COMPETING PATHS TO ENTERPRISE AUTONOMY

A Critical Evaluation of Architectural Paradigms

for Autonomous Enterprise AI

Abhimanyu Singhal
Zustis Technologies Limited

March 2026

ABSTRACT

The enterprise AI discourse has fractured into competing architectural camps—large language models, neuro-symbolic hybrids, multi-agent swarms, active inference, and runtime assurance—each claiming primacy on the path to autonomous operations. This paper argues that the debate is structurally misconceived. No single paradigm resolves the fundamental tension between probabilistic capability and deterministic safety that enterprise autonomy demands. The connectionist paradigm offers unmatched linguistic fluency but cannot guarantee the correctness of a single action. The neuro-symbolic paradigm provides the auditability enterprises require but shifts the cost center from training compute to knowledge engineering—a trade that most organizations are unprepared to make. Multi-agent architectures promise modularity but introduce coordination overhead and compounding error dynamics that convert small uncertainties into cascading failures. Active inference offers the most principled formalism for persistent state awareness but remains years from production readiness. Runtime assurance can formally bound behavior, but verification technology cannot yet reach the scale of production models. The resolution is not selection but composition: the enterprise that will dominate is not the one deploying the largest model, but the one assembling the most robust architecture from complementary paradigms—with safety guarantees enforced at the infrastructure layer, not the prompt layer. This paper examines each paradigm on its own terms, identifies the structural ceilings that prevent any from standing alone, and proposes a framework for evaluating hybrid compositions against the six axes that determine enterprise deployability: epistemic reliability, action safety, auditability, scalability, security, and operational robustness.

Keywords: *autonomous agents; enterprise AI architecture; neuro-symbolic AI; active inference; multi-agent systems; runtime assurance; compound AI systems; knowledge graphs; LLM security; formal verification*

1. The Structural Problem: Why No Single Paradigm Suffices

The contemporary enterprise faces a deceptively simple question: how should an artificial intelligence system be architected to act autonomously in mission-critical workflows? The question is deceptive because each of the five

dominant paradigms—connectionist, neuro-symbolic, multi-agent, active inference, and runtime-assured—offers a compelling answer that is also, on its own, fundamentally incomplete. The incompleteness is not a matter of engineering immaturity that will be resolved by the next model release. It is structural, rooted in formal limitations of computation and verification that no amount of scaling can overcome.

The core tension can be stated precisely. Enterprise autonomy requires two properties simultaneously: broad capability (the ability to handle the full diversity of enterprise tasks, including novel and ambiguous situations) and bounded correctness (the guarantee that actions will not violate safety, compliance, or business constraints). These properties are in tension because the mechanisms that maximize one tend to undermine the other. Neural networks achieve broad capability precisely because they are unconstrained by explicit rules—but this freedom is the source of hallucination. Symbolic systems achieve bounded correctness precisely because they are constrained by explicit rules—but this rigidity is the source of brittleness. The multi-agent paradigm attempts to decompose the problem, but decomposition introduces coordination failure modes that can be worse than the monolithic failures they replace.

This paper evaluates each paradigm not as a technology to be benchmarked but as a design philosophy with inherent trade-offs. The evaluation is anchored on six enterprise-relevant axes: epistemic reliability (how the system distinguishes truth from uncertainty), action safety (how it constrains behavior when interfacing with real systems), auditability (whether decisions can be traced to rules and evidence), scalability (cost to adapt to new policy and distribution shift), security (susceptibility to adversarial manipulation), and operational robustness (how failures propagate across time and across modules). A recurring finding across all paradigms is that the boundary between probabilistic inference and correctness guarantees is not a technical inconvenience—it is the central architectural decision that determines whether an enterprise deploys a capable assistant or a trustworthy autonomous agent.

2. The Connectionist Paradigm: Capability Without Guarantees

2.1 What Scale Actually Delivers

The pure connectionist approach—large Transformer networks trained end-to-end on massive corpora—has produced the most commercially visible AI systems of the current era. The Transformer's self-attention mechanism, introduced by Vaswani et al. (2017), enables powerful sequence modeling without recurrence, and scaling laws documented by Kaplan et al. (2020) and refined by Hoffmann et al. (2022) demonstrate that capabilities improve predictably with model size and training compute. The practical result is systems with remarkable linguistic fluency, few-shot generalization, and flexible instruction following.

