International Journal of Computer (IJC)
https://ijcjournal.org
ISSN: 2307-4523
Vol. 57, No. 1, 2026 | pp. 600-613

AgenticLens: An Intelligent Optimization Framework for Enterprise Agentic AI Systems Using Semantic Workflow Graphs, Federated Model Orchestration, and AI Economics

Nithesh Gudipuri*

Independent Researcher / Associate Director, Technology, St. Petersburg, Florida, USA
* *Corresponding author: nithesh.gudipuri@gmail.com | ORCID: 0009-0000-5732-3722*

Accepted: 20 May 2026 **Published:** 25 August 2026

Abstract

Large language models have accelerated the adoption of agentic artificial intelligence across enterprise software, enabling autonomous planning, reasoning, and execution at a scale that earlier automation could not achieve. Most current research emphasizes model capability and reasoning performance, while enterprise deployments confront a different set of constraints tied to inference cost, token efficiency, governance, latency, and operational reliability. These constraints intensify in mission-critical environments such as financial services, where deployed systems must satisfy demanding requirements for resilience, auditability, regulatory compliance, and predictable operational behavior. This paper introduces AgenticLens, an enterprise optimization framework for designing, analyzing, and continuously improving agentic AI systems. It treats enterprise AI as a coordinated ecosystem of interacting agents, retrieval components, reasoning workflows, and model execution paths. AgenticLens combines Semantic Workflow Graphs that represent agent interactions, Federated Model Orchestration that dynamically selects the most appropriate model for each task, and an AI Economics layer that weighs execution cost against quality, latency, and governance objectives. The framework introduces a system-level perspective for measuring and improving enterprise-agentic workflows, and it shifts the unit of optimization from an isolated prompt or a single model response to the coordinated ecosystem. The paper proposes a set of architectural metrics covering workflow efficiency, model utilization, token efficiency, execution latency, reasoning quality, and operational cost, and discusses benchmarking methodologies for comparing enterprise AI systems across diverse deployment scenarios. The paper is a conceptual contribution: it specifies a reference architecture and an evaluation design, it reports no measured results, and it identifies the empirical validation of the framework against fixed-model baselines as the necessary next step.

Keywords: agentic AI; AI economics; enterprise AI governance; federated model orchestration; LLM optimization; semantic workflow graphs; token efficiency; workflow optimization

ISSN: 2307-4523
Vol. 57, No. 1, 2026 | pp. 600-613

1. Introduction

Enterprise software has entered a phase in which large language models drive systems that plan, reason, and act with limited human supervision. Agentic architectures decompose a business request into subtasks, delegate those subtasks to specialized agents, and coordinate retrieval, tool invocation, and reasoning across many model calls. The multi-agent paradigm has matured quickly, and surveys of the field document a rapid expansion of frameworks for coordinating cooperating agents [3]. This maturation raises a question that model-centric research leaves open: the performance of a single model tells an incomplete story when dozens of coordinated calls compose a production workflow.

The dominant thread of research measures the capabilities of individual foundation models, and evaluation surveys catalog benchmarks and protocols that quantify reasoning, knowledge, and task competence [8]. Enterprise operators face pressures that these capability measures do not capture. Inference cost accumulates across every model call in a workflow, token consumption grows with each redundant reasoning step, latency compounds along the execution path, and governance obligations attach to every decision the system records. A workflow assembled from strong models can still waste budget, exceed latency targets, and produce audit gaps when its coordination is inefficient.

The economics of the situation compound the difficulty, because the pricing of commercial models spans orders of magnitude, and a workflow that assigns the most capable model to every subtask inflates the cost far past what the task mix requires. Work on cost reduction has shown that intelligent model selection recovers a large fraction of the wasted spend for an individual query [1], and studies of model serving reveal how heterogeneous hardware shifts the cost profile of a deployment [6]. These findings address the cost of a single call and the efficiency of a single server, leaving open the question of how the arrangement of many calls into a workflow affects the total. A workflow imposes its own overhead through redundant reasoning, unnecessary retrievals, and paths that revisit a decision already made, and this overhead escapes any analysis confined to a single call.

