CERT: Coordination Pipeline Observability for Multi-Agent LLM Systems

Javier Marín[1]

***Abstract*— Multi-agent LLM systems face a fundamental observability gap: while individual language models can be benchmarked in isolation, no systematic infrastructure exists for observing, debugging, and reasoning about coordination pipeline behavior in production deployments. We present CERT (Coordination Error and Risk Tracking), a mathematical framework that makes coordination pipeline behavior systematically observable through statistical characterization of interaction patterns, coordination effects, and compositional unpredictability.**

CERT extends established observability principles from distributed systems engineering to LLM coordination contexts, providing empirical measurement capabilities for systems based on discrete token manipulation that exhibit inherent compositional limitations. Experimental validation across single-agent, two-agent, and multi-agent configurations reveals systematic patterns: positive coordination effects ($\gamma = 1.394$) in controlled two-agent scenarios, and measurable compositional unpredictability (30% prediction error) in five-agent pipelines that exceeds what information-theoretic baselines can predict.

CERT's primary value lies in systematic measurement of coordination pipeline behavior that was previously invisible to engineering analysis, enabling quantitative assessment of which coordination patterns exhibit reliable performance characteristics versus those that should not be deployed in production environments. This represents necessary observability infrastructure for current discrete token manipulation systems while documenting the fundamental compositional limitations that prevent reliable scaling.

I. INTRODUCTION

Consider a concrete systems engineering scenario that illustrates why coordination pipeline observability represents a critical infrastructure gap. Our demand forecasting agent achieves 87% accuracy in isolation. Our inventory optimizer reduces waste by 23% when operating independently. Our negotiation agent secures favorable terms 78% of the time in standalone testing. We deploy them together in a sequential coordination pipeline, and the combined performance degrades catastrophically to 34% of expected effectiveness.

What systematic debugging tools do we have available for this coordination pipeline failure? In traditional distributed systems, we would have request tracing showing exactly which services were called and in what order, performance metrics revealing bottlenecks and timing issues, structured error logs capturing failures with full context, and system monitoring tracking resource utilization and service health over time [4], [6].

For LLM coordination pipelines, the honest answer is: essentially none that are systematic. We have demonstrations of when coordination works well and anecdotal evidence of when it fails, but no way to systematically observe, instrument, or reason about coordination pipeline behavior.

Traditional distributed systems have established formal foundations for coordination [8], [13], [23], enabling systematic reasoning about consensus, fault tolerance, and coordinated behavior even under adversarial conditions. These systems achieve reliability through mathematical guarantees about coordination protocols, precise timing semantics, and formal verification of distributed consensus mechanisms. In contrast, LLM coordination relies on natural language exchanges that lack formal semantics, making traditional coordination analysis inapplicable.

This represents what we characterize as a fundamental *observability gap* in LLM systems engineering. We're attempting to build reliable coordination pipelines using components we cannot systematically observe, debug, or optimize. It's equivalent to building distributed systems without request tracing, performance monitoring, or structured logging.

A. Current Approaches and Limitations

Existing multi-agent frameworks like AutoGen [55], OpenAI Swarm [36], and similar platforms provide sophisticated coordination mechanisms but no systematic observability infrastructure [38], [56]. They enable impressive coordination demonstrations but provide no systematic tools for understanding why coordination succeeds or fails, optimizing interaction patterns based on empirical evidence, or detecting performance unpredictability over time.

Traditional software observability approaches cannot be directly applied to LLM coordination because they violate fundamental assumptions that distributed systems observability depends upon [4]. Recent work on multiagent systems [49], [51] has primarily focused on reinforcement learning contexts where agents learn coordinated behaviors through environmental interaction, but these approaches assume deterministic action spaces that don't translate to natural language coordination contexts. Classical multi-agent evaluation methodologies [50], [52] provide frameworks for assessing coordination in structured environments with explicit protocols, but lack the statistical measurement capabilities needed for natural language coordination where behavioral contracts cannot be formally specified.

Interface Contract Violations: Traditional observability assumes well-defined component interfaces with measurable behavioral contracts. LLM agents coordinate through natural language interpretation, creating "interfaces" that lack formal specifications and exhibit semantic ambiguity that traditional monitoring cannot characterize [18], [57].

[1] mail: javier@jmarin.info

Non-Deterministic Behavior: Traditional observability tools assume identical inputs produce identical outputs, enabling systematic analysis through controlled testing. LLM systems exhibit non-deterministic responses that vary with context in ways that make traditional debugging approaches ineffective. This connects to broader challenges in machine learning model interpretability [28], [44] and the fundamental difficulties in understanding neural network decision processes [7], [35].

