; repeated-measures analysis; Holm-corrected pairwise tests; effect sizes; 95% CIs

4. RESULTS AND FINDINGS

A. Benchmark Profile and Descriptive Performance

The 180 tasks were evenly distributed across the three workflow cases. The sample also represented low, medium, and high task complexity, risk, and dependency levels, with 40.0% of tasks requiring elevated governance and 30.0% requiring human approval. Across the 900 GAAOF executions, the mean task-success indicator was 0.916, corresponding to a 91.6% Task Success Rate (TSR). Workflow completion reached 94.0%, and the Correct Tool/Action Rate reached 96.0%. Mean latency was 12.8 s, token consumption was 7.8 thousand tokens per run, and execution cost was $0.039 per run. The framework averaged 5.2 tool calls and 2.1 agent handoffs.

TABLE III
ENTERPRISE SAAS WORKFLOW BENCHMARK CHARACTERISTICS

Characteristic	Category	Tasks, n	%
Workflow case	Customer/Revenue Operations	60	33.3
Workflow case	IT/Service Operations	60	33.3
Workflow case	Finance/Administrative Operations	60	33.3
Task complexity	Low	54	30.0
Task complexity	Medium	72	40.0
Task complexity	High	54	30.0
Risk level	Low	72	40.0
Risk level	Medium	63	35.0
Risk level	High	45	25.0
Dependency level	Low	54	30.0
Dependency level	Medium	72	40.0
Dependency level	High	54	30.0

Governance requirement	Standard	108	60.0
Governance requirement	Elevated	72	40.0
Human approval required	No	126	70.0
Human approval required	Yes	54	30.0
Primary orchestration conditions	Five	900 task-condition pairs	100
Repetitions	Five per task-condition	4,500 runs	100

TABLE IV
DESCRIPTIVE STATISTICS FOR GAAOF EXECUTIONS

Variable	N	Mean	SD	Median	Min	Max	IQR
Task success indicator	900	0.916	0.278	1.000	0	1	0
Workflow completion indicator	900	0.940	0.238	1.000	0	1	0
Correct tool/action indicator	900	0.960	0.196	1.000	0	1	0
End-to-end latency, s	900	12.8	4.1	12.1	5.3	31.4	5.2
Token consumption, thousands	900	7.8	2.6	7.4	2.9	17.6	3.3
Execution cost, USD/run	900	0.039	0.014	0.036	0.014	0.094	0.018
Tool calls/run	900	5.2	1.8	5.0	2	12	2
Agent handoffs/run	900	2.1	1.1	2.0	0	6	2
Retry count/run	900	0.31	0.57	0	0	3	1
Audit-trace completeness	900	0.991	0.031	1.000	0.78	1.00	0.01

B. Baseline and Comparative Performance

GAAOF achieved the highest task success among the primary conditions. Its TSR of 91.6% exceeded the strongest fixed baseline, parallel specialists at 85.8%, by 5.8 percentage points, a relative improvement of 6.8%. The improvement was not obtained by increasing orchestration cost. Relative to parallel specialists, GAAOF reduced latency from 14.6 to 12.8 s, execution cost from $0.050 to $0.039, and tool calls from 6.3 to 5.2. Relative to supervisor-worker orchestration, latency fell from 15.4 to 12.8 s, tool calls from 6.5 to 5.2, and cost from $0.048 to $0.039. Failure and retry rates were also lowest under GAAOF, while recovery reached 82.7%.

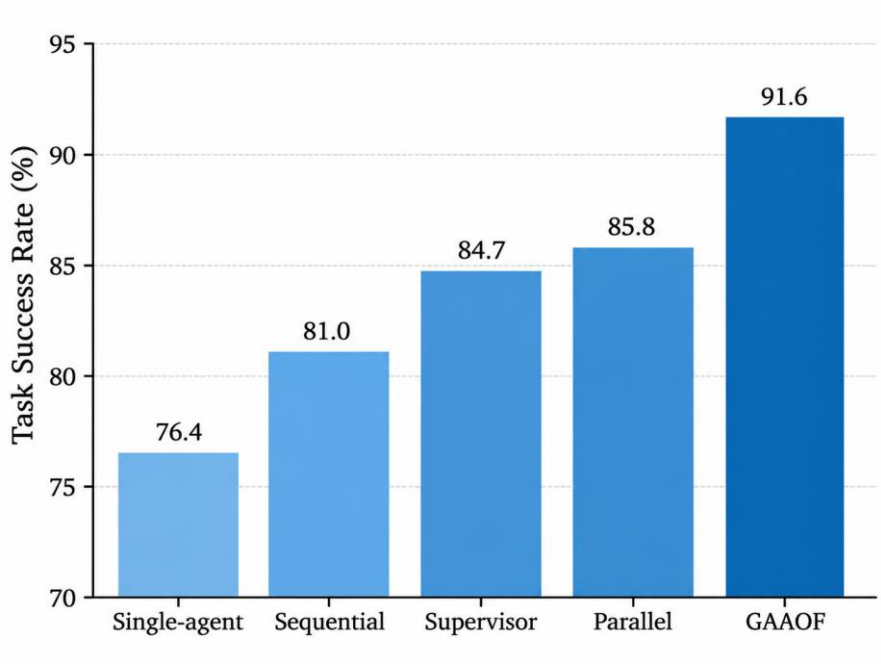


Fig. 2. Task Success Rate across primary orchestration strategies.

