
Ripple Effect Protocol: Coordinating Agent Populations

Ayush Chopra^{1,2} Aman Sharma² Feroz Ahmad² Luca Muscariello³ Vijoy Pandey³
Ramesh Raskar^{1,2}

¹Massachusetts Institute of Technology ²Project Iceberg ³Cisco

Abstract. Modern AI agents can exchange messages using protocols such as A2A and ACP, yet these mechanisms emphasize communication over coordination. As agent populations grow, this limitation produces brittle collective behavior, where individually “smart” agents converge on poor group outcomes. We introduce the Ripple Effect Protocol (REP), a coordination protocol in which agents share not only their decisions but also lightweight sensitivities—signals expressing how their choices would change if key environmental variables shifted. These sensitivities ripple through local networks, enabling groups to align faster and more stably than with agent-centric communication alone. We formalize REP’s protocol specification, separating required message schemas from optional aggregation rules, and evaluate it across scenarios with varying incentives and network topologies. Benchmarks across three domains—supply chain cascades (Beer Game), preference aggregation in sparse networks (Movie Scheduling), and sustainable resource allocation (Fishbanks)—show that REP improves coordination accuracy and efficiency over A2A by 41–100%, while flexibly handling multi-modal sensitivity signals from LLMs. By making coordination a protocol-level capability, REP provides scalable infrastructure for the emerging Internet of Agents.

Date: October 16, 2025

Correspondence: Ayush Chopra (ayushc@mit.edu)

SDK: <https://github.com/AgentTorch/rep>



1 Introduction

The proliferation of LLM-based agents across web services, enterprise environments, and IoT devices is creating a new coordination challenge: how can intelligent agents that express reasoning in natural language coordinate effectively without centralized control? Unlike traditional multi-agent systems designed for controlled simulations, this emerging “Agentic Web” requires coordination protocols that work with heterogeneous agents developed independently by different organizations, each with distinct objectives and reasoning capabilities [32, 35].

Existing coordination approaches assume either centralized design (multi-agent reinforcement learning [25, 31]), shared global objectives (consensus algorithms) [7, 22], or orchestrated control (LLM coordination frameworks like MetaGPT [16] and AgentVerse [8]). These paradigms work within controlled environments but cannot scale to independently owned agents operating across trust boundaries. Modern agent communication protocols like A2A [14], ACP [5], Agora [19] SLIM [2] provide message exchange and discovery mechanisms, yet they offer no structured approach for sharing the reasoning flexibility that underlies agent decisions.

The fundamental challenge is that LLM agents naturally express decision sensitivity through textual reasoning—assessments like “demand spike seems temporary” or “upstream capacity more

constraining than downstream orders"—but no coordination protocol exists for systematically sharing and aggregating such qualitative sensitivities. When agents share only final decisions without the reasoning flexibility behind them, coordination becomes brittle: locally rational choices lead to collective failures as agents lack visibility into others’ reasoning processes and constraints [11, 17, 30].

We introduce the Ripple Effect Protocol (REP), a coordination protocol specifically designed for LLM-based agents where agents share not only their decisions but also lightweight textual sensitivities—natural language signals that express how their choices would change if key environment variables shifted. These sensitivities ripple through local networks via existing transport protocols like SLIM, enabling groups to align faster and more stably than with decision-only communication.

REP’s design separates agent cognition from protocol coordination. Agents remain responsible for local reasoning and policy evaluation using their native LLM capabilities, while the protocol handles sensitivity aggregation and consensus mechanisms. This separation ensures that REP can coordinate heterogeneous agents—from different LLM architectures to hybrid rule-based systems—without constraining their internal decision-making processes.

The key insight underlying REP is that scalable coordination for LLM agents requires protocol-mediated sharing of decision flexibility rather than structured interaction frameworks. REP provides this coordination intelligence as a protocol primitive, enabling diverse agents to align their actions at scale without constraining their natural language reasoning capabilities.