Decision Opacity: Traditional observability requires instrumentation of system state and decision logic. LLM coordination decisions emerge from opaque neural network computations that cannot be directly observed using standard monitoring approaches. This opacity problem extends beyond individual models to coordination patterns, where emergent behaviors arise from interactions between complex systems.

Dynamic Performance Characteristics: Traditional observability depends on stable behavioral baselines. LLM system performance varies significantly based on context, data distribution, and model updates in ways that make traditional performance monitoring inadequate [26], [40].

B. AI Safety and Reliability Context

The AI safety community has identified fundamental challenges in ensuring reliable behavior from AI systems [1], [45]. Research focuses on alignment problems, robustness to distribution shift, and providing theoretical guarantees about AI behavior [2], [9], [25]. Recent work on goal misgeneralization [33], [47] highlights how AI systems can pursue objectives that deviate from intended goals in ways that are difficult to detect and correct.

However, much of this research assumes architectural capabilities that current systems don't possess. Theoretical frameworks often require AI systems with genuine world understanding, stable goal representations, or predictable reasoning processes. While these represent important long-term research directions, they don't address the immediate challenge of measuring reliability in current pattern-matching systems.

CERT complements this research by providing empirical measurement tools that can validate theoretical predictions about current system limitations while testing whether new architectures exhibit the reliability properties that safety research identifies as necessary.

C. CERT Approach

CERT addresses this observability gap by providing mathematical foundations for systematic observation and analysis of LLM coordination pipeline behavior. Rather than attempting to eliminate coordination uncertainty—which appears mathematically impossible with current discrete token manipulation architectures—CERT makes coordination pipeline behavior systematically observable through statistical characterization.

The key insight underlying our approach draws from established principles in both distributed systems observability and AI interpretability research: while we cannot make LLM coordination deterministic, we can make it *systematically observable* through rigorous statistical measurement and analysis [4], [35].

CERT accomplishes this through mathematical frameworks that enable three critical engineering capabilities: systematic debugging through statistical characterization of coordination pipeline patterns, evidence-based optimization through quantitative measurement of coordination effectiveness, and proactive monitoring through real-time statistical analysis that detects coordination anomalies before they impact production systems.

II. MATHEMATICAL FOUNDATIONS

The systematic observability of LLM coordination pipelines requires mathematical frameworks that can characterize non-deterministic behavior patterns while providing the statistical rigor necessary for engineering decision-making. This section presents the theoretical foundations that enable systematic coordination pipeline observability despite the inherent uncertainty of discrete token manipulation systems.

A. Mathematical Framework Overview

The fundamental challenge in coordination pipeline observability is systematically characterizing the behavior of systems whose individual components are non-deterministic pattern-matching systems operating through discrete token manipulation.

The mathematical framework draws from statistical analysis of stochastic processes [43] and information-theoretic approaches to measuring semantic similarity [11], [48]. Consider the core measurement challenge: how do we quantify whether an LLM agent behaves "consistently" when it never produces identical responses to identical prompts? We address this through statistical variance analysis of semantic distances:

$$C(A_i, p) = 1 - \frac{\text{Var}[d(r_j, r_k)]}{\mathbb{E}[d(r_j, r_k)]}$$

where $d(\cdot, \cdot)$ measures semantic distance between responses using established embedding techniques. This extends the statistical concept of the coefficient of variation to semantic spaces, providing a dimensionless measure of behavioral consistency that remains meaningful despite response variability.

Similarly, coordination effect measurement requires comparing observed pipeline performance against statistical expectations from independent component behavior, following established experimental design principles [31]:

$$\gamma^P_{i,j}(t) = \frac{\mathbb{E}[\mathbf{P}^{\text{coordinated}}_{ij}]}{\mathbb{E}[\mathbf{P}^{\text{independent}}_{i}] \times \mathbb{E}[\mathbf{P}^{\text{independent}}_{j}]}$$

When $\gamma > 1$, coordination provides synergistic benefits; when $\gamma < 1$, coordination introduces systematic unpredictability requiring engineering attention. This framework

enables systematic analysis of coordination effects using established statistical methodology while handling the semantic complexity inherent in LLM systems, drawing from broader work in stochastic systems analysis [5].

B. Problem Formulation

We model an LLM coordination system as a collection of agents $\mathcal{A} = \{A_1, A_2, \ldots, A_n\}$ coordinating through structured pipelines to solve tasks $\mathcal{T}$, where coordination pipeline behavior must be made systematically observable for engineering purposes.