The enterprise-critical extension is tool use. Yao et al.'s ReAct framework (2023) demonstrated that interleaving reasoning traces with action steps substantially reduces hallucination while boosting interactive task success. Schick et al.'s Toolformer (2023) showed that a 6.7-billion-parameter model, self-taught to invoke APIs, could outperform GPT-3 at 175 billion parameters on question answering tasks. By 2025, open-source models in the 7–32 billion parameter range routinely exceed 70% pass rates on tool-use benchmarks. The implication for enterprises is significant: tool-augmented language models can translate natural language goals into sequences of actions across enterprise systems—querying databases, calling functions, synthesizing results—shifting automation from rigid workflow scripting toward dynamic orchestration.

2.2 Three Structural Ceilings

Despite these advances, three structural limitations prevent the connectionist paradigm from supporting enterprise autonomy on its own.

The first is hallucination. The phenomenon is not a rare edge case amenable to engineering fixes; it is a central reliability barrier inherent to probabilistic text generation. Comprehensive surveys in ACM Computing Surveys (Huang et al., 2025) establish taxonomies covering input-conflicting, context-conflicting, and fact-conflicting hallucinations. More troubling, recent adversarial evaluations show that models across the parameter spectrum—from 2 billion to 27 billion parameters—exhibit hallucination rates exceeding 80% when exposed to symbolic modifiers, named entities, and negation triggers. Detection methods have improved, but multiple research groups characterize hallucination as an intrinsic property of the architecture rather than a bug to be patched. For enterprises, the consequence is categorical: a system that hallucinates even 1% of the time cannot be trusted with autonomous write-back operations to ERP systems, financial ledgers, or industrial controls.

The second is long-context fragility. Liu et al.'s "Lost in the Middle" study, published in Transactions of the ACL (2024), reveals that language models exhibit a U-shaped performance curve: accuracy is highest when relevant information appears at the beginning or end of the context window, degrading by over 30% for information positioned in the middle. This is not merely an academic curiosity. Enterprise multi-step workflows accumulate state precisely in the middle of long contexts—the invoices processed, the approvals obtained, the exceptions handled. The positional bias means that the very information most critical to maintaining operational coherence is the information most likely to be ignored.

The third is security. A landmark joint study by researchers from OpenAI, Anthropic, and Google DeepMind (Nasr, Carlini et al., October 2025) tested twelve published defenses against adaptive attacks. Every defense was bypassed, with attack success rates above 90% for most. The UK's National Centre for Cyber Security subsequently characterized LLMs as "inherently confusable deputies," suggesting that prompt injection may never be fully mitigated at the model level. The Agent Security Bench (ICLR 2025) reported average attack success rates of 84.30% across ten enterprise scenarios. The enterprise implication is stark: the moment a language model is granted tool access—the moment it transitions from text generator to operational controller—the attack surface expands from prompt manipulation to full system compromise. Secure-by-design orchestration is not a best practice; it is a survival requirement.

2.3 The Bottom Line

The pure connectionist architecture is the fastest path to broad capability, especially for language-heavy workflows involving emails, tickets, policy documents, and incident reports. But the moment the system crosses into high-stakes autonomous execution, the enterprise must treat the base model as an unreliable policy proposer—a source of action candidates that must be grounded by retrieval, constrained by symbolic policies, and sandboxed by verified safety monitors. The model proposes; the architecture disposes.

3. The Neuro-Symbolic Paradigm: Compliance by Construction

3.1 The Separation of Concerns

The neuro-symbolic paradigm, framed compellingly by Garcez and Lamb (2023) through Kahneman's System 1/System 2 distinction, synthesizes neural perception with symbolic reasoning. The neural component processes high-dimensional unstructured data—identifying defects in video feeds, extracting clauses from legal documents, detecting anomalous patterns in sensor telemetry. The symbolic component receives these perceptual outputs and

applies them against a formal knowledge representation: ontologies, rules, knowledge graphs. When a neural network detects a surface scratch with 98% confidence, the symbolic reasoner queries the knowledge graph to determine that the scratch violates Quality Standard B, which triggers Reject Protocol C. This architecture provides the audit trail that is the prerequisite for removing the human from the operational loop.