These pressures sharpen in regulated financial systems. Digital payments, securities processing, and transaction platforms operate under resilience, auditability, and compliance requirements that shape every deployment decision. Risk-based approaches to artificial intelligence governance impose graded obligations that vary with the sensitivity of the application, and the regulatory literature maps how these obligations extend to general-purpose systems, including large language models [9]. A financial institution that deploys an agentic system inherits these obligations for the full ecosystem of agents, and the cost of an inefficient or opaque workflow becomes a compliance concern in addition to an operational one. The obligation to record and justify each automated decision means that untraceable optimization carries regulatory exposure, which motivates the coupling of optimization with verification that the framework adopts.

ISSN: 2307-4523
Vol. 57, No. 1, 2026 | pp. 600-613

AgenticLens responds to this gap through a framework that optimizes the ecosystem of agents that compose an enterprise workflow. The framework rests on three coordinated mechanisms. Semantic Workflow Graphs represent the agents, retrieval components, and reasoning dependencies as an observable structure. Federated Model Orchestration selects a model for each task according to capability, cost, latency, and governance constraints. An AI Economics layer evaluates the execution of the workflow against combined objectives that no single dimension captures alone. Continuous behavioral verification closes the control loop and is intended to confirm that an optimization preserves the behavior the enterprise depends on.

The scope of the framework spans the design and operational phases of an agentic system. During design, the workflow graph provides an architect with a structure to reason about before deployment and surfaces inefficiencies that a code review alone would miss. During operation, the same graph feeds the optimization loop that adjusts the running system as its workload evolves. This dual scope distinguishes AgenticLens from a tool that analyzes a system once and reports findings, as the framework remains active throughout the deployment's lifetime. The framework also stays agnostic to the specific models and providers that populate the pool, and it accommodates the frequent turnover of the model landscape by treating each model as an interchangeable candidate that the orchestration layer evaluates against the same criteria. This construction protects an enterprise investment from the volatility of the underlying models, because a new model enters the pool through the same evaluation that governs the existing candidates.

This paper makes four contributions. First, it defines a system-level optimization framework that shifts the unit of analysis from an isolated model call toward the coordinated agentic ecosystem. Second, it specifies three coupled mechanisms that operate as a single control loop and function together as an integrated system. Third, it proposes a set of architectural metrics, along with a benchmarking methodology, for comparing enterprise agentic systems across deployment scenarios. Fourth, it presents a reference architecture suited to regulated environments, where auditability and predictable behavior carry equal weight with raw capability. The remaining sections review related work, describe the framework design methodology, present the proposed evaluation framework, discuss applicability and limitations, and conclude.

2. Related work

Research adjacent to AgenticLens divides into four streams, each addressing one facet of enterprise agentic AI without unifying them into a system-level optimization framework. The first stream targets the cost of model calls. FrugalGPT introduced cascades that query inexpensive models first and escalate to expensive models only when response quality falls short, thereby reducing inference cost while preserving answer quality [1]. Cascading optimizes a sequence of calls for a single query and treats each

ISSN: 2307-4523
Vol. 57, No. 1, 2026 | pp. 600-613

query as the unit of optimization.

The second stream studies routing across a pool of heterogeneous models. RouteLLM learns a router from preference data that assigns each query to a stronger or weaker model based on the query's difficulty, and the learned router lowers cost while maintaining response quality [2]. Cost-efficiency studies extend this idea to the infrastructure layer and characterize how model serving behaves across heterogeneous hardware [6]. These methods provide a control mechanism that AgenticLens incorporates, though they optimize selection for a single call and do not reason about the workflow surrounding the call.

The third stream develops the components that enterprise agentic systems assemble. Retrieval-augmented generation grounds model output in external knowledge and reduces hallucination for knowledge-intensive tasks, and the survey literature traces the progression from naive retrieval toward modular architectures [7]. Efficiency-focused surveys catalog techniques that compress models and accelerate inference, and organize methods that reduce the resource footprint of a deployed model [5]. These components form the nodes in a workflow graph, and optimizing them in isolation leaves their coordination unaddressed.