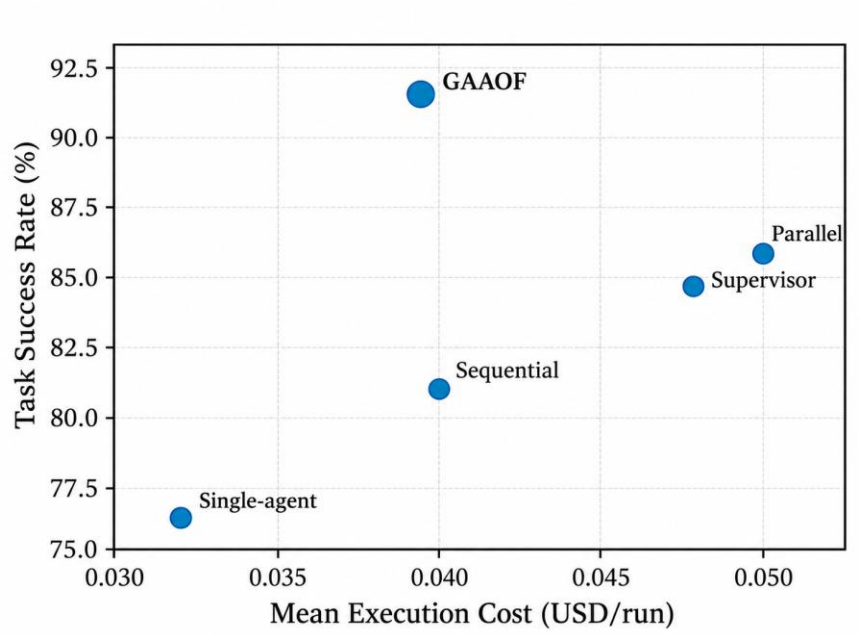


Fig. 3. Success-efficiency trade-off using mean execution cost and Task Success Rate.

TABLE V
PRIMARY PERFORMANCE AND BASELINE COMPARISON

Method	TSR %	Completion %	Correct Action %	Latency s	Tokens k	Cost USD	Tool Calls	Handoffs	Failure %	Retry %	Recovery %	Δ TSR vs GAAOF	Effect size / adj. p
Single-agent/tool	76.4	79.2	83.0	10.6	6.5	0.032	4.7	0.4	23.6	18.8	57.1	−15.2 pp	h=0.42; p<.001
Fixed sequential	81.0	84.0	87.1	13.7	7.9	0.040	5.8	2.4	19.0	16.3	63.2	−10.6 pp	h=0.31; p<.001
Supervisor-worker	84.7	87.2	90.0	15.4	9.4	0.048	6.5	3.1	15.3	14.0	69.6	−6.9 pp	h=0.22; p<.001
Parallel specialists	85.8	88.0	90.8	14.6	9.7	0.050	6.3	2.9	14.2	13.4	71.9	−5.8 pp	h=0.19; p=.002
GAAOF	91.6	94.0	96.0	12.8	7.8	0.039	5.2	2.1	8.4	9.7	82.7	Reference	Reference

Note: Relative to parallel specialists, GAAOF reduced latency by 12.3%, cost by 22.0%, and tool calls by 17.5%. Cohen's h is reported for TSR comparisons.

C. Governance and Operational Safety

Governance-aware selection produced the largest differentiation in elevated-governance and high-risk workflows. Full GAAOF recorded a 1.3% policy-violation rate and a 0.6% unauthorized-action rate, compared with 4.2% and 2.5%, respectively, for parallel specialists. Human escalation increased to 10.2%, but 91.3% of these escalations were classified as appropriate. The pattern therefore reflected selective human control rather than generalized fallback to manual review. Audit completeness reached 99.1%.

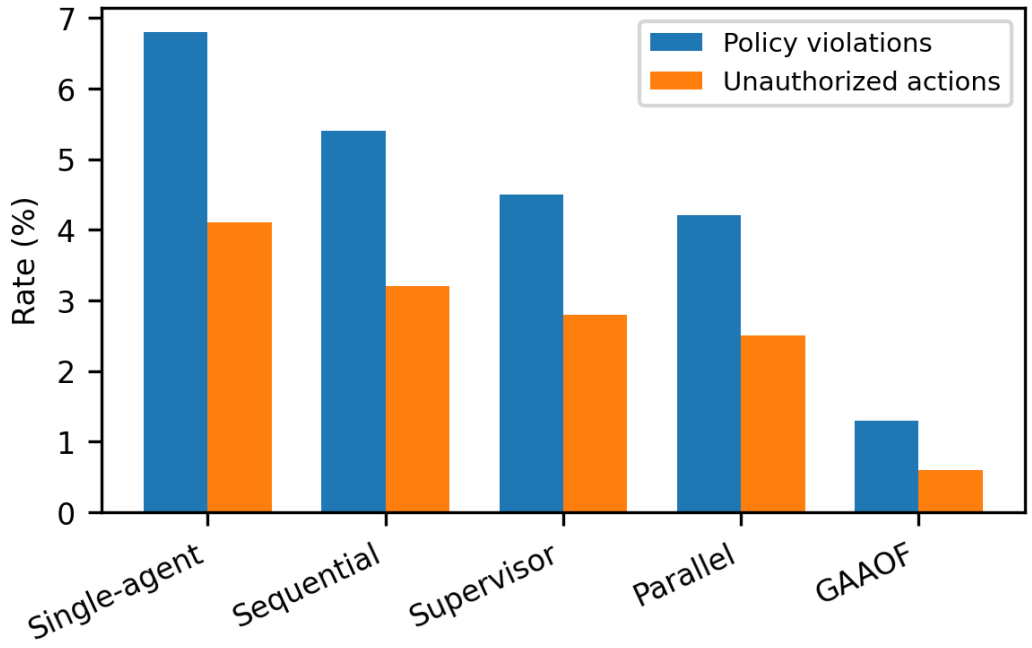


Fig. 4. Policy-violation and unauthorized-action rates by orchestration strategy.