We demonstrate REP across three coordination domains that represent fundamental challenges in distributed agent systems: supply chain information cascades (Beer Game) [29], preference aggregation in sparse social networks (movie coordination), and sustainable resource allocation under competing incentives (Fishbanks) [28]. Across these experiments, REP demonstrates substantial improvements in coordination accuracy compared to traditional agent-to-agent communication, while showing how textual sensitivity sharing accommodates the natural language reasoning of LLM-based agents.

2 Background and Related Work

2.1 Protocol Stack for LLM Agents

Recent work has produced a set of communication protocols that enable LLM agents to interoperate across platforms. The Model Context Protocol (MCP) standardizes how agents expose tools and share resources. Cross-agent messaging frameworks such as Google’s Agent-to-Agent (A2A) [1], IBM’s ACP, the Agent Network Protocol (ANP) [13], and research systems like Agora [19] support discovery, authenticated capability exchange, and task hand-offs. For performance-critical settings, SLIM (Secure Low-Latency Interactive Messaging) provides a gRPC-based transport with multicast and publish-subscribe patterns [2]. These protocols form the messaging layer of the emerging Agentic Web, analogous to TCP/IP for networking. They ensure reliable message delivery, routing, and security, but stop short of coordination. While agents can exchange decisions or reasoning traces, there is no mechanism to aggregate information or ensure collective convergence. This creates a gap between communication and true multi-agent coordination.

2.2 Coordination in Multi-agent LLM Systems

Prior work on multi-agent LLMs has focused on centralized orchestration, where frameworks such as AutoGen [33], MetaGPT [16], CrewAI [20], and Swarm [23] enforce structure through shared prompts, memory, and control. These systems provide role definitions, task hand-offs, structured workflows, and stopping criteria. These assumptions work well within a single organization but do not generalize to open networks where agents are independently owned.

For decentralized settings, teams often reuse the same communication protocols introduced

above (e.g., A2A, ACP, ANP) and rely entirely on the LLMs themselves to negotiate through free-form reasoning and message exchange. However, without explicit coordination mechanisms, this approach is fragile: agents fall into infinite loops, oscillating handoffs, or fail to detect completion [18, 27, 37]. For example, two agents negotiating a task may repeatedly hand it back and forth, each waiting for confirmation from the other, creating infinite loops that consume resources without ever reaching completion. These issues worsen as the number of agents grows, highlighting that message exchange alone cannot ensure progress or stability.

Classical distributed mechanisms—such as consensus protocols [7, 22], voting rules [3, 26], and auction mechanisms [12] provide such structure in traditional systems but assume static preferences and shared objectives. They cannot capture the dynamic, textual, and partially aligned reasoning of LLM agents operating in domains with diverse incentives. Recent studies on LLM voting and census-level preference aggregation [10, 34] explore some of these mechanisms in controlled, centralized settings, but real-time decentralized coordination remains an open challenge. This gap motivates the need for protocols that operate above message exchange to provide true coordination capabilities. REP addresses this challenge by introducing a dedicated coordination layer for the Agentic Web, enabling agents to move from merely communicating to collectively aligning their actions in open, decentralized networks.

2.3 Sensitivity Sharing in Distributed Systems

Sensitivity sharing has long been used to enable coordination without central control. In distributed optimization, nodes exchange gradients or dual variables rather than final solutions, as in ADMM and gradient tracking algorithms [6, 21]. In decentralized simulations, agents similarly exchange lightweight signals that summarize local state instead of sharing full datasets. This enables global calibration while preserving privacy—for example, contact tracing protocols that share infection-state updates without revealing individual health data [4, 9].

These approaches assume numerical signals and shared global objectives. In contrast, LLM-based agents reason in natural language, are independently developed, and often operate with partially aligned or competing incentives. Existing sensitivity sharing techniques cannot capture this qualitative reasoning or scale to open, heterogeneous networks. REP builds on this lineage by leveraging textual sensitivity [36] as a coordination primitive, providing the missing layer above current messaging protocols and enabling large populations of LLM agents to coordinate without centralized control.