The fourth stream examines coordination and governance. Multi-agent surveys describe frameworks that orchestrate cooperating agents for complex tasks, and applied studies demonstrate hybrid agentic systems that combine strategic reasoning with efficient domain-specific execution in manufacturing settings [3], [4]. Work on governing agentic systems specifies practices for accountability and oversight when systems act with autonomy [10]. These frameworks supply the execution and the oversight machinery, and they leave the economic optimization of the coordinated workflow to the operator who assembles them.

A synthesis across the four streams reveals the gap that AgenticLens addresses. Each stream optimizes a narrow unit that comprises a query, a model call, a component, or a coordination pattern, and no stream treats the workflow as an economic object whose total cost, latency, and governance profile emerge from the arrangement of its parts. The evaluation literature reinforces this observation, as benchmarks that quantify model capability measure a model in isolation and do not assess the efficiency of workflows that compose many models [8]. AgenticLens draws the mechanisms from these streams into a single loop and adds the economic and verification layers that the streams omit. The unit of optimization becomes the coordinated ecosystem, and the objective becomes a combined criterion spanning the full workflow. The next section presents the framework and situates AgenticLens against the four streams along the dimension of the optimization unit each adopts.

ISSN: 2307-4523
Vol. 57, No. 1, 2026 | pp. 600-613

3. Materials and Methodology

The design of AgenticLens proceeds from the requirements imposed by regulated enterprise environments and derives its architecture from them. This section states the requirements, defines the system model, presents the overall architecture, and describes each mechanism together with the verification loop that binds them.

3.1. Design requirements

Five requirements govern the framework. First, auditability demands that every optimization decision leave a record that a compliance function can inspect. Second, cost predictability demands that the aggregate spend of a workflow remain bounded and explainable. Third, latency governance demands that the framework respect service-level objectives along the execution path. Fourth, regulatory compliance demands that model selection honor data residency and oversight constraints imposed by regulated workloads. Fifth, resilience demands that the framework degrade gracefully when a model or a component fails. These requirements shape the mechanisms that follow, and they explain why the framework couples optimization with verification.

3.2. System model

AgenticLens models an enterprise workflow as a set of agents that exchange data and delegate subtasks. An agent denotes a unit of autonomous behavior that receives inputs, performs reasoning or a tool call, and produces outputs. A workflow denotes an ordered composition of agents that transforms a business request into a result. An execution path denotes the specific sequence of agents that a given request traverses. A task class denotes a family of requests that share structural and cost characteristics. This vocabulary lets the framework reason about the ecosystem as a coordinated structure whose efficiency depends on the arrangement of agents and the routing of tasks among them.

The task class occupies a central place in the model because it supplies the granularity at which optimization decisions apply. A single workflow serves many task classes that differ in difficulty, and a routing decision that suits one class harms another when the framework applies it uniformly. The model partitions the request stream into classes and reasons about each class separately, which allows the orchestration layer to assign a model per class rather than a model per workflow. The cost characteristics of a class comprise the expected token consumption, the required capability level, and the latency budget that the class inherits from its business context, and the framework estimates these characteristics from the telemetry produced by the workflow graph. The estimation is expected to sharpen as the framework observes more requests, which would ground the optimization in the actual distribution of work that the enterprise processes rather than in an assumed one.

ISSN: 2307-4523
Vol. 57, No. 1, 2026 | pp. 600-613

3.3. Overall architecture

The framework arranges its mechanisms into a closed optimization loop that surrounds the agentic workflow, replacing the pipeline view in which optimization occurs once before deployment. The workflow executes at the center of the loop, Semantic Workflow Graphs observe its behavior and produce telemetry, the AI Economics layer evaluates that telemetry against combined objectives, Federated Model Orchestration acts on the workflow by adjusting model assignments, and continuous behavioral verification checks that each adjustment preserves the required behavior before the loop repeats. The framework is thus intended to improve the workflow as it runs and grounds each improvement in observed behavior. Fig. 1 depicts this arrangement.