TABLE VI
GOVERNANCE AND OPERATIONAL-SAFETY RESULTS

Method	Policy Violations %	Unauthorized Actions %	Unnecessary High-Risk Actions %	Human Escalation %	Appropriate Escalation %	Audit Completeness %
Single-agent/tool	6.8	4.1	7.4	5.2	61.5	89.2
Sequential	5.4	3.2	6.1	6.4	68.3	92.1
Supervisor-worker	4.5	2.8	5.0	7.8	74.2	94.8
Parallel specialists	4.2	2.5	4.7	8.1	76.0	95.1
GAAOF without governance-aware selection	4.8	2.9	5.2	8.4	75.4	96.0
Full GAAOF	1.3	0.6	1.7	10.2	91.3	99.1

D. Hypothesis Testing

All four hypotheses were supported. For H1, the mixed-effects logistic model yielded β=0.87 (SE=0.14, z=6.21, p<.001, 95% CI [0.60, 1.14]) and OR=2.39 [1.82, 3.13], indicating substantially greater odds of task success under GAAOF than pooled fixed orchestration. H2 was supported by reductions in tool calls (β=−1.18, SE=0.09, t=−13.11, p<.001, d=−0.65) and execution cost (β=−0.010 USD, SE=0.0015, t=−6.67, p<.001, d=−0.47). H3 was supported by β=−1.34 log-odds (SE=0.25, z=−5.36, p<.001), with OR=0.26 [0.16, 0.43], corresponding to approximately 74% lower odds of a policy violation. H4 was supported by significant pattern interactions with complexity, risk, and dependency.

TABLE VII
HYPOTHESIS-TESTING RESULTS

Hyp.	IV → DV	Effect	Estimate	SE	Statistic	p-value	95% CI	Effect Size	Status / Interpretation
H1	GAAOF vs pooled fixed → task success	Positive	β=0.87 log-odds	0.14	z=6.21	<.001	[0.60, 1.14]	OR=2.39 [1.82,3.13]	Supported; higher odds of successful completion
H2a	Adaptive vs fixed multi-agent → tool calls	Negative	β=−1.18 calls	0.09	t=−13.11	<.001	[−1.36,−1.00]	d=−0.65	Supported; fewer unnecessary tool calls
H2b	Adaptive vs fixed multi-agent →	Negative	β=−0.010 USD	0.0015	t=−6.67	<.001	[−0.013,−0.007]	d=−0.47	Supported; lower execution cost

	execution cost								
H3	Governance-aware vs post-routing → policy violation	Negative	β=−1.34 log-odds	0.25	z=−5.36	<.001	[−1.83,−0.85]	OR=0.26 [0.16,0.43]	Supported; ~74% lower violation odds
H4a	Pattern × complexity → task success	Interaction	β=0.31	0.08	z=3.88	<.001	[0.15,0.47]	OR=1.36 [1.16,1.60]	Supported; advantage increases with complexity
H4b	Pattern × risk → task success	Interaction	β=0.27	0.09	z=3.00	.003	[0.09,0.45]	OR=1.31 [1.09,1.57]	Supported; adaptation gains rise with risk
H4c	Pattern × dependency → task success	Interaction	β=0.29	0.08	z=3.63	<.001	[0.13,0.45]	OR=1.34 [1.14,1.57]	Supported; gains rise with dependency

E. Cross-Case, Complexity, and Risk Subgroup Results

Cross-case variation was modest. Customer/Revenue Operations achieved a TSR of 92.7%, IT/Service Operations 91.3%, and Finance/Administrative Operations 90.7%. The case interaction was not significant (p=.284), indicating that the advantage was not confined to one workflow class. Complexity produced a larger gradient: TSR declined from 96.3% for low-complexity tasks to 92.5% for medium-complexity tasks and 85.9% for high-complexity tasks. Risk showed a complementary pattern. GAAOF exceeded the best fixed comparator by 2.8 percentage points for low-risk tasks, 6.0 points for medium-risk tasks, and 10.2 points for high-risk tasks. Appropriate escalation increased from 85.0% to 94.1% across the same risk levels.