The foundational survey by Hitzler et al. in the National Science Review (2022) established the structured overview of the field, and a PRISMA-based systematic review (Colelough and Regli, 2025) identified over 1,400 papers published between 2020 and 2024. Differentiable approaches—Neural Logic Programming, differentiable theorem proving, and DeepProbLog's neural predicates—demonstrate that the neural–symbolic boundary is becoming increasingly fluid. The convergence of LLMs with knowledge graphs has proven particularly impactful: Microsoft's GraphRAG achieves 72–83% comprehensiveness on global queries, and benchmarks demonstrate 3.4× accuracy improvement over traditional retrieval-augmented generation for complex multi-hop reasoning tasks.

3.2 The Enterprise Alignment

Three properties make neuro-symbolic architectures structurally aligned with enterprise requirements. First, compliance by construction: if enterprise autonomy must never violate certain constraints—segregation of duties, financial control rules, safety interlocks—symbolic policy enforcement provides a natural mechanism. The system refuses actions that violate formally encoded rules regardless of the neural network's confidence score. This moves governance from post-hoc audit to pre-emptive architectural constraint. Second, auditability: knowledge-graph-based approaches provide explicit relational structure that can be referenced in explanations—"what rule fired," "what entity relationship mattered," "what constraint was binding." Third, rule-change resilience: enterprises face constant rule churn from regulatory updates, new products, and reorganized controls. Neuro-symbolic designs can update symbolic knowledge without retraining large models, shifting part of the maintenance burden from expensive compute to knowledge management.

3.3 The Knowledge Engineering Bottleneck

The limiting reagent for neuro-symbolic systems is not model capacity but knowledge engineering capacity. The literature on expert systems identified knowledge acquisition as the fundamental bottleneck decades ago, and the problem persists in modern forms. Constructing a comprehensive enterprise ontology—mapping sensors to materials, materials to constraint thresholds, thresholds to remediation protocols—is an enormous undertaking. The labor economics of ontology creation—who builds it, who validates it, who keeps it current as the business evolves—remain largely unresolved.

Symbolic layers are strongest when rules are crisp. But enterprise reality contains ambiguous concepts—"reasonable," "material," "suspicious"—evolving taxonomies, and exceptions. When the symbolic layer is incomplete or misspecified, the system fails abruptly, often more catastrophically than a probabilistic model that degrades gracefully. The neuro-symbolic paradigm thus trades one cost center for another: instead of training compute, the enterprise pays for ontology lifecycle management. Most organizations underestimate this trade.

4. The Multi-Agent Paradigm: Modularity at the Cost of Coordination

4.1 The Decomposition Thesis

Multi-agent architectures distribute cognition across multiple interacting components—specialized models, planners, critics, monitors, and tool-execution modules that coordinate to solve larger tasks. The framework landscape has consolidated by late 2025 into three dominant platforms: CrewAI for role-based collaboration, LangGraph for graph-based state machines, and Microsoft's Agent Framework merging AutoGen with Semantic Kernel. The theoretical foundation rests on multi-agent reinforcement learning (MARL), comprehensively surveyed in Artificial Intelligence Review, where multiple decision-makers interact under partial observability, non-stationarity, and strategic dependence.

The enterprise appeal is modularity. A multi-agent system can isolate compliance checks, document extraction, reconciliation, and planning into separately maintained components—compatible with organizational structures where different teams own different controls, tools, and risk boundaries. The supervisor-worker pattern has emerged as the most production-ready architecture, with a supervisor agent decomposing goals and delegating to specialist workers with restricted tool sets.