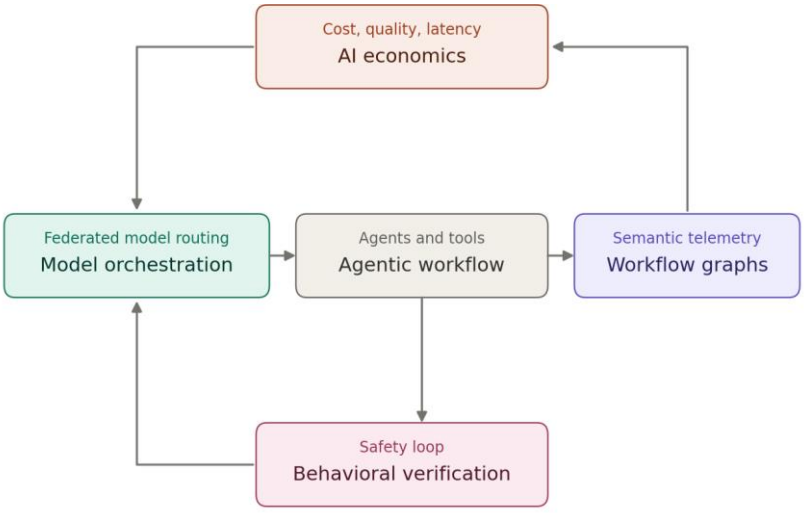

Figure 1: AgenticLens closed-loop optimization architecture

The loop's cadence separates AgenticLens from a one-time tuning pass, because it revisits its decisions as the workload evolves. A task class that a cheaper model served well can drift toward greater difficulty when the business context changes, and a static assignment would degrade quality without detection. The loop observes this drift through the telemetry that the graph produces, the economic layer reevaluates the assignment against the shifted distribution, and orchestration adjusts the assignment when the combined objective favors a change. The loop thus maintains optimization over time and treats it as an ongoing property of the running system. The interval between iterations becomes a design parameter because a short interval reacts to drift quickly but consumes more evaluation resources, while a long interval conserves resources while tolerating a longer period of suboptimal assignment. The framework exposes this interval to the operator, who tunes it based on the workload's volatility and the application's tolerance.

ISSN: 2307-4523
Vol. 57, No. 1, 2026 | pp. 600-613

3.4. Semantic Workflow Graphs

Semantic Workflow Graphs represent an enterprise workflow as a graph whose nodes denote agents, retrieval components, and tool invocations and whose edges denote data flows and reasoning dependencies. The graph provides the framework with an observable structure on which it can reason about efficiency, and it exposes patterns that a flat trace hides. A redundant edge that routes a request through an additional reasoning step without changing the outcome becomes visible as a structural feature, and the framework marks that edge as a candidate for optimization. The graph also reveals execution paths that consume disproportionate resources and provides the substrate on which the economic evaluation later operates. Fig. 2 illustrates a workflow graph in which a planner agent dispatches a request to retrieval and tool agents before a reasoning agent produces the response, and it highlights a redundant reasoning path that the framework flags.

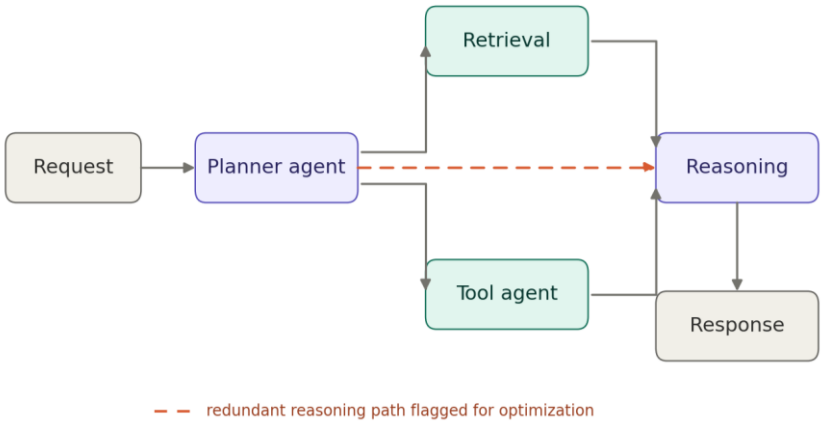

Figure 2: Semantic workflow graph with a flagged redundant reasoning path

3.5. Federated Model Orchestration

Federated Model Orchestration selects a model for each task based on the constraints it imposes. The orchestration layer evaluates candidate models against four criteria that comprise capability, cost, latency, and governance, and it assigns the least-cost model that satisfies the capability requirement of the task while honoring the residency and oversight constraints that regulated workloads impose. This mechanism generalizes learned routing between two models to a federated pool spanning providers and deployment locations [2]. The layer maintains a fallback policy that escalates a task to a more capable model when a quality signal indicates that the assigned model underperforms, and this policy borrows the escalation logic that cascades established [1]. Orchestration serves as the actuator of the control loop by translating the objectives computed by the economic layer into concrete model assignments within the workflow.