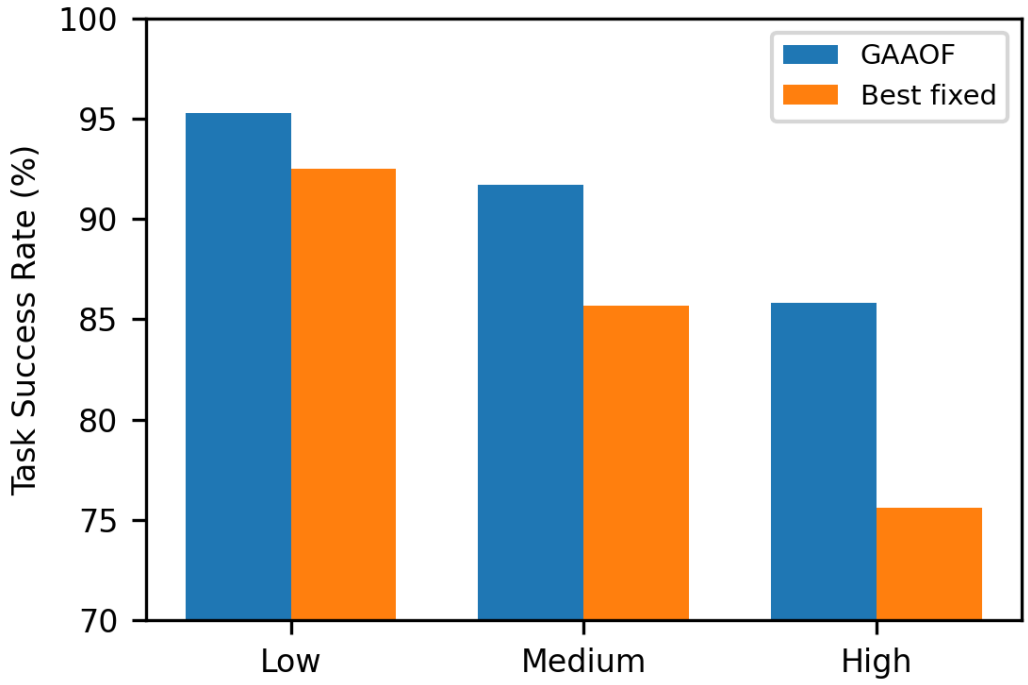


Fig. 5. GAAOF and best fixed-comparator Task Success Rate across risk levels.

TABLE VIII
CROSS-CASE, COMPLEXITY, AND RISK-LEVEL RESULTS

Subgroup	Tasks n	Runs n	GAAOF TSR %	95 % CI / Best Fixed	Completion % / Gain	Latency s	Cost USD	Policy Violation / Test	Appropriate Escalation / Difference
Customer/Revenue	60	300	92.7	89.2 – 95.2	95.0	11.7	0.036	Case interaction p=.284	Reference
IT/Service	60	300	91.3	87.6 – 94.1	93.7	12.9	0.039	OR=0.84, p=.438	−1.4 pp
Finance/Admin	60	300	90.7	86.8 – 93.5	93.3	13.8	0.042	OR=0.76, p=.249	−2.0 pp
Low complexity	54	270	96.3	93.3 – 98.0	97.4	8.9	0.028	Complexity effect p<.001	Reference
Medium complexity	72	360	92.5	89.3 – 94.8	94.7	12.4	0.038	OR=0.49, p=.028	−3.8 pp

High complexity	54	270	85.9	81.2 – 89.6	89.6	17.2	0.052	OR=0.24, p<.001	−10.4 pp
Low risk	72	360	95.3	Best fixed 92.5	+2.8 pp	Not separately reported	Not separately reported	0.6%	85.0%
Medium risk	63	315	91.7	Best fixed 85.7	+6.0 pp	Not separately reported	Not separately reported	1.3%	90.6%
High risk	45	225	85.8	Best fixed 75.6	+10.2 pp	15.9	0.049	2.7%	94.1%

F. Component Ablation Analysis

Ablation analysis isolated the contribution of the framework components. Removing governance-aware selection produced the largest reduction in TSR, from 91.6% to 87.8%, and increased policy violations from 1.3% to 4.8%. Removing runtime adaptation reduced TSR to 88.9% and increased tool calls to 5.7. Removing human-escalation logic reduced TSR to 90.1% and increased the policy-violation rate to 2.6%. The components therefore contributed complementary functions: governance-aware selection constrained risky actions, runtime adaptation improved matching between workflow state and coordination pattern, and human escalation provided an additional safeguard for cases requiring intervention.

