The City College of New York

CSC 36000:
Modern Distributed Computing *with AI Agents*

By Saptarashmi Bandyopadhyay
Email: sbandyopadhyay@ccny.cuny.edu
Assistant Professor of Computer Science
City College of New York and Graduate Center at City University of New York

December 8, 2025 CSC 36000

Today's Lecture

The "Black Box" Problemin Distributed Systems

The Complexity Multiplier

- Local vs. Global Interpretability
- Emergent Behaviors

Techniques for Explanation

- Distributed Shapley Values
- Attention Maps in Multi-Agent Systems
- Communication Analysis

Causal Inference in Distributed AI

The "Black Box" Problem

Why do we need Interpretability?

Recall: We discussed Deep Q-Networks (DQN) and Actor-Critic models (Nov 3).

The Issue: Neural Networks are universal function approximators, but they are opaque.

We know the Input (State) and the Output (Action), but the millions of parameters in the middle are uninterpretable to humans.

High Stakes: In safety-critical distributed systems (e.g., Autonomous Grids, Self-Driving Fleets), "It just works" isn't good enough. We need to know why.

The Complexity Multiplier

The Complexity of Distributed AI Interactions

In Single-Agent AI, we ask: "Why did the agent do that?"

In Distributed AI, we must ask:

- "Why did Agent A do that?"
- "How did Agent B's message influence Agent A?"
- "Did the network latency affect the decision?"

Emergent Behavior: Often, the system-level outcome (e.g., a traffic jam clearing up) cannot be explained by looking at a single agent's code. It arises from the interaction.

The "Rashomon Effect" in Distributed Systems

Partial Observability: In distributed systems (like the MapReduce examples or Ring All-Reduce), no single node sees the whole picture.

Conflicting Explanations:

- Agent A might think the system failed because of a packet loss.
- Agent B might think it failed because Agent C acted irrationally.

Goal: Interpretable Distributed AI aims to create a unified, consistent narrative from these fragmented viewpoints.

Techniques for Explanation

Distributed Shapley Values

Recall: Game Theory and Nash Equilibrium (Nov 17).

Shapley Values: A concept from cooperative game theory used to fairly distribute "payout" among players.

In AI Interpretability: We use this to calculate the marginal contribution of a specific feature (or specific agent) to the total prediction or reward.

The Challenge: Calculating exact Shapley values is NP-Hard (exponential complexity). In a distributed system with N agents, this is computationally expensive.

Solution: Monte-Carlo approximations running in parallel (using JAX!).

Attention as Explanation

Transformers: The architecture behind LLMs uses "Attention" mechanisms.

Multi-Agent Attention:

- We can design our agents to use attention heads to look at other agents.
- The "Heatmap" of Influence: By visualizing the attention weights, we can see exactly who Agent A was ignoring and who it was listening to at timestep T.

This turns the "Black Box" into a "Glass Box" regarding communication topology.

Analyzing Communication

Recall: Agent Ontologies and FIPA-ACL (Nov 5).

Explicit vs. Implicit:

- Explicit (Symbolic): Agents send JSON/Text. We can read the logs. High Interpretability.
- Implicit (Vector): Agents send raw embeddings (vectors of floats). Lower Interpretability.

Translation: We can train secondary "Observer Agents" to translate vector messages back into human-readable text to debug distributed coordination failures.

Causal Inference

Correlation vs Causation

Correlation != Causation: Just because Agent A sent a message before Agent B crashed, doesn't mean A killed B.

Counterfactual Reasoning: To truly interpret a distributed event, we simulate "What if?" scenarios.

- "What if the network link between NY and London hadn't failed?"
- "What if Agent C had chosen action **Y** instead of **X**?"

We use Structural Causal Models (SCMs) to map the flow of influence through the distributed graph.

Real-World Example: Supply Chain Debugging

Scenario: A global shipment of GPUs is delayed.

The Blame Game:

- Truck Agent: "I was on time."
- Warehouse Agent: "I had no space."
- Planning Agent: "The weather data was wrong."

Interpretable AI Solution:

- Use Shapley Values to determine that the Planning Agent's reliance on stale weather data contributed 80% to the failure.
- Use Attention Maps to see that the Warehouse Agent successfully signaled "Full," but the Truck Agent's attention was focused on "Traffic."

Summary

Distributed AI introduces Interaction Complexity on top of Model Complexity.

We need Interpretability for trust, safety, and debugging.

Tools:

- Game Theory (Shapley Values)
- Network Analysis (Attention Maps)
- Causal Inference (Counterfactuals)

The Future: Self-Explaining Distributed Systems (Systems that generate their own "Post-Mortem" reports).

Questions?