The federated construction of the layer distinguishes it from a router that selects among a fixed local

ISSN: 2307-4523
Vol. 57, No. 1, 2026 | pp. 600-613

pool. The pool spans providers, model families, and deployment locations, and a regulated workload may require a task to execute on a model hosted within a specific jurisdiction. The layer encodes these residency constraints as hard filters that remove ineligible models before it applies the cost criterion, and this ordering is intended to prevent a cost saving from overriding a compliance requirement. The layer also accounts for each candidate's operational profile, because a model that offers a low nominal price but a high tail latency can violate the service-level objective the task carries. Studies of serving over heterogeneous hardware supply the cost and latency profiles that this accounting consumes [6], and the layer combines those profiles with the residency filters to produce an assignment that satisfies the full constraint set.

3.6. AI Economics layer

The AI Economics layer evaluates a workflow against an objective that combines cost, quality, latency, and governance into a single criterion, and provides the decision signal that the orchestration layer acts on. The layer analyzes token utilization across the workflow and attributes consumption to specific execution paths, and this attribution locates the reasoning steps that consume budget without contributing to the outcome. The layer then weighs a proposed adjustment, such as reassigning a task class to a cheaper model, against its effect on quality and latency, and it accepts the adjustment when the combined objective improves. The economic framing lets the framework capture trade-offs that a single metric cannot represent, because a change that lowers cost while raising latency beyond a service-level objective fails the combined criterion even when it succeeds on cost alone.

3.7. Continuous behavioral verification

Continuous behavioral verification serves as the safety element of the control loop, confirming that an optimization preserves the behavior the enterprise depends on before the framework commits the change. The verification component compares the behavior of the adjusted workflow against a reference and is designed to block any adjustment that alters outcomes in a material field. This element aligns with the broader requirement that agentic systems operate under accountability and oversight, and the practices proposed for governing autonomous systems support the inclusion of an explicit verification stage [10]. The verification component also produces the audit record that the first design requirement demands, because each confirmed adjustment leaves evidence that a compliance function can inspect. The author has developed complementary behavioral verification methods in related patent work, and the framework treats verification as a first-class element of the optimization loop instead of an afterthought applied at deployment.

The four mechanisms function as one integrated system, because each depends on the outputs of the others. The graph provides observability, the economic layer provides the decision criterion,

ISSN: 2307-4523
Vol. 57, No. 1, 2026 | pp. 600-613

orchestration provides actuation, and verification provides the safety mechanism. An implementation that adopts a subset of the mechanisms would be expected to realize a correspondingly smaller share of the benefit, and the coupling among the four produces the system-level optimization that distinguishes the framework.

4. Proposed Evaluation Framework

AgenticLens provides a set of metrics and a benchmarking methodology that enable operators to measure and compare enterprise agentic systems, and this section presents them as a proposed evaluation methodology. The metrics quantify the dimensions the framework optimizes, and this section states their proposed definitions without reporting measured results, because the framework at this stage supplies a reference architecture and an evaluation design.

4.1. Architectural metrics

Six metrics characterize an enterprise agentic system, and each metric maps to a dimension that the framework optimizes. The metrics build on evaluation practice for language models, which distinguishes between what to evaluate and how to evaluate, and provide the rubric-based approach that the quality metric adopts [8]. Table 1 presents the metrics, their proposed definitions, and the dimension each addresses. The definitions treat the workflow as the unit of measurement and allow an operator to compare two systems that solve the same task class under distinct architectures.