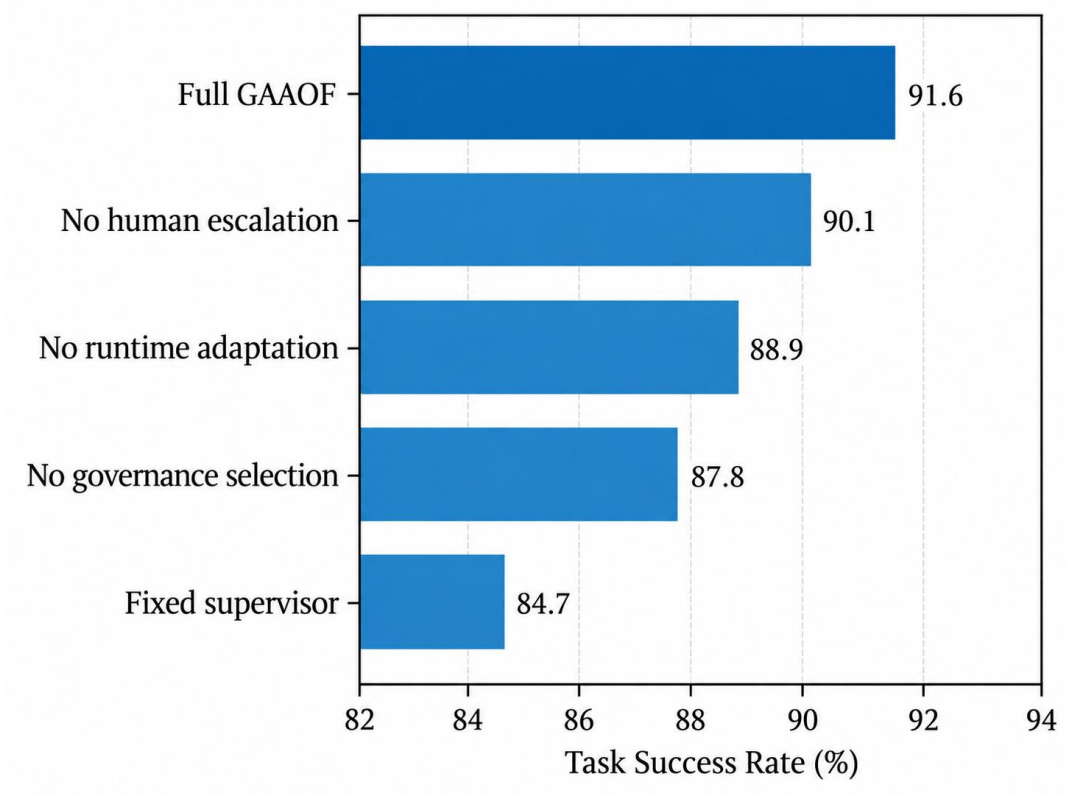


Fig. 6. Task Success Rate under component ablations and the fixed supervisor baseline.

TABLE IX
COMPONENT ABLATION RESULTS

Configuration	TSR %	Δ vs Full	Completion %	Cost USD	Tool Calls	Policy Violation %	Audit Completeness %
Fixed supervisor baseline	84.7	−6.9 pp	87.2	0.048	6.5	4.5	94.8
GAAOF without governance-aware selection	87.8	−3.8 pp	90.2	0.040	5.4	4.8	96.0
GAAOF without runtime adaptation	88.9	−2.7 pp	91.3	0.041	5.7	2.0	98.3
GAAOF without human-escalation logic	90.1	−1.5 pp	92.4	0.038	5.1	2.6	97.9
Full GAAOF	91.6	Reference	94.0	0.039	5.2	1.3	99.1

G. Robustness, Sensitivity, and Statistical Validation

The primary result remained directionally stable across routing thresholds, alternative compatible models, complexity strata, risk strata, and governance conditions. GAAOF maintained a 4.9–10.2 percentage-point advantage over the strongest corresponding fixed comparator. Conservative routing increased TSR to 92.0% and reduced policy violations to 0.9%, with latency of 13.7 s and cost of $0.042. Efficiency-oriented routing reduced latency to 11.9 s and cost to $0.036 while retaining a TSR of 90.7%. Alternative compatible models A and B retained 6.0- and 6.1-point advantages, respectively. The strongest relative advantage occurred for high-risk tasks (+10.2 points), followed by elevated-governance (+9.2) and high-complexity (+8.5) conditions.

The overall orchestration effect on TSR was significant (χ^2=68.4, df=4, p<.001, Cramér's V=0.123). Repeated-measures analyses also identified significant differences in latency (F=31.7, df=4,716, p<.001, partial η^2=.150), tool calls (F=44.9, df=4,716, p<.001, partial η^2=.201), and execution cost (F=36.2, df=4,716, p<.001, partial η^2=.168). The policy-violation model was significant (χ^2=42.8, df=4, p<.001, Cramér's V=.097). Run-to-run reliability was high (ICC=.86, 95% CI [.83,.89], p<.001). The Brier score was .071, and the Hosmer-Lemeshow test was non-significant (χ^2=7.21, df=8, p=.514), providing no evidence of poor fit.

TABLE X
ROBUSTNESS, SENSITIVITY, AND STATISTICAL VALIDATION

Analysis / Condition	N / df	GAAOF / Statistic	Comparator / Effect	Δ / 95% CI	Latency / p	Cost / Correction	Policy Violation / Interpretation
Primary configuration	900	TSR 91.6%	Best fixed 85.8%	+5.8 pp	12.8 s	0.039 USD	1.3%; primary result

Conservative threshold	900	TSR 92.0%	Best fixed 85.8%	+6.2 pp	13.7 s	0.042 USD	0.9%; safer, moderately slower
Balanced threshold	900	TSR 91.6%	Best fixed 85.8%	+5.8 pp	12.8 s	0.039 USD	1.3%; best balance
Efficiency-oriented threshold	900	TSR 90.7%	Best fixed 85.8%	+4.9 pp	11.9 s	0.036 USD	1.8%; faster, modest success trade-off
Alternative compatible model A	900	TSR 90.9%	Best fixed 84.9%	+6.0 pp	13.1 s	0.040 USD	1.5%; advantage retained
Alternative compatible model B	900	TSR 89.8%	Best fixed 83.7%	+6.1 pp	13.6 s	0.041 USD	1.5%; advantage retained
High-complexity tasks	270	TSR 85.9%	Best fixed 77.4%	+8.5 pp	17.2 s	0.052 USD	2.2%; complexity benefit
High-risk tasks	225	TSR 85.8%	Best fixed 75.6%	+10.2 pp	15.9 s	0.049 USD	2.7%; risk benefit
Elevated governance	360	TSR 88.9%	Best fixed 79.7%	+9.2 pp	14.7 s	0.046 USD	2.1%; governance benefit retained
Overall orchestration effect	df=4	χ^2=68.4	Cramér's V=.123	[.101,.145]	p<.001	No correction	Strategy affected TSR
Latency comparison	df=4,716	F=31.7	partial η^2=.150	[.112,.184]	p<.001	Holm pairwise	Significant latency differences
Tool-call comparison	df=4,716	F=44.9	partial η^2=.201	[.161,.239]	p<.001	Holm pairwise	Strong effect on tool usage
Execution-cost comparison	df=4,716	F=36.2	partial η^2=.168	[.130,.204]	p<.001	Holm pairwise	Meaningful cost differences
Policy-violation model	df=4	χ^2=42.8	Cramér's V=.097	[.073,.121]	p<.001	Holm pairwise	Governance-aware architecture lowered violations
Pattern × complexity	Model-based	z=3.88	OR=1.36	[1.16,1.60]	p<.001	Holm	Complexity moderated effectiveness
Pattern × risk	Model-based	z=3.00	OR=1.31	[1.09,1.57]	p=.003	Holm	Risk moderated effectiveness