Following established principles from distributed systems observability [4], CERT provides mathematical frameworks for three fundamental observability capabilities: behavioral instrumentation through statistical characterization of coordination pipeline patterns that enables debugging through systematic analysis, performance analytics through quantitative analysis of coordination effectiveness that enables optimization based on empirical evidence, and anomaly detection through real-time monitoring that enables proactive identification of coordination unpredictability.

This formulation extends proven observability principles from traditional systems to LLM coordination contexts while maintaining the systematic engineering capabilities that reliable systems require, incorporating insights from complex systems theory and network analysis approaches.

C. Individual Agent Behavioral Characterization

To debug coordination pipeline failures systematically, we need quantitative measures of how consistently individual LLM agents behave. In traditional systems, we'd use exact output matching, but LLM agents express similar ideas in different words, requiring semantic approaches to consistency measurement.

Definition 1 (Behavioral Consistency): For agent A_i, prompt p, and n independent trials, behavioral consistency is:

$$C(A_i, p) = 1 - \frac{\sigma(\{d(r_j, r_k) : 1 \leq j < k \leq n\})}{\mu(\{d(r_j, r_k) : 1 \leq j < k \leq n\})}$$

where $r_j = A_i(p)$ on trial j, $d(\cdot, \cdot)$ measures semantic distance using established embedding techniques [12], [41], and σ, μ denote the standard deviation and mean.

High consistency ($C \to 1$) indicates predictable agent behavior that facilitates coordination pipeline observability. Low consistency ($C \to 0$) indicates high variability that complicates coordination analysis and may require specialized observability approaches.

Understanding individual agent performance distributions is essential for distinguishing coordination effects from inherent agent variability. This builds on established approaches to text quality assessment [14], [22] while extending them to coordination contexts.

Definition 2 (Performance Distribution): For agent A_i and task category $\mathcal{C}$, performance is characterized through success indicators $S_{i,\mathcal{C}} = \{s_1, s_2, \ldots, s_m\}$ across m trials:

$$\mu_{i,\mathcal{C}} = \frac{1}{m} \sum_{j=1}^{m} s_j \tag{1}$$

$$\sigma^2_{i,\mathcal{C}} = \frac{1}{m-1} \sum_{j=1}^{m} (s_j - \mu_{i,\mathcal{C}})^2 \tag{2}$$

This statistical characterization provides the behavioral baseline needed for coordination pipeline observability, incorporating principles from automatic evaluation methodologies [27], [37] while adapting them for coordination analysis rather than translation or summarization tasks.

D. Coordination Pipeline Effect Analysis

The critical challenge for coordination pipeline observability is making agent interactions systematically measurable. Traditional systems trace API calls and measure response times, but LLM coordination happens through natural language exchanges that can't be traced conventionally. Instead, we measure coordination effects by comparing observed outcomes with expected independent performance.

Definition 3 (Coordination Effect): For agents A_i, A_j coordinating on task t through pattern P:

$$\gamma^P_{i,j}(t) = \frac{\text{Observed Coordination Performance}}{\text{Expected Independent Performance}}$$

where coordination effects become systematically observable through statistical comparison.

When $\gamma > 1$, coordination provides performance benefits that can be systematically analyzed and optimized. When $\gamma < 1$, coordination introduces performance unpredictability that requires systematic debugging.

Complex coordination involves chains of agent interactions where observability can decay as interaction complexity increases. This connects to broader questions about compositional behavior in complex systems.

Definition 4 (Coordination Chain Analysis): For coordination chains $A_{i_1} \xrightarrow{P_1} A_{i_2} \to \cdots \to A_{i_{k-1}} \xrightarrow{P_{k-1}} A_{i_k}$:

$$O_{\text{chain}} = O_{\text{base}} \times \prod_{j=1}^{k-1} \gamma^{P_j}_{i_j, i_{j+1}}$$

where O_{base} represents baseline observability and coordination effects compound across interaction chains.

This multiplicative model enables systematic analysis of how coordination complexity affects system observability and engineering capabilities.

E. Compositional Unpredictability Analysis

Multi-agent LLM systems exhibit compositional unpredictability where system behavior cannot be predicted from individual agent analysis. However, measuring this unpredictability requires principled baseline models that account for the information-theoretic constraints of sequential processing.

Definition 5 (Information-Theoretic Baseline): For a sequential coordination pipeline $\mathcal{A} = \{A_1, \ldots, A_n\}$, the predicted performance incorporates measured coordination effects and information degradation:

$$P_{\text{baseline}}(\mathcal{A}, t) = \prod_{i=1}^{n} \mathbb{E}[P_i^{\text{independent}}] \times \prod_{j=1}^{n-1} \gamma_{j,j+1} \times \phi(n)$$

where $\gamma_{j,j+1}$ represents measured pairwise coordination effects and $\phi(n) = \alpha^{n-1}$ models information degradation through processing chains with $\alpha \in [0.9, 0.95]$ based on empirical studies of sequential processing.