Table 1: Proposed architectural metrics for enterprise agentic AI systems

Metric	Proposed definition	Dimension
Workflow efficiency	The ratio of value-producing steps to total execution steps traversed within the semantic workflow graph for a completed task	Structure
Model utilization	The share of tasks routed to the least-cost model that still satisfies the capability requirements of the task	Orchestration
Token efficiency	The volume of useful output tokens measured against the total tokens consumed across the workflow, counting redundant reasoning	Cost
Execution latency	The end-to-end and per-path completion time distribution observed across the workflow graph	Performance
Reasoning quality	The task-level correctness and consistency of outputs assessed under a defined evaluation rubric	Quality
Operational cost	The aggregate inference and infrastructure cost per completed workflow, normalized by workload class	Economics

The six metrics interact because an improvement in one dimension can degrade another. A change that raises token efficiency by removing a reasoning step can lower reasoning quality when that step contributed to correctness, and the combined economic objective captures this tension. The metrics

ISSN: 2307-4523
Vol. 57, No. 1, 2026 | pp. 600-613

support the economic evaluation performed by the framework and provide a quantitative foundation for the benchmarking methodology.

4.2. Benchmarking methodology

The proposed benchmarking methodology compares enterprise agentic systems across deployment scenarios through a controlled protocol. The protocol fixes a task class and a workload distribution, executes each candidate system against the same distribution, and records the six metrics over a defined evaluation window. The window aggregates metrics across many requests, producing a distribution for each metric rather than a single point, and this aggregation accommodates the variance that agentic workflows exhibit. The protocol then compares systems based on the combined economic objective and reports the metric distributions so an operator can inspect the trade-offs behind the aggregate. Cost-efficiency studies of model serving on heterogeneous hardware provide a precedent for this kind of controlled comparison, as they measure serving behavior under fixed workloads across distinct infrastructure [6].

The variance of agentic workflows deserves particular attention within the protocol, because a single request can traverse different execution paths depending on the content it carries and the state of the retrieval components it consults. A protocol that reported a single mean would obscure this dispersion and hide the tail behavior that matters most for a service-level objective. The methodology therefore records the full distribution and reports the tail statistics alongside the central tendency, and it repeats the evaluation across independent windows to separate systematic differences from sampling noise. This design draws on the evaluation practice for language models, which stresses that a robust comparison requires attention to the protocol as much as to the metric [8]. The protocol also fixes the retrieval corpus and tool endpoints across candidate systems, because differences in the external environment would confound comparisons of the systems themselves.

4.3. Illustrative application

A hypothetical document-processing workflow in a financial setting illustrates how the framework is intended to operate. The walkthrough that follows is illustrative rather than empirical, and no part of it was implemented, executed, or measured. The workflow receives a settlement instruction, retrieves reference data, extracts structured fields, reasons over compliance rules, and produces a validated record. AgenticLens constructs the workflow graph for this pipeline and observes that the reasoning agent invokes a large model for every field, even when a smaller model suffices for the routine fields. The economic layer attributes the token consumption to the reasoning path and identifies the routine fields as a task class that a cheaper model can serve. Federated Model Orchestration reassigns that task class to the smaller model; verification would confirm that the reassignment leaves the validated record

ISSN: 2307-4523
Vol. 57, No. 1, 2026 | pp. 600-613

unchanged in all material fields, and the framework commits the change. This walkthrough describes the mechanics qualitatively because operational metrics from the production environment were unavailable at the time of writing, and the framework prohibits substituting invented figures for measured results.

The same workflow exhibits a second inefficiency that the graph reveals: the retrieval component fetches reference data on every request, even when consecutive requests share the same counterparty and instrument. The graph represents this repeated retrieval as an edge that the framework can examine, and the economic layer attributes the latency and the cost of the redundant fetch to the retrieval path. Retrieval grounds model output in external knowledge and reduces error for knowledge-intensive tasks [7], and the value of retrieval justifies its presence in the workflow, while the redundancy of repeated fetches justifies their optimization. The framework introduces a caching decision for the task class that shares reference data, verification would confirm that the cached data matches the freshly fetched data within the required window, and the framework would commit the caching policy. The two optimizations are expected to compound, because the reassignment lowers the reasoning cost while the caching lowers the retrieval latency; whether their combined effect improves the economic objective by a margin that justifies the added machinery is an empirical question that the present paper does not answer.

5. Discussion

The value of AgenticLens follows from the system-level perspective it adopts, which explains why the optimization of an ecosystem differs from the sum of optimizations applied to its parts. A workflow assembled from individually optimized models can still route requests inefficiently, duplicate reasoning, and exceed latency targets when its coordination goes unexamined. The framework optimizes coordination and captures interactions among components that a component-level analysis omits.