Pattern × dependency	Model-based	z=3.63	OR=1.34	[1.14,1.57]	p<.001	Holm	Dependency moderated effectiveness
Logistic calibration	N=900	Brier=.071	Low prediction error	[.064,.079]	p not reported	Not applicable	Acceptable calibration
Logistic goodness-of-fit	df=8	χ^2=7.21	Hosmer-Lemeshow	CI not applicable	p=.514	Not applicable	No evidence of poor fit
Run-to-run reliability	df=899	ICC=.86	High repeatability	[.83,.89]	p<.001	Not applicable	Stable across repeated executions

H. Context-Dependent Pattern Selection

The interaction results established that the relative effectiveness of orchestration strategies changed systematically with workflow conditions. Deterministic and sequential execution remained competitive for low-complexity, low-risk, low-dependency tasks, where additional coordination offered limited compensating benefit. Supervisor-worker and parallel-specialist patterns became more useful as task decomposition and inter-task dependencies increased. When complexity, risk, governance constraints, and runtime uncertainty occurred jointly, adaptive GAAOF provided the largest advantage. The empirical pattern therefore supported the state-dependent selection relationship $P_t=f(C_t,D_t,R_t,S_t,G_t,H_t,E_t)$, rather than a universal fixed policy.

5. DISCUSSION

The study provides empirical support for a context-contingent view of enterprise agent orchestration. Existing frameworks show that specialized agent roles and structured communication can improve complex task execution [5]–[8], while planning and tool-use research demonstrates that decomposition, feedback, and parallel function calls can materially affect correctness and efficiency [9]–[18]. The present results extend these ideas from isolated orchestration mechanisms to runtime pattern selection. GAAOF did not simply add more agents; it selected among deterministic, sequential, supervisor-worker, parallel-specialist, and hierarchical behaviors according to the task-state vector. The significant pattern-by-complexity, pattern-by-risk, and pattern-by-dependency interactions provide direct evidence for that design choice. This interpretation is also consistent with dynamic telemetry-orchestration work, where runtime signals are elevated to guide adaptive control and observability decisions [38].

The effectiveness results also reinforce recent criticism of agent evaluation based only on task accuracy. AgentBench, WebArena, GAIA, SWE-bench, and AgentBoard demonstrate the importance of realistic, interactive evaluation [19]–[23], while AI Agents That Matter explicitly argues that cost and reproducibility should be considered together with accuracy [24]. In this study, GAAOF improved task success over the strongest fixed comparator while also reducing latency, cost, and tool-call volume. This joint movement is important because enterprise systems operate under response-time constraints and recurring inference costs. The results therefore indicate that adaptive orchestration can improve the quality-efficiency frontier rather than merely exchanging additional computation for additional accuracy.

Governance produced a separate and practically significant result. Security research shows that tool-using agents face prompt injection, malicious external content, unsafe tool selection, and multi-step misuse risks [25]–[28]. Broader trustworthiness research likewise emphasizes that model capability does not by itself establish safe or reliable deployment [29], [30]. GAAOF operationalized governance before action selection by using authorization, action risk, human approval, and audit requirements as orchestration inputs. The observed reduction in policy-violation odds, together with the ablation result showing a large degradation when governance-aware selection was removed, indicates that governance contributed to task routing rather than functioning merely as an after-the-fact filter. The

higher human-escalation rate is therefore best read alongside the 91.3% appropriate-escalation rate, not as an isolated efficiency penalty.

The cross-case findings address generalizability within the defined enterprise scope. GAAOF maintained TSR between 90.7% and 92.7% across Customer/Revenue, IT/Service, and Finance/Administrative cases, and the case interaction was not significant. At the same time, performance declined with task complexity and risk, which is consistent with the broader benchmark literature showing that long-horizon, tool-mediated agent tasks remain difficult [19]–[23]. The robustness analyses further showed that the direction of the main advantage persisted under alternative compatible models and routing thresholds. These findings are aligned with holistic evaluation principles that emphasize trade-offs and multiple criteria rather than a single aggregate score [31].