Definition 6 (Compositional Prediction Error): For system $\mathcal{A} = \{A_1, \ldots, A_n\}$ executing task t:

$$\epsilon_{\text{comp}}(t) = \frac{|\mathbb{E}[S(\mathcal{A}, t)] - P_{\text{baseline}}(\mathcal{A}, t)|}{P_{\text{baseline}}(\mathcal{A}, t)}$$

where P_{baseline} incorporates individual performance, measured coordination effects, and information degradation through processing chains.

Large prediction errors indicate compositional unpredictability that requires additional observability infrastructure and potentially different engineering approaches. This framework makes previously unobservable compositional limitations systematically measurable and thereby debuggable.

Definition 7 (Compositional Unpredictability):

$$\Omega(n, t) = \frac{\text{Var}[S(\mathcal{A}, t)]}{\text{Var}[P_{\text{baseline}}(\{A_i\}, \{\gamma_{i,j}\}, t)]}$$

When $\Omega > 1$, compositional unpredictability increases system complexity beyond what can be predicted from component analysis.

F. Production Observability Metrics

The mathematical frameworks translate into practical observability metrics that enable systematic engineering of coordination pipelines in production environments, drawing from established practices in distributed systems monitoring [4] and machine learning system deployment [26].

Definition 8 (Coordination Pipeline Health Score):

$$H_{\text{coord}}(t) = \frac{1}{1 + \epsilon_{\text{comp}}(t)} \times \prod_{i,j} \min(1, \gamma_{i,j}(t)) \times C_{\text{obs}}(t)$$

where $\epsilon_{\text{comp}}(t)$ represents compositional prediction error, $\gamma_{i,j}(t)$ captures pairwise coordination effects, and $C_{\text{obs}}(t)$ measures observability coverage.

Definition 9 (Coordination Reliability Metric):

$$R_{\text{coord}}(t) = \exp(-\epsilon_{\text{comp}}(t)) \times \frac{\text{Successful Coordination Events}}{\text{Total Coordination Attempts}}$$

This provides a quantitative reliability assessment that accounts for both prediction accuracy and operational success rates.

Definition 10 (Observability Coverage):

$$C_{\text{obs}} = \frac{\text{Number of Instrumented Interactions}}{\text{Total Number of Interactions}}$$

This measures how comprehensively the coordination pipeline is instrumented, analogous to code coverage metrics in traditional software engineering.

These metrics provide systematic observability infrastructure that enables reliable engineering of LLM coordination pipelines using established engineering practices, building on successful patterns from traditional systems while adapting them for the unique challenges of coordination pipeline monitoring.

III. IMPLEMENTATION ARCHITECTURE

While CERT's primary contribution is the mathematical framework for coordination pipeline observability, practical deployment requires systematic infrastructure architecture that extends proven observability principles to LLM coordination contexts.

A. Three-Layer Architecture

CERT's implementation architecture follows established patterns from distributed systems observability [4], [6], incorporating lessons from visualization and debugging tools for complex systems [16], [54]. Modern production observability frameworks [21], [34] emphasize proactive monitoring, circuit breaker patterns [15], and systematic failure detection, but these approaches assume deterministic component behavior that doesn't apply to LLM coordination contexts.

Instrumentation Layer: Systematic data collection from coordination interactions using statistical sampling protocols adapted for non-deterministic LLM behavior. This layer implements the behavioral consistency measurement and performance distribution characterization described in Section II.

Analytics Layer: Processing of observability data to compute coordination effects, detect compositional unpredictability, and identify optimization opportunities. This layer implements the mathematical frameworks for coordination effect quantification and unpredictability analysis.

Interface Layer: Integration with existing engineering workflows through APIs, dashboards, and alerting systems that provide coordination observability insights using familiar monitoring patterns.

B. Integration Patterns

CERT observability integrates with existing multi-agent frameworks through standardized instrumentation interfaces that minimize implementation complexity while providing comprehensive coordination visibility. This builds on established patterns from distributed systems architecture while addressing the unique challenges of LLM coordination contexts.

Framework Wrappers: Non-invasive instrumentation that wraps existing agent implementations to collect behavioral telemetry without requiring changes to agent logic or coordination patterns.

Middleware Integration: Observability infrastructure that operates at the coordination layer, instrumenting message passing and interaction patterns independent of specific agent implementations.