3 Ripple Effect Protocol

REP introduces a new coordination layer for multi-agent systems, enabling agents to move beyond merely exchanging messages to aligning their actions in open, decentralized networks. At its core, REP defines a set of shared *coordination variables* and a process for *sensitivity sharing*, where agents communicate lightweight signals indicating how their decisions would change under different conditions. These sensitivities are aggregated across local neighborhoods using either gradient-style updates or LLM-based synthesis, producing a compact coordination state that guides future decisions. This structure allows diverse LLM agents to coordinate without centralized control, providing scalability and stability missing from current systems.

REP cleanly decouples internal agent cognition from the external coordination mechanism:

- **Agents think:** Each agent uses its internal reasoning (LLM or rule-based) to decide on domain actions and generate sensitivities—signals describing how its decision would change under different environmental conditions.
- **Protocols coordinate:** REP manages the exchange and aggregation of these sensitivities, updating a shared set of *coordination variables* that influence future decisions.

3.1 Workflow Overview

REP operates over a network $G = (V, E)$, where each node $i \in V$ represents an agent and edges E indicate direct communication links between neighbors $N_i = \{j \in V \mid (i, j) \in E\}$. Time proceeds in discrete **rounds** $t = 1, 2, \dots$, with each round consisting of four steps:

1. **Receive Messages:** Each agent i collects messages $M_i^t = \{(d_j^t, s_j^t) : j \in N_i\}$ containing neighbor decisions d_j^t and sensitivities s_j^t .
2. **Generate Decision & Sensitivity:** The agent applies its policy π_i to the current coordination variables θ_i^t and private constraints c_i to produce:

$$(d_i^t, s_i^t) = \pi_i(\theta_i^t, c_i)$$

where d_i^t is the domain action (e.g., order quantity, vote, fleet deployment) and s_i^t is a signal describing how the decision would change if conditions shifted.

3. **Aggregate Neighbor Sensitivities:** REP combines neighboring sensitivities to update local coordination variables:

$$\theta_i^{t+1} = \theta_i^t + \text{Agg}_{j \in N_i}(s_j^t)$$

Sensitivities may be numeric or textual, with REP applying either standard optimization rules or LLM-based synthesis as needed.

4. **Consensus (Optional):** In domains requiring group agreement, REP applies deterministic rules such as coordinate-wise median to converge on shared proposals.

Example: In a supply chain, an agent’s decision might be to order 120 units. Its sensitivity could read: *“If demand increases by 10%, increase order by 15 units; if upstream capacity improves, decrease by 5 units.”* These sensitivities are aggregated by neighbors, producing updated coordination variables like **TARGET INVENTORY** and **ORDER ADJUSTMENT FACTOR**, which then shape the next round’s decisions.

3.2 Coordination and Aggregation

At each round t , every agent i maintains a local coordination state θ_i^t that influences its next-round decision. Using its internal policy $\pi_i(\theta_i^t, c_i)$, the agent produces two outputs: a domain action d_i^t and a *sensitivity message* s_i^t describing how d_i^t would change under counterfactual conditions. These sensitivities are exchanged among neighbors \mathcal{N}_i and combined to update the agent’s local state.

REP frames this process as a generalized gradient step:

$$\theta_i^{t+1} = \theta_i^t - \eta \cdot g_i^t,$$

where η is a step size and g_i^t is a “gradient-like” signal representing the aggregated influence of neighboring agents:

$$g_i^t = \text{Agg}_\phi(\{s_j^t : j \in \mathcal{N}_i\}, \theta_i^t).$$

The operator Agg_ϕ is parameterized by a synthesis function ϕ , which determines how sensitivities are integrated. REP is modality-agnostic: in some domains, ϕ is a numeric rule such as weighted averaging or distributed gradient descent; in others, ϕ is a language model that synthesizes free-form reasoning into structured, low-dimensional updates. This approach is related to recent methods such as TextGrad [36], which treat natural language as a medium for computing gradient-like updates, but in REP these signals are used to coordinate *populations of agents* rather than optimize a single model.

This unified formulation allows REP to seamlessly handle both quantitative and qualitative coordination signals without changing the overall update rule. For example, s_j^t might encode either production adjustments in a supply chain or price/time flexibility in group scheduling.