4.2 The Compounding Error Problem

The fundamental theoretical challenge is compounding errors. Ross, Gordon, and Bagnell (2011) proved that behavior cloning in sequential decision-making produces errors that grow quadratically with the horizon—up to εT^2 mistakes for a T-step task—which their DAgger algorithm reduces to linear growth through iterative expert-corrected aggregation. In multi-agent LLM systems, this manifests as cascading hallucinations: a minor fabrication from a research agent becomes accepted fact for an execution agent, causing errors to snowball into complete system failure. The MAST taxonomy (UC Berkeley et al., 2025) identifies fourteen unique failure modes across three categories: role ambiguity, step repetition leading to infinite loops, and context loss causing system restarts.

A critical finding from recent scaling studies challenges the multi-agent orthodoxy. Experiments across 180 configurations demonstrate a "tool-coordination trade-off": multi-agent systems fragment the per-agent token budget, leaving insufficient capacity for complex tool orchestration. When baseline single-agent performance is already high, coordination overhead becomes a net cost. This does not invalidate multi-agent architectures for complex workflows requiring genuine specialization, but it does suggest that the reflexive move to "add more agents" is often counterproductive.

4.3 The Security Multiplier

Security vulnerabilities scale with topological complexity. Research submitted to ICLR 2026 found that "increasing topological complexity of LLM multi-agent systems does not guarantee security; risks are distributed across agents." Defenses designed for single-agent prompt injection do not reliably transfer to multi-agent settings, and narrowly scoped defenses may inadvertently increase vulnerabilities in adjacent agents. Every inter-agent interface is a seam, and seams are where attackers and failures concentrate: authentication boundaries, schema mismatches, weak validation, and tool misuse.

5. The Active Inference Paradigm: Principled Uncertainty, Practical Nascency

5.1 The Theoretical Elegance

Active inference, grounded in Karl Friston's Free Energy Principle (2010), reframes agency as a process of minimizing variational free energy—an upper bound on surprisal. The agent maintains a generative model of its

environment, continuously comparing predictions with incoming sensory data. The discrepancy constitutes prediction error, and the agent minimizes it through two pathways: perceptual inference (updating beliefs to accommodate new evidence) and active inference (acting upon the world to bring reality into alignment with expectations). This recursive loop creates persistent state awareness—the agent maintains a continuous, recursive understanding of system health rather than processing discrete transactions.

The framework's enterprise appeal lies in its treatment of the exploration–exploitation problem. Expected Free Energy decomposes into pragmatic value (goal pursuit) and epistemic value (uncertainty reduction), providing a built-in exploration–exploitation balance without ad-hoc exploration bonuses. For enterprise settings closer to cyber-physical operations than to chat—industrial monitoring, robotics, adaptive process control—this formalism offers the most coherent available blueprint for agents that proactively seek information to resolve uncertainty.

5.2 The Honest Assessment

Intellectual honesty requires stating what the active inference community has been reluctant to acknowledge publicly. Beren Millidge, a key contributor to the field, provided a candid retrospective in 2024: active inference is "essentially isomorphic to RL" and offers "relatively little special sauce" above standard deep RL methods when distributions are parameterized by neural networks and optimized via black-box variational inference. This does not render active inference useless—its explicit generative model and belief-state formalism provide genuine advantages for interpretability and for domains requiring persistent state tracking—but it does deflate claims of paradigmatic superiority.

Computational tractability remains the primary barrier. Standard discrete POMDP formulations face combinatorial explosion in policy evaluation. Recent advances—Bellman-style decompositions by Paul et al. (2024) and Bethe factor graph approaches by Nuijten et al. (2025)—reduce complexity from exponential to polynomial, but neither has been validated at enterprise scale. Software implementations serve the research community (pymdp, RxInfer.jl, ActiveInference.jl) but lack enterprise engineering investment. The technology readiness level stands at approximately TRL 3–4, with no known production enterprise deployments. The claim of this paper is not that active inference has been proven at industrial scale, but that it provides the most coherent available formalism for persistent state awareness—a hypothesis that warrants rigorous empirical testing.