Analytics Integration: Coordination observability data integrated with existing monitoring, logging, and alerting infrastructure using standard formats and protocols.

IV. EXPERIMENTAL VALIDATION

A. Experimental Design and Methodology

Our experimental validation implements the mathematical framework through systematic behavioral measurement across three coordination scenarios: single-agent baseline characterization, two-agent coordination effect analysis, and multi-agent compositional unpredictability analysis.

1) Infrastructure Configuration: Experiments were conducted using Anthropic's Claude model family via API, providing controlled access to production-quality language models with consistent behavioral characteristics. We selected Claude-3-Haiku and Claude-3.5-Haiku as test agents for both the 2-agents and the 5-agents experiments, providing sufficient architectural diversity for coordination analysis while maintaining consistent evaluation protocols.

2) Evaluation Methodology: Our multidimensional response evaluation framework addresses the limitations of superficial metrics through composite quality assessment, building on established approaches to text quality measurement [14], [19], [22], [30] while adapting them for coordination contexts.

Semantic Relevance (30%): Measured using cosine similarity between prompt and response embeddings, following established practices in semantic similarity assessment [12], [41].

Linguistic Coherence (30%): Evaluated through readability metrics with optimal ranges: 10-30 words per sentence, 4-8 characters per word, incorporating principles from computational linguistics [42].

Content Density (40%): Assessed via analytical term frequency patterns, drawing from information-theoretic approaches [48] and automatic content assessment methodologies [20], [29], [46].

The composite quality score follows:

$$Q(p, r) = 0.3 \times \text{semantic\_score} + 0.3 \times \text{coherence\_score} + 0.4 \times \text{content\_score} \quad (3)$$

3) Statistical Validation Protocol: All experimental measurements follow rigorous statistical protocols: 20 independent trials per agent-task combination for behavioral consistency, 15 trials across diverse analytical prompts for baseline establishment, 15 independent coordination scenarios for coordination measurement, and Welch's t-test with $\alpha = 0.05$ for statistical significance assessment.

Coordination effects are computed as:

$$\gamma = \frac{\mathbb{E}[\text{Coordination Performance}]}{\prod_{i=1}^{n} \mathbb{E}[\text{P}_i^{\text{independent}}]}$$

B. Experimental Results

*1) **Experiment 1 - Single Agent Behavioral Characterization**:* We establish baseline behavioral consistency and performance characteristics for individual LLM agents.

TABLE I
SINGLE AGENT BASELINE MEASUREMENTS

Metric	Claude-3-Haiku
Behavioral Consistency (C)	0.828
Performance Mean (μ)	0.612
Performance Std (σ)	0.077
Trials	20/15

The consistency score $C = 0.828$ indicates substantial behavioral predictability within the constraints of discrete token manipulation systems. The performance baseline provides a stable foundation for coordination effect measurement.

*2) **Experiment 2 - Two-Agent Coordination Analysis**:* We measure coordination effects between two different LLM agents in a sequential pipeline configuration.

TABLE II
TWO-AGENT COORDINATION RESULTS

Agent	C	$\mu \pm \sigma$
Agent 1	0.844	0.616 ± 0.039
Agent 2	0.885	0.771 ± 0.053
Coordination Metrics		
Expected	–	0.475
Observed	–	0.662 ± 0.080
Effect (γ)	–	1.394
Significance	–	$p < 0.001$

The coordination effect $\gamma = 1.394$ represents a 39.4% performance improvement over expected independent performance, proving that structured information flow between different model architectures can provide measurable benefits under controlled conditions.

*3) **Experiment 3 - Multi-Agent Compositional Unpredictability Analysis**:* Using the information-theoretic baseline model, we establish predicted performance by incorporating:

- Individual agent baselines: $\mu_1 = 0.616$, $\mu_2 = 0.771$ (from Experiment 2)
- Measured coordination effect: $\gamma = 1.394$ (from two-agent analysis)
- Information degradation factor: $\phi(5) = 0.93^4 = 0.748$ for five-agent chains

$$P_{\text{baseline}} = (0.616 \times 0.771) \times 1.394^2 \times 0.748 = 0.524$$

TABLE III
MULTI-AGENT COMPOSITIONAL UNPREDICTABILITY RESULTS

Metric	Five-Agent Pipeline
Predicted Performance	0.524
Observed Performance	0.682 ± 0.043
Prediction Error (ϵ)	0.302
Effect Size (Cohen's d)	3.67
Trials	15

The 30.2% prediction error indicates moderate compositional unpredictability that exceeds what can be explained by measurement uncertainty alone (effect size = 3.67).