In many domains, these local updates alone are sufficient. However, some tasks require the entire population to agree on a shared proposal, such as selecting a common meeting time or price. In these cases, REP adds a second, global consensus step after local aggregation:

$$\bar{\theta}^{t+1} = \text{Median}(\{\theta_i^{t+1} : i \in \mathcal{V}\}),$$

where \mathcal{V} is the set of all agents. This coordinate-wise median prevents extreme local outliers from destabilizing the collective decision while retaining diversity of preferences. The resulting two-level process enables REP to span a broad spectrum of coordination problems, from local alignment in sequential supply chains and resource management to explicit consensus formation in distributed decision-making.

3.3 Implementation Architecture

REP is implemented as a lightweight coordination layer that wraps existing agents without modifying their internal policies. A **REPClient** manages message exchange, sensitivity aggregation, and updates to coordination variables. The design is intentionally modular: the transport backend, aggregation rule, and consensus mechanism can each be configured independently. This makes REP portable across domains and independent of any single communication protocol or decision rule. For example, we use SLIM for multicast messaging and coordinate-wise median for consensus, but other systems or rules can be substituted without changing the protocol logic.

Configuration Interface. Each agent is initialized by specifying which implementations to use for these three components:

```
rep_client = rep.configure(
    agent=llm_agent,
    transport="slim",           # any messaging backend
    updater="textual_grad",    # or "numerical_grad"
    consensus="median_coordinate" # or other rules
)
```

Protocol Execution. During each round, agents run the same loop in parallel: receiving messages, generating local decisions and sensitivities, broadcasting them to neighbors, and performing local and (optionally) global aggregation:

```
# Protocol execution
neighbor_msg, neighbor_sensitivity = rep_client.receive()
rep_client.sync(neighbor_sensitivity) # aggregate neighbor sensitivities
decision, sensitivity = rep_client.decide()
rep_client.send(decision=decision, sensitivity=sensitivity)
```

When textual aggregation is enabled, domain-specific prompt templates guide the LLM in synthesizing free-form reasoning into structured updates. This allows the same core protocol to support diverse coordination problems—from supply chain optimization to distributed consensus—without changing agent logic or protocol implementation.

4 Experimental Setup

We evaluate REP across three domains that systematically span different network structures, incentive patterns, and coordination challenges. The Beer Game models sequential coordination in

linear supply chains (4–8 agents) with aligned incentives but delayed information flows that create bullwhip effects. Movie Coordination captures consensus formation in sparse social networks (5–20 agents), where heterogeneous preferences must be reconciled despite limited connectivity. Fishbanks examines resource allocation in fully connected networks (5–8 agents) with directly competing incentives, where individual profit maximization conflicts with collective sustainability. Together, these environments test REP across a spectrum of coordination conditions: information cascades versus multi-hop influence versus broadcast interactions; aligned, heterogeneous, and conflicting incentives; and network topologies ranging from linear to sparse to dense.

Table 1 summarizes the core elements of each domain, including the decisions made by agents, the coordination variables introduced by REP, and the structural properties of each environment.

Table 1. Agent decision variables, coordination variables, and challenges for each domain.

Domain	Agent Decisions	Coord. Variables	Network	Challenge
Beer Game	Order quantities per round	Target inventory levels	Linear chain	Information cascades and bullwhip effect
Fishbank	Fleet deployment across regions	Sustainable quotas	Fully connected	Mixed-motive allocation and over-exploitation
Movie Scheduling	Participation (Y/N) or local preference votes	Preferred price/-time thresholds	Sparse small-world network	Consensus formation with heterogeneous preferences

Our baseline represents the current state of practice in open LLM-agent networks. We use an Agent-to-Agent (A2A) approach where agents share their final decisions and free-form reasoning traces. This mirrors the message exchange patterns supported by protocols such as A2A and ACP, and the coordination style seen in multi-agent orchestration frameworks like MetaGPT, CrewAI, and OpenAI Swarm today. For example, in the Beer Game, an upstream agent might message, “Ordering 12 units because inventory is low.” While this communication enables neighbors to understand intent, it cannot be systematically aggregated to update shared coordination variables. Both REP and the baseline use identical transport, consensus rules, LLM model, and network topologies, ensuring that performance differences are solely due to REP’s introduction of structured sensitivity sharing.