6. Runtime Assurance: The Missing Layer

6.1 The Simplex Principle

A major gap in most architectural paradigm discussions is that enterprises often need hard safety bounds even when the core intelligence is probabilistic. The Simplex architecture, introduced by Sha and colleagues in the control systems literature (1998, 2001), formalizes this requirement: a high-performance but unverified advanced controller operates alongside a verified-safe baseline controller, with a decision module switching to the baseline when safety constraints are threatened. This is not a theory; it is a deployed engineering pattern with extensions for neural network controllers (Neural Simplex Architecture, Phan et al., 2020), black-box systems requiring no static baseline verification (Black-Box Simplex, Mehmood et al., 2022), and systems with physical uncertainties (L1Simplex, Wang et al., 2013).

6.2 The Verification Gap

Formal verification of neural networks has advanced through five consecutive VNN-COMP victories by α,β-CROWN (Zhang et al.), which supports feedforward, convolutional, residual, transformer, and LSTM architectures. Marabou 2.0 achieved a 60× preprocessing speedup with proof production. Yet the scalability gap is stark: complete verification operates on networks with thousands to low tens of thousands of neurons, while enterprise LLMs contain billions of parameters—a three-to-six order-of-magnitude gap that no current approach can bridge. This impossibility has driven the emergence of pragmatic LLM-specific runtime assurance: lightweight constraint languages for agent behavior (AgentSpec, ICSE 2026), intent-formalized safety monitors (VeriGuard, 2025), and category-specific Markov Logic Network detectors (R2-Guard, ICLR 2025).

6.3 The Enterprise Translation

The emerging enterprise pattern directly mirrors Simplex's dual-controller design. Fast rule-based checks operating in microseconds serve as the first filter; machine learning classifiers operating in milliseconds form the second layer; LLM-as-judge operating in seconds serves as the final arbiter, with escalation only when needed. This architecture is paradigm-agnostic: it can wrap connectionist agents, constrain multi-agent swarms, and provide certified fallbacks for active inference controllers. Runtime assurance does not replace intelligence; it makes intelligence deployable. The enterprise that embeds safety at the infrastructure layer—rather than hoping the model will behave—is the enterprise that can grant its agents write access to production systems.

Table 1. Comparative Analysis of Architectural Paradigms for Enterprise Autonomy

7. The Convergence Thesis: Compound Hybrid Architectures

7.1 The Industry Pattern

The practical enterprise landscape has converged on what Zaharia et al. at Berkeley (2024) termed "compound AI systems"—multi-component architectures combining models, retrievers, tools, and symbolic validators. Databricks reports that 60% of LLM applications already use retrieval-augmented generation and 30% employ multi-step chains. McKinsey's 2025 global survey found 62% of enterprises experimenting with AI agents, though only 23% have scaled deployment to at least one business function. Gartner predicts 40% of enterprise applications will feature task-specific agents by 2026 but warns that over 40% of agentic AI projects will be cancelled by end of 2027 due to escalating costs, unclear value, or inadequate risk controls.

These numbers reveal a field in productive tension between ambition and pragmatism. The most instructive data point comes from MIT Sloan's deployment study: when building an AI agent for cancer patient adverse event detection, 80% of the work was consumed by data engineering, stakeholder alignment, governance, and workflow integration—not prompt engineering. The message is clear: the hard problem of enterprise autonomy is not building the model; it is building the architecture around the model.

7.2 The Composition Principle

The most robust enterprise architectures are hybrid in substance even when pure in marketing. The composition principle that emerges from this analysis assigns each paradigm to its natural layer. Connectionist models serve as the perception and language layer—ingesting unstructured data, generating action candidates, and handling the linguistic variability of enterprise communications. Symbolic reasoning serves as the constraint and traceability layer—the digital constitution that restricts the agent's action space to what is legally permissible and physically possible. Multi-agent decomposition serves as the organizational layer—isolating concerns, enabling independent

maintenance, and mapping to enterprise team structures. Active inference, where appropriate, serves as the continuous monitoring layer—maintaining persistent state awareness for cyber-physical operations. Runtime assurance serves as the safety layer—the infrastructure-level enforcement mechanism that bounds behavior regardless of what the intelligence layer proposes.

This layered composition is not merely pragmatic eclecticism. It reflects a deeper insight: enterprise autonomy requires a separation of epistemic functions analogous to the separation of powers in constitutional governance. The entity that proposes action must not be the same entity that validates action, which must not be the same entity that monitors execution. When these functions are collapsed into a single model, the enterprise loses the ability to independently verify any one of them.