The framework extends beyond financial services because the requirements that motivate it are common across regulated and large-scale domains. Healthcare deployments carry auditability and compliance obligations that mirror the financial case; manufacturing deployments combine strategic reasoning with efficient domain-specific execution in a manner that hybrid agentic systems already demonstrate [4]; government deployments demand predictable and accountable behavior; and telecommunications deployments operate at a scale where cost and latency dominate. The domain-agnostic design of the framework is meant to accommodate each of these settings, though the present work examines none of them empirically, and the financial motivation provides concrete examples without narrowing its intended applicability.

AgenticLens sits above existing orchestration frameworks as a layer. Multi-agent frameworks supply

ISSN: 2307-4523
Vol. 57, No. 1, 2026 | pp. 600-613

the execution substrate that coordinates cooperating agents [3], and AgenticLens adds the observability, economic evaluation, and verification that turn execution into optimization. The framework complements the orchestration tools enterprises already use and extends them with a system-level optimization loop.

The governance posture of the framework warrants closer scrutiny, as the verification loop produces an artifact that a compliance function can consume. Each confirmed adjustment generates a record that states the change, the objective that motivated it, and the verification result that cleared it, providing the audit trail a regulated environment requires. Practices for governing agentic systems call for exactly this kind of traceable oversight when a system acts autonomously [10], and AgenticLens responds to this call by making verification a precondition for every change. The framework thereby aligns the economic objective with the governance objective, because a change that improves cost yet fails verification is not intended to reach production. This alignment matters most in the financial setting that motivates the work, where an untraceable optimization would trade an operational gain for a regulatory liability, and the framework is designed to avoid that trade.

The framework is designed on the expectation that its return grows with the scale of the deployment, since a larger request volume would amplify the effect of each optimization the framework commits. A reassignment that lowered the cost of a single task class would apply that reduction across every request in the class, so a per-request saving, if one were realised, would aggregate over high request volumes; the magnitude of any such saving has not been measured and would depend on the workload mix. The framework is therefore aimed at the high-volume workloads that dominate financial and telecommunications settings, on the assumption that its overhead amortizes across the volume it optimizes, which the evaluation outlined below is designed to test. An incremental adoption path lets an enterprise introduce the framework without a wholesale redesign, because the observability layer attaches to an existing workflow and produces the graph before any optimization applies. The operator can then enable orchestration for a single task class, measure whether it yields a net benefit, and extend the scope as confidence accrues.

Several limitations bound the present work, and the most consequential is the absence of empirical evidence. The framework at this stage provides a reference architecture that has not been implemented, deployed, or compared against a fixed-model baseline, so every efficiency, latency, and quality benefit attributed to it in this paper remains an architectural hypothesis rather than a measured result, and the quantitative behavior of the metrics remains to be measured on production workloads. The quality of orchestration depends on the accuracy of task profiling, because a misclassified task can receive a model that underperforms, and the verification loop mitigates this risk without eliminating it. The observability layer introduces its own overhead, and an implementation must balance the granularity of the workflow

ISSN: 2307-4523
Vol. 57, No. 1, 2026 | pp. 600-613

graph against the cost of maintaining it. A further limitation concerns the stability of the model pool: a provider changing a model under a fixed identifier can shift the behavior of an assignment that verification previously confirmed, and the framework must re-verify assignments when the pool changes. Future work will address the absence of evidence directly through a proof-of-concept evaluation. The planned study implements the framework over a small set of representative workflows, which comprise structured field extraction from financial documents, retrieval-augmented question answering over a policy corpus, and multi-step tool-using planning, and compares each against a fixed-model baseline that assigns a single high-capability model to every step. The study will report operational cost per completed request, end-to-end latency at the median and at the tail, input and output token consumption, and output quality assessed under a defined rubric with human review of a sample, and it will include ablations that separate the contribution of routing, caching, and the verification gate. Subsequent work will validate the metrics on production workloads, construct reference datasets of agentic workflows, and integrate the verification loop with the governance practices required in regulated environments.