C. Cross-Experimental Analysis

1) Coordination Scaling Patterns: Our results reveal systematic patterns in coordination pipeline behavior:

1) **Single Agent**: Stable baseline with predictable behavioral consistency
2) **Two-Agent**: Positive coordination effects with reliable benefits
3) **Multi-Agent**: Measurable compositional unpredictability with fundamental limitations

TABLE IV
OBSERVABILITY COVERAGE ANALYSIS

Configuration	C_{obs}	Status
Single Agent	1.0	Complete
Two-Agent	0.95	Systematic
Multi-Agent	0.60	Limited

2) Observability Coverage Analysis: This coverage decay confirms that coordination pipeline observability faces fundamental limitations as interaction complexity increases.

TABLE V
COORDINATION PIPELINE SCALING SUMMARY

Metric	Value	Interpretation
γ (Two-Agent)	1.394	Measurable Benefits
ϵ (Multi-Agent)	0.302	Compositional Limits
C_{obs} Coverage	$1.0 \rightarrow 0.60$	Observability Decay

V. CONCLUSIONS

A. Infrastructure Contributions and Engineering Value

Our experimental validation demonstrates that systematic observability of LLM coordination pipelines is both achievable and practically valuable for engineering reliable systems based on discrete token manipulation architectures. CERT successfully extends established observability principles from distributed systems engineering to LLM coordination contexts, providing systematic measurement capabilities for coordination phenomena that were previously invisible to engineering analysis.

This represents necessary observability infrastructure for current LLM deployment rather than a breakthrough in intelligence architectures. The mathematical framework enables systematic debugging, evidence-based optimization, and proactive monitoring of coordination pipelines—capabilities essential for reliable engineering of production systems based on discrete token manipulation.

B. Empirical Evidence and Architectural Implications

Our two-agent coordination experiment demonstrates that structured information flow between different model architectures can yield statistically significant performance improvements ($\gamma = 1.394$, $p < 0.001$). However, the multi-agent results reveal fundamental compositional limitations. The prediction error $\epsilon = 0.302$ indicates moderate compositional unpredictability that exceeds measurement uncertainty, providing quantitative evidence of the limitations inherent in discrete token manipulation systems.

This connects to broader principles about intelligence architectures. Genuine intelligence requires learned representations of continuous physical dynamics through sensorimotor interaction [10], [24], [39], not discrete symbol manipulation. Our work provides quantitative evidence of this fundamental constraint, which is essential preparation for developing architectures based on world models learned from real physical interaction.

C. Engineering Implications

For organizations deploying LLM coordination systems, our results provide quantitative guidance: two-agent pipelines can provide measurable benefits when properly structured and monitored, multi-agent pipelines exhibit fundamental compositional unpredictability requiring extensive observability infrastructure and conservative deployment strategies, and scaling limitations indicate that complex coordination patterns may not be suitable for production deployment without significant risk management.

Our findings suggest a dual-track approach: continue developing observability and reliability tools for current discrete token systems, focusing on coordination patterns that exhibit predictable benefits, while pursuing research into architectures based on learned representations that could provide intrinsic compositional properties necessary for reliable coordination.

D. Final Assessment

CERT represents valuable infrastructure engineering that enables systematic measurement and analysis of coordination pipeline behavior in current discrete token manipulation systems. While this work does not address the fundamental architectural limitations it measures, it provides necessary observability scaffolding for understanding current system constraints and making informed deployment decisions.

The experimental validation confirms both the utility of systematic coordination pipeline observability and the fundamental compositional limitations of current approaches. For immediate practical applications, CERT enables responsible deployment with quantitative risk assessment. For research strategy, it provides empirical guidance about coordination patterns that work reliably versus those requiring architectural innovations beyond current paradigms.

This infrastructure work creates necessary capabilities for current deployment needs while providing quantitative evidence that motivates research toward architectures based on learned world models. The 30% prediction error we document provides systematic measurement of compositional limitations in discrete token manipulation systems—essential data for understanding current constraints while developing architectures that could exhibit genuine compositional understanding through learned representations of continuous physical dynamics.

REFERENCES

[1] D. Amodei, C. Olah, J. Steinhardt, P. Christiano, J. Schulman, and D. Mané, "Concrete Problems in AI Safety," *arXiv preprint arXiv:1606.06565*, 2016.

[2] Y. Bai, S. Kadavath, S. Kundu, A. Askell, J. Kernion, A. Jones, A. Chen, A. Goldie, A. Mirhoseini, C. McKinnon, *et al.*, "Constitutional AI: Harmlessness from AI feedback," *arXiv preprint arXiv:2212.08073*, 2022.

[3] Y. Bar-Yam, *Making things work: solving complex problems in a complex world*, NECSI Knowledge Press, 2004.