Table 2. The Separation of Epistemic Functions in Hybrid Architecture

8. Limitations and Open Questions

Intellectual honesty requires acknowledging the distance between the analysis presented here and the engineering realities of production deployment. Several limitations merit explicit discussion.

The integration problem. This paper identifies the natural layer for each paradigm, but the interfaces between layers remain an open engineering frontier. How a symbolic constraint engine communicates efficiently with a neural perception layer—without introducing latency, losing information, or creating new attack surfaces—is a systems integration challenge that no current framework fully resolves. The field does not yet offer a drop-in "universal neuro-symbolic stack" that works across enterprise domains without substantial customization.

The empirical validation gap. The paradigm evaluations in this paper draw on peer-reviewed research, but much of that research operates on academic benchmarks rather than production enterprise environments. The transition from benchmark performance to operational reliability frequently reveals failure modes that theoretical analysis cannot anticipate. The 84.30% attack success rate from the Agent Security Bench is measured in a controlled setting; the real-world attack surface of a deployed enterprise agent is likely broader and more heterogeneous.

The cost of composition. Hybrid architectures are more robust in theory, but they are also more expensive to build, maintain, and operate. Each additional layer introduces engineering overhead, operational complexity, and potential failure modes at the interfaces. The enterprise that cannot afford a dedicated knowledge engineering team should not adopt a neuro-symbolic layer. The enterprise that lacks formal methods expertise should not attempt to build custom runtime assurance monitors. The composition principle is sound, but its implementation requires honest assessment of organizational capability.

Liability and accountability. When an autonomous agent executes a decision that produces harm—a misrouted shipment, an incorrectly enforced penalty, a sanctioned transaction that slips through—the question of legal liability is largely uncharted. Existing legal frameworks assume human decision-makers. The regulatory infrastructure required to support autonomous enterprise agency does not yet exist in most jurisdictions, and standards development (ISO/IEC 42001, EU AI Act) remains in early stages.

9. Conclusion: The Architecture Is the Product

The 2025–2026 enterprise AI landscape reveals a field that has outgrown the question "which model should we use?" and arrived at the more consequential question: "what architecture makes autonomous action safe enough to deploy?" This paper's analysis yields three insights that transcend the paradigm comparison.

First, the security impossibility result—joint research from the leading AI laboratories demonstrating that all model-level defenses are bypassable at greater than 90% success rates—means that enterprise safety must migrate from model-level defenses to infrastructure-level enforcement. This mirrors the Simplex principle: verified external monitors, not trusted internal behavior. The enterprise that treats safety as a prompt engineering problem will be the enterprise that suffers the first catastrophic autonomous failure.

Second, the convergence of inference-time scaling laws with compound system architectures creates a new optimization frontier. Smaller, cheaper models with more test-time compute, orchestrated across specialized agents with symbolic validators, can outperform monolithic frontier models at a fraction of the cost. The economics of enterprise AI are shifting from "pay for the biggest model" to "compose the most efficient architecture."

Third, the separation of epistemic functions—perception, constraint enforcement, decomposition, state awareness, and safety bounding—is not an implementation detail but the core architectural principle. The entity that proposes action must not be the entity that validates it. The entity that validates must not be the entity that monitors execution. This separation is what distinguishes an enterprise deploying a capable assistant from an enterprise deploying a trustworthy autonomous agent.

The winners of the coming decade will not be the organizations with the most capable chatbots. They will be the organizations with the most robust architectures—where connectionist fluency is grounded in symbolic truth, where multi-agent modularity is disciplined by coordination engineering, where active inference maintains persistent state awareness, and where runtime assurance provides the constitutional boundary that makes autonomy safe enough to trust. The model is a component. The architecture is the product.

References

Colelough, R. & Regli, W. (2025). Neuro-Symbolic AI in 2024: A Systematic Review. arXiv:2501.05435.

Friston, K. (2010). The Free-Energy Principle: A Unified Brain Theory? Nature Reviews Neuroscience, 11(2), 127–138.