All experiments use Claude-3-Haiku-20240307 (temperature = 0.1, max_tokens = 300) to ensure consistent reasoning power across conditions. Each simulation runs for 20 rounds, with 3–5 trials per configuration to account for stochasticity. Coordination accuracy is reported using domain-specific metrics, detailed in the sections describing each experiment. Final results are aggregated across trials with mean and standard deviation reporting. To validate communication efficiency, we scale the number of agents from 10 to 200 and measure wall-clock coordination time. This test demonstrates that REP’s structured design, combined with SLIM’s low-latency multicast messaging, supports efficient coordination for population scale.

5 Supply Chain Coordination (Beer Game)

The Beer Game [29] is a canonical benchmark for studying coordination failures in sequential supply chains. It consists of four stages—retailer, wholesaler, distributor, and manufacturer—arranged in a linear chain. In each round, every agent observes only its local inventory and orders from its immediate neighbor, deciding how many units to order upstream. Fixed lead

times delay feedback, meaning agents cannot immediately observe the downstream consequences of their actions. This limited visibility causes small fluctuations in customer demand to amplify as they propagate upstream, producing the well-known bullwhip effect [17].

In the baseline Agent-to-Agent (A2A) approach, agents share both order quantities and free-form reasoning traces describing the rationale behind those decisions. While this helps neighbors understand why an action was taken, it does not capture how the agent’s behavior would change under different future conditions. For instance, when a retailer places an unusually large order and explains it as “inventory low due to a local demand spike,” upstream agents still lack visibility into whether this action represents a temporary fluctuation or a longer-term shift. Without explicit modeling of decision flexibility, these free-form messages cannot be aggregated into a stable shared understanding, leaving the system vulnerable to information cascades and runaway amplification.

REP introduces structured sensitivity messages, where each agent communicates how its decision would adapt under key environmental changes. For example, “If demand increases by 10%, increase order by 15 units. If upstream capacity improves, reduce order by 5 units. Current spike seems temporary.” These sensitivities are combined across neighboring agents into low-dimensional coordination variables, such as `TARGET_INVENTORY` and `ORDER_ADJUSTMENT_FACTOR`, which guide subsequent decisions. By circulating structured decision flexibility rather than only final actions and free-form text, agents gain predictive context that allows them to distinguish transient noise from structural change, reducing instability.

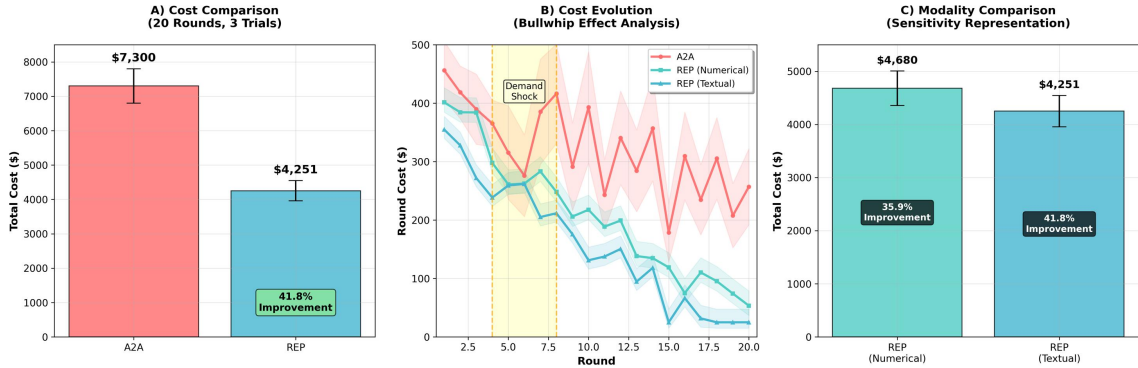


Figure 1. Supply chain coordination outcomes over 20 rounds with a demand shock introduced at round 4, where customer demand doubles. (A) Baseline A2A shows persistent bullwhip oscillations as agents repeatedly over- and under-react to incomplete information. REP reduces total supply chain cost by 41.8%, from \$7,300 (A2A) to \$4,251 (B) REP rapidly stabilizes within 3–4 rounds, preventing runaway amplification. (C) REP textual aggregation outperforms numerical aggregation, achieving lower total costs by capturing richer causal reasoning.