[4] B. Beyer, C. Jones, J. Petoff, and N. R. Murphy, *Site Reliability Engineering: How Google Runs Production Systems*, O'Reilly Media, 2016.

[5] P. Billingsley, *Probability and Measure*, 4th ed., John Wiley & Sons, 2017.

[6] C. Majors, L. Fong-Jones, and G. Miranda, *Distributed Systems Observability: A Guide to Building Robust Systems*, O'Reilly Media, 2018.

[7] N. Cammarata, S. Carter, G. Goh, C. Olah, M. Petrov, L. Schubert, C. Voss, B. Egan, and S. K. Lim, "Thread: Circuits," *Distill*, vol. 5, no. 3, p. e24, 2020.

[8] M. Castro and B. Liskov, "Practical Byzantine fault tolerance," *Proceedings of the third symposium on Operating systems design and implementation*, pp. 173-186, 1999.

[9] P. F. Christiano, J. Leike, T. Brown, M. Martic, S. Legg, and D. Amodei, "Deep reinforcement learning from human preferences," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[10] A. Clark, *Being there: Putting brain, body, and world together again*, MIT Press, 1999.

[11] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, 2nd ed., John Wiley & Sons, 2012.

[12] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[13] M. J. Fischer, N. A. Lynch, and M. S. Paterson, "Impossibility of distributed consensus with one faulty process," *Journal of the ACM*, vol. 32, no. 2, pp. 374-382, 1985.

[14] R. Flesch, "A new readability yardstick," *Journal of Applied Psychology*, vol. 32, no. 3, pp. 221-233, 1948.

[15] M. Fowler, "CircuitBreaker," 2013. [Online]. Available: https://martinfowler.com/bliki/CircuitBreaker.html

[16] F. Hohman, A. Head, R. Caruana, R. DeLine, and S. M. Drucker, "Gamut: A design probe to understand how data scientists understand machine learning models," *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pp. 1-13, 2019.

[17] J. H. Holland, *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*, MIT Press, 1992.

[18] A. Holtzman, J. Buys, L. Du, M. Forbes, and Y. Choi, "The curious case of neural text degeneration," *Proceedings of the International Conference on Learning Representations*, 2020.

[19] S. Jarvis, "Capturing the diversity in lexical diversity," *Language learning*, vol. 63, pp. 87-106, 2013.

[20] T. Joachims, "Text categorization with support vector machines: Learning with many relevant features," in *European conference on machine learning*, Springer, 1998, pp. 137-142.

[21] A. Kamps and J. Brendel, *Observability Engineering: Achieving Production Excellence*, O'Reilly Media, 2022.

[22] J. P. Kincaid, R. P. Fishburne Jr, R. L. Rogers, and B. S. Chissom, "Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel," Naval Technical Training Command Millington TN Research Branch, Tech. Rep., 1975.

[23] L. Lamport, "Time, clocks, and the ordering of events in a distributed system," *Communications of the ACM*, vol. 21, no. 7, pp. 558-565, 1978.

[24] Y. LeCun, "A path towards autonomous machine intelligence," *OpenReview*, Version 0.9.2, 2022.

[25] J. Leike, M. Martic, V. Krakovna, P. A. Ortega, T. Everitt, A. Lefrancq, L. Orseau, and S. Legg, "AI Safety Gridworlds," *arXiv preprint arXiv:1711.09883*, 2017.

[26] P. Liang, R. Bommasani, T. Lee, D. Tsipras, D. Soylu, M. Yasunaga, Y. Zhang, D. Narayanan, Y. Wu, A. Kumar, *et al.*, "Holistic evaluation of language models," *arXiv preprint arXiv:2211.09110*, 2022.

[27] C.-Y. Lin, "Rouge: A package for automatic evaluation of summaries," in *Text summarization branches out*, 2004, pp. 74-81.

[28] Z. C. Lipton, "The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery," *Queue*, vol. 16, no. 3, pp. 31-57, 2018.

[29] A. Louis and A. Nenkova, "Automatically assessing machine summary content without a gold standard," *Computational Linguistics*, vol. 36, no. 3, pp. 421-447, 2010.

[30] P. M. McCarthy and S. Jarvis, "MTLD, vocd-D, and HD-D: A validation study of sophisticated approaches to lexical diversity assessment," *Behavior research methods*, vol. 42, no. 2, pp. 381-392, 2010.

[31] D. C. Montgomery, *Design and Analysis of Experiments*, 9th ed., John Wiley & Sons, 2017.

[32] D. C. Montgomery, *Introduction to Statistical Quality Control*, 8th ed., John Wiley & Sons, 2019.