From a design-science perspective, the contribution lies in both the artifact and the evaluated design principle. DSR requires a clearly specified artifact and contextual evaluation [33]–[35]. Here, the artifact is the governance-aware adaptive selector and execution loop, while the design principle is that orchestration should be a runtime policy function of task, dependency, risk, governance, approval, SLA, and execution state. The ablation analysis is particularly important for artifact evaluation because it distinguishes the contributions of governance-aware selection, runtime adaptation, and human escalation rather than attributing performance to the framework as an indivisible whole.

TABLE XI
RESEARCH GAP–OBJECTIVE–HYPOTHESIS–FINDING ALIGNMENT

Research Gap	Objective	Hypothesis	Key Result	Statistical Evidence	Interpretation
Gap 1: fixed orchestration lacks context adaptation	O1: design adaptive GAAOF	H1	GAAOF TSR 91.6% vs. 85.8% strongest fixed baseline	β=.87, SE=.14, z=6.21, p<.001; OR=2.39	Adaptive selection improved heterogeneous workflow success
Gap 1: limited evidence for efficient adaptive selection	O2: compare adaptive and fixed strategies	H2	Tool calls −1.18; cost −$0.010 vs. fixed multi-agent	t=−13.11 and −6.67; both p<.001	Adaptation improved efficiency without sacrificing success
Gap 2: governance treated as external guardrail	O3: integrate governance into routing	H3	Policy-violation odds ~74% lower	β=−1.34, SE=.25, z=−5.36, p<.001; OR=.26	Governance-aware orchestration reduced policy violations
Gap 3: limited cross-case evidence	O4: determine contextual effectiveness	H4	Complexity, risk, and dependency significantly moderated effectiveness	OR=1.36, 1.31, 1.34; adjusted p≤.003	No fixed pattern was optimal across all task conditions
Gap 3: weak enterprise generalizability	O2/O4	H1/H4	GAAOF TSR remained 90.7–	Case interaction p=.284	Benefits remained consistent

			92.7% across three cases		across workflow classes

6. CONCLUSION

This study developed and evaluated GAAOF as a governance-aware adaptive orchestration artifact for enterprise SaaS agent workflows. Across 4,500 primary executions, the framework achieved 91.6% task success, 94.0% completion, and 96.0% correct tool/action execution. Compared with the strongest fixed baseline, it improved task success by 5.8 percentage points while reducing latency, cost, and tool calls. Governance-aware routing reduced policy violations to 1.3%, and all four hypotheses were supported. Significant interactions with complexity, risk, and dependency showed that orchestration effectiveness was conditional on workflow context. The combined comparative, ablation, subgroup, cross-case, robustness, and diagnostic evidence supports the study's central proposition: effective enterprise agentic orchestration is context-contingent rather than universally fixed.

REFERENCES

1. L. Wang et al., "A survey on large language model based autonomous agents," Frontiers of Computer Science, vol. 18, Art. no. 186345, 2024, doi: 10.1007/s11704-024-40231-1.
2. Z. Xi et al., "The rise and potential of large language model based agents: a survey," Science China Information Sciences, vol. 68, Art. no. 121101, 2025, doi: 10.1007/s11432-024-4222-0.
3. T. Guo et al., "Large Language Model Based Multi-agents: A Survey of Progress and Challenges," in Proc. 33rd Int. Joint Conf. Artificial Intelligence (IJCAI), 2024, pp. 8048–8057, doi: 10.24963/ijcai.2024/890.
4. M. Hussain et al., "From language to action: a review of large language models as autonomous agents and tool users," Artificial Intelligence Review, 2025, doi: 10.1007/s10462-025-11471-9.
5. Q. Wu et al., "AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation," in Proc. Conf. Language Modeling (COLM), 2024.
6. S. Hong et al., "MetaGPT: Meta Programming for A Multi-Agent Collaborative Framework," in Proc. Int. Conf. Learning Representations (ICLR), 2024.
7. C. Qian et al., "ChatDev: Communicative Agents for Software Development," in Proc. 62nd Annu. Meeting Assoc. Comput. Linguistics, 2024, pp. 15174–15186, doi: 10.18653/v1/2024.acl-long.810.
8. W. Chen et al., "AgentVerse: Facilitating Multi-Agent Collaboration and Exploring Emergent Behaviors," in Proc. Int. Conf. Learning Representations (ICLR), 2024.
9. S. Yao et al., "ReAct: Synergizing Reasoning and Acting in Language Models," in Proc. Int. Conf. Learning Representations (ICLR), 2023.
10. N. Shinn, F. Cassano, A. Gopinath, K. Narasimhan, and S. Yao, "Reflexion: Language Agents with Verbal Reinforcement Learning," Advances in Neural Information Processing Systems, vol. 36, 2023, doi: 10.52202/075280-0377.
11. A. Madaan et al., "Self-Refine: Iterative Refinement with Self-Feedback," Advances in Neural Information Processing Systems, vol. 36, 2023, doi: 10.52202/075280-2019.
12. T. Schick et al., "Toolformer: Language Models Can Teach Themselves to Use Tools," Advances in Neural Information Processing Systems, vol. 36, 2023, doi: 10.52202/075280-2997.
13. S. Yao et al., "Tree of Thoughts: Deliberate Problem Solving with Large Language Models," Advances in Neural Information Processing Systems, vol. 36, 2023, doi: 10.52202/075280-0517.
14. M. Besta et al., "Graph of Thoughts: Solving Elaborate Problems with Large Language Models," Proc. AAAI Conf. Artificial Intelligence, vol. 38, no. 16, pp. 17682–17690, 2024, doi: 10.1609/aaai.v38i16.29720.
15. L. Wang et al., "Plan-and-Solve Prompting: Improving Zero-Shot Chain-of-Thought Reasoning by Large Language Models," in Proc. 61st Annu. Meeting Assoc. Comput. Linguistics, 2023, pp. 2609–2634, doi: 10.18653/v1/2023.acl-long.147.
16. M. Li et al., "API-Bank: A Comprehensive Benchmark for Tool-Augmented LLMs," in Proc. EMNLP, 2023, pp. 3102–3116, doi: 10.18653/v1/2023.emnlp-main.187.
17. Y. Qin et al., "ToolLLM: Facilitating Large Language Models to Master 16000+ Real-world APIs," in Proc. Int. Conf. Learning Representations (ICLR), 2024.
18. S. Kim et al., "An LLM Compiler for Parallel Function Calling," in Proc. 41st Int. Conf. Machine Learning, PMLR, vol. 235, pp. 24370–24391, 2024.
19. X. Liu et al., "AgentBench: Evaluating LLMs as Agents," in Proc. Int. Conf. Learning Representations (ICLR), 2024.
20. S. Zhou et al., "WebArena: A Realistic Web Environment for Building Autonomous Agents," in Proc. Int. Conf. Learning Representations (ICLR), 2024.
21. G. Mialon, C. Fourrier, T. Wolf, Y. LeCun, and T. Scialom, "GAIA: A Benchmark for General AI Assistants," in Proc. Int. Conf. Learning Representations (ICLR), 2024.
22. C. E. Jimenez et al., "SWE-bench: Can Language Models Resolve Real-World GitHub Issues?" in Proc. Int. Conf. Learning Representations (ICLR), 2024.