Friston, K., FitzGerald, T., Rigoli, F., Schwartenbeck, P., & Pezzulo, G. (2017). Active Inference: A Process Theory. Neural Computation, 29(1), 1–49.

Garcez, A. d'A. & Lamb, L. C. (2023). Neurosymbolic AI: The 3rd Wave. Artificial Intelligence Review, 56(11), 12387–12406.

Hitzler, P., Sarker, M. K., & Eberhart, A. (2022). Neuro-Symbolic Artificial Intelligence: Current Trends. National Science Review, 9(6), nwac035.

Hoffmann, J. et al. (2022). Training Compute-Optimal Large Language Models. Proceedings of NeurIPS.

Huang, L. et al. (2025). A Survey on Hallucination in Large Language Models. ACM Transactions on Information Systems.

Kandogan, E. et al. (2024). A Blueprint Architecture of Compound AI Systems for Enterprise. arXiv:2406.00584.

Kaplan, J. et al. (2020). Scaling Laws for Neural Language Models. arXiv:2001.08361.

Liu, N. F. et al. (2024). Lost in the Middle: How Language Models Use Long Contexts. Transactions of the ACL, 12, 157–173.

Malekzadeh, P. & Plataniotis, K. (2024). Active Inference and Reinforcement Learning: A Unified Inference on Continuous State and Action Spaces. Neural Computation, 36(10), 2073–2135.

Marra, G. et al. (2024). From Statistical Relational to Neurosymbolic Artificial Intelligence: A Survey. Artificial Intelligence, 328, 104062.

Mehmood, U. et al. (2022). The Black-Box Simplex Architecture for Runtime Assurance of Autonomous CPS. NASA Formal Methods (NFM).

Millidge, B. (2024). A Retrospective on Active Inference. Blog post, July 2024.

Nasr, M., Carlini, N., Sitawarin, C., Schulhoff, S. V. et al. (2025). The Attacker Moves Second: Stronger Adaptive Attacks Bypass Defenses Against LLM Jailbreaks and Prompt Injections. arXiv:2510.09023. Joint OpenAI/Anthropic/Google DeepMind study.

Pan, S. et al. (2024). Unifying Large Language Models and Knowledge Graphs: A Roadmap. IEEE TKDE, 36(7), 3580–3599.

Paul, S. et al. (2024). On Efficient Computation in Active Inference. Expert Systems with Applications, 253, 124118.

Phan, D. et al. (2020). Neural Simplex Architecture. NASA Formal Methods (NFM), LNCS 12229.

Ross, S., Gordon, G. J., & Bagnell, J. A. (2011). A Reduction of Imitation Learning and Structured Prediction to No-Regret Online Learning. AISTATS.

Schick, T. et al. (2023). Toolformer: Language Models Can Teach Themselves to Use Tools. Proceedings of NeurIPS.

Sha, L. (2001). Using Simplicity to Control Complexity. IEEE Software, 18(4), 20–28.

Vaswani, A. et al. (2017). Attention Is All You Need. Proceedings of NeurIPS.

Wang, Y. et al. (2025). Agent Security Bench (ASB): Formalizing and Benchmarking Attacks and Defenses in LLM-based Agents. Proceedings of ICLR.

Wang, Y., Hovakimyan, N., & Sha, L. (2013). L1Simplex: Fault-Tolerant Control of Cyber-Physical Systems. Proceedings of ICCPS.

Wu, H. et al. (2025). Inference Scaling Laws: An Empirical Analysis of Compute-Optimal Inference. Proceedings of ICLR.

Wu, H. et al. (2024). Marabou 2.0: A Versatile Formal Analyzer of Neural Networks. Proceedings of CAV.

Yao, S. et al. (2023). ReAct: Synergizing Reasoning and Acting in Language Models. Proceedings of ICLR.

Zaharia, M. et al. (2024). The Shift from Models to Compound AI Systems. Berkeley Artificial Intelligence Research Blog.

Zhang, H. et al. (2018–2025). α,β-CROWN: Neural Network Verifier. VNN-COMP 2021–2025 winner.