6. Conclusion

This paper presents AgenticLens, a framework that optimizes enterprise agentic AI systems at the level of the coordinated ecosystem rather than the individual model. The framework unites Semantic Workflow Graphs, Federated Model Orchestration, and an AI Economics layer within a closed optimization loop in which continuous behavioral verification is intended to make automated adjustment acceptable in regulated environments.

The framework contributes a system-level perspective, three coupled mechanisms, a set of architectural metrics with a benchmarking methodology, and a reference architecture for auditable enterprise deployment. The next generation of enterprise AI platforms will distinguish themselves through the capacity to optimize intelligence, and AgenticLens establishes a foundation on which future research in enterprise-scale agentic optimization can build.

Declarations

Ethics approval and consent to participate: Not applicable.

Consent for publication: Not applicable.

Funding: The authors received no external funding for this work.

Competing interests: The authors declare that they have no competing interests.

Author contributions: All authors contributed to this research and approved the final manuscript.

References

ISSN: 2307-4523
Vol. 57, No. 1, 2026 | pp. 600-613

[1] L. Chen, M. Zaharia, and J. Zou, "FrugalGPT: how to use large language models while reducing cost and improving performance," arXiv preprint arXiv:2305.05176, May 2023, doi: 10.48550/arXiv.2305.05176.

[2] I. Ong, A. Almahairi, V. Wu, W.-L. Chiang, T. Wu, J. E. Gonzalez, M. W. Kadous, and I. Stoica, "RouteLLM: learning to route LLMs from preference data," in Proc. Int. Conf. Learn. Represent. (ICLR), Singapore, 2025, pp. 34433–34448, doi: 10.48550/arXiv.2406.18665.

[3] T. Guo, X. Chen, Y. Wang, R. Chang, S. Pei, N. V. Chawla, O. Wiest, and X. Zhang, "Large language model based multi-agents: a survey of progress and challenges," in Proc. 33rd Int. Joint Conf. Artif. Intell. (IJCAI), Jeju, South Korea, Aug. 2024, pp. 8048–8057, doi: 10.24963/ijcai.2024/890.

[4] M. A. Farahani, M. I. Khan, and T. Wuest, "Hybrid agentic AI and multi-agent systems in smart manufacturing," J. Manuf. Syst., vol. 86, pp. 612–623, Jun. 2026, doi: 10.1016/j.jmsy.2026.04.002.

[5] J. Cheng, H. Kang, Y. Shao, N. Li, P. Chen, R. Wang, et al., "Survey on efficient large language models: principles, algorithms, applications, and open issues," IEEE Trans. Neural Netw. Learn. Syst., vol. 37, no. 5, pp. 2025–2045, May 2026, doi: 10.1109/TNNLS.2025.3628671.

[6] Y. Jiang, F. Fu, X. Yao, G. He, X. Miao, A. Klimovic, B. Cui, B. Yuan, and E. Yoneki, "Demystifying cost-efficiency in LLM serving over heterogeneous GPUs," in Proc. 42nd Int. Conf. Mach. Learn. (ICML), PMLR, vol. 267, Vancouver, Canada, Jul. 2025, pp. 27534–27552, doi: 10.48550/arXiv.2502.00722.

[7] Y. Gao, Y. Xiong, X. Gao, K. Jia, J. Pan, Y. Bi, et al., "Retrieval-augmented generation for large language models: a survey," arXiv preprint arXiv:2312.10997, 2023, rev. Mar. 2024, doi: 10.48550/arXiv.2312.10997.

[8] Y. Chang, X. Wang, J. Wang, Y. Wu, L. Yang, K. Zhu, et al., "A survey on evaluation of large language models," ACM Trans. Intell. Syst. Technol., vol. 15, no. 3, Art. no. 39, pp. 39:1–39:45, 2024, doi: 10.1145/3641289.

[9] T. Szadeczky and Z. Bederna, "Risk, regulation, and governance: evaluating artificial intelligence across diverse application scenarios," Secur. J., vol. 38, Art. no. 35, 2025, doi: 10.1057/s41284-025-00495-z.

[10] Y. Shavit, S. Agarwal, M. Brundage, et al., "Practices for governing agentic AI systems," OpenAI, white paper, Dec. 14, 2023.