23. C. Ma et al., “AgentBoard: An Analytical Evaluation Board of Multi-turn LLM Agents,” Advances in Neural Information Processing Systems, vol. 37, 2024, doi: 10.52202/079017-2365.
24. S. Kapoor, B. Stroebl, Z. S. Siegel, N. Nadgir, and A. Narayanan, “AI Agents That Matter,” Transactions on Machine Learning Research, 2025.
25. E. Debenedetti, J. Zhang, M. Balunović, L. Beurer-Kellner, M. Fischer, and F. Tramèr, “AgentDojo: A Dynamic Environment to Evaluate Prompt Injection Attacks and Defenses for LLM Agents,” Advances in Neural Information Processing Systems, vol. 37, 2024, doi: 10.52202/079017-2636.
26. Q. Zhan, Z. Liang, Z. Ying, and D. Kang, “InjecAgent: Benchmarking Indirect Prompt Injections in Tool-Integrated Large Language Model Agents,” Findings of the Association for Computational Linguistics: ACL 2024, pp. 10471–10506, 2024, doi: 10.18653/v1/2024.findings-acl.624.
27. H. Zhang et al., “Agent Security Bench (ASB): Formalizing and Benchmarking Attacks and Defenses in LLM-Based Agents,” in Proc. Int. Conf. Learning Representations (ICLR), 2025.
28. M. Andriushchenko et al., “AgentHarm: A Benchmark for Measuring Harmfulness of LLM Agents,” in Proc. Int. Conf. Learning Representations (ICLR), 2025.
29. L. Weidinger et al., “Taxonomy of Risks Posed by Language Models,” in Proc. ACM Conf. Fairness, Accountability, and Transparency (FAccT), 2022, pp. 214–229, doi: 10.1145/3531146.3533088.
30. B. Wang et al., “DecodingTrust: A Comprehensive Assessment of Trustworthiness in GPT Models,” Advances in Neural Information Processing Systems, vol. 36, 2023, doi: 10.52202/075280-1361.
31. P. Liang et al., “Holistic Evaluation of Language Models,” Transactions on Machine Learning Research, 2023.
32. L. Zheng et al., “Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena,” Advances in Neural Information Processing Systems, vol. 36, 2023, doi: 10.52202/075280-2020.
33. H. Weigand and P. Johannesson, “How to Identify your Design Science Research Artifact,” in 2023 IEEE 25th Conf. Business Informatics, 2023, pp. 1–10, doi: 10.1109/CBI58679.2023.10187511.
34. P. Antunes, N. H. Thuan, and D. Johnstone, “Nature and purpose of visual artifacts in design science research,” Information Systems and e-Business Management, vol. 20, pp. 515–550, 2022, doi: 10.1007/s10257-022-00559-2.
35. A. Hevner and S. Gregor, “Envisioning entrepreneurship and digital innovation through a design science research lens: A matrix approach,” Information & Management, vol. 59, no. 3, Art. no. 103350, 2022, doi: 10.1016/j.im.2020.103350.
36. Manish, “Sequential cooperative multi-agent online learning and adaptive coordination control in dynamic and uncertain environments,” 2026, doi: 10.1109/EIECC67963.2025.11409655.
37. Abhilash, Manish, and Prabhav, “Design patterns for deterministic low-latency distributed systems in modern application servers,” Jan. 2026, doi: 10.56726/IRJMETS88836.
38. Suneeth and Kamal, “Dynamic telemetry orchestration and signal elevation: An edge-native AI proxy architecture for industrial cyber-physical systems,” Oct. 2023, doi: 10.64882/ijrt.v11.i4.1097.