[33] R. Ngo, L. Chan, and S. Mindermann, "The Alignment Problem from a Deep Learning Perspective," *arXiv preprint arXiv:2209.00626*, 2022.

[34] M. T. Nygard, *Release It!: Design and Deploy Production-Ready Software*, 2nd ed., Pragmatic Bookshelf, 2018.

[35] C. Olah, N. Cammarata, L. Schubert, G. Goh, M. Petrov, and S. Carter, "Zoom in: An introduction to circuits," *Distill*, vol. 5, no. 3, p. e00024.001, 2020.

[36] OpenAI, "Swarm: Educational framework for multi-agent orchestration," 2024. [Online]. Available: https://github.com/openai/swarm

[37] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, 2002, pp. 311-318.

[38] J. S. Park, J. C. O'Brien, C. J. Cai, M. R. Morris, P. Liang, and M. S. Bernstein, "Generative agents: Interactive simulacra of human behavior," *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, pp. 1-22, 2023.

[39] R. Pfeifer and J. Bongard, *How the body shapes the way we think: a new view of intelligence*, MIT Press, 2007.

[40] J. Quiñonero-Candela, *Dataset shift in machine learning*, Mit Press, 2009.

[41] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using Siamese BERT-networks," *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, 2019.

[42] A. Rogers, O. Kovaleva, and A. Rumshisky, "A Primer in BERTology: What We Know About How BERT Works," *Transactions of the Association for Computational Linguistics*, vol. 8, pp. 842-866, 2020.

[43] S. M. Ross, *Introduction to Probability Models*, 11th ed., Academic Press, 2014.

[44] C. Rudin, "Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead," *Nature Machine Intelligence*, vol. 1, no. 5, pp. 206-215, 2019.

[45] S. Russell, *Human Compatible: Artificial Intelligence and the Problem of Control*, Viking, 2019.

[46] F. Sebastiani, "Machine learning in automated text categorization," *ACM computing surveys*, vol. 34, no. 1, pp. 1-47, 2002.

[47] R. Shah, V. Varma, R. Kumar, M. Phuong, V. Krakovna, J. Uesato, and Z. Kenton, "Goal Misgeneralization: Why Correct Specifications Aren't Enough For Correct Goals," *arXiv preprint arXiv:2210.01790*, 2022.

[48] C. E. Shannon, "A mathematical theory of communication," *The Bell system technical journal*, vol. 27, no. 3, pp. 379-423, 1948.

[49] P. Stone and M. Veloso, "Multiagent systems: a survey from a machine learning perspective," *Autonomous Robots*, vol. 8, no. 3, pp. 345-383, 2000.

[50] P. Stone, G. A. Kaminka, S. Kraus, and J. S. Rosenschein, "Ad hoc autonomous agent teams: Collaboration without pre-coordination," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 24, pp. 1504-1509, 2010.

[51] A. Tampuu, T. Matiisen, D. Kodelja, I. Kuzovkin, K. Korjus, J. Aru, J. Aru, and R. Vicente, "Multiagent deep reinforcement learning with extremely sparse rewards," *arXiv preprint arXiv:1707.01495*, 2017.

[52] G. Weiss, ed., *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*, MIT Press, 1999.

[53] D. J. Wheeler and D. S. Chambers, *Understanding Statistical Process Control*, 3rd ed., SPC Press, 2010.

[54] K. Wongsuphasawat, D. Smilkov, J. Wexler, J. Wilson, D. Mane, D. Fritz, D. Krishnan, F. B. Viegas, and M. Wattenberg, "Visualizing dataflow graphs of deep learning models in TensorFlow," *IEEE Transactions on Visualization and Computer Graphics*, vol. 24, no. 1, pp. 1-12, 2017.

[55] Q. Wu, G. Bansal, J. Zhang, Y. Wu, B. Li, E. Zhu, L. Jiang, X. Zhang, S. Zhang, J. Liu, *et al.*, "AutoGen: Enabling next-gen LLM applications via multi-agent conversation," *arXiv preprint arXiv:2308.08155*, 2023.

[56] Z. Xi, W. Chen, X. Guo, W. He, Y. Ding, B. Hong, M. Zhang, J. Wang, S. Jin, E. Zhou, *et al.*, "The rise and potential of large language model based agents: A survey," *arXiv preprint arXiv:2309.07864*, 2023.

[57] T. Z. Zhao, E. Wallace, S. Feng, D. Klein, and S. Singh, "Calibrate before use: Improving few-shot performance of language models," *Proceedings of the International Conference on Machine Learning*, pp. 12697-12706, 2021.