Insight 1: REP mitigates bullwhip effects. Under identical LLM agents and network structures, REP reduces total supply chain cost by 41.8%, from \$7,300 (A2A) to \$4,251, and stabilizes demand shocks within 3–4 rounds, compared to 10+ rounds under A2A (Fig. 1(a-b)). REP’s structured signals prevent the repeated over- and under-reactions that drive oscillations in traditional systems, leading to rapid convergence after initial disturbances.

Insight 2: Textual sensitivities outperform numerical gradients REP supports both numerical aggregation—where sensitivities are treated as structured derivatives—and textual aggregation, where natural language reasoning is synthesized into compact updates. Textual sensitivities yield lower total cost (4,251 vs 4,680, a 9.2% improvement; Fig. 1)(c) by capturing

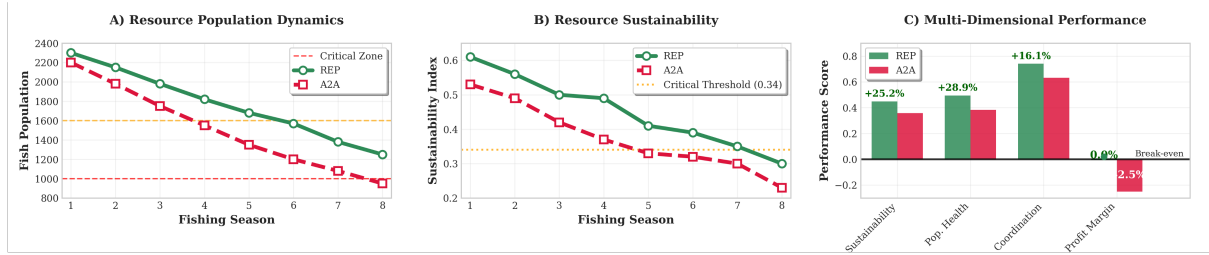


Figure 2. Fishbanks simulation with 12 LLM agents over 8 seasons. A demand for fish remains constant while overharvesting leads to eventual depletion. (A) Fish population dynamics showing delayed resource collapse under REP compared to rapid decline in the A2A baseline. (B) Sustainability index with a critical viability threshold at 0.35; REP maintains the resource above this threshold 2 seasons longer than A2A. (C) Comparison of final outcomes across normalized metrics and profit margins. REP improves sustainability (+25.2%), population health (+28.9%), and coordination (+16.1%) while avoiding the financial losses (-2.5%) experienced by A2A agents. Results averaged over five trials.

nuanced causal relationships, such as supplier behavior or distinguishing short-term volatility from systemic shifts, that numerical signals cannot express.

6 Resource Allocation (Fishbanks)

Resource allocation under sustainability constraints exemplifies the tragedy of the commons, where individually rational decisions lead to collectively irrational outcomes [15, 24]. The Fishbanks simulation models this dynamic through 12 fishing companies operating in shared waters [28]. Each company aims to maximize profits by deploying its fleet to different fishing zones, but uncoordinated strategies lead to overfishing and eventual resource collapse, harming all participants.

In the baseline Agent-to-Agent (A2A) approach, companies share their deployment decisions and reasoning—such as the number of boats sent to each zone—based on local observations of catch rates and market prices. Without additional structure, agents cannot determine whether others are pursuing short-term opportunistic gains or making long-term sustainability commitments. This information isolation drives competitive escalation: each company acts defensively, deploying aggressively to avoid being left behind, which accelerates resource depletion.

REP extends this setting by allowing companies to communicate textual sensitivities about resource conditions and coordination willingness alongside deployment decisions. For example, “Fish population shows recovery signs based on catch composition. Fleet maintenance costs justify a conservative approach if others also coordinate.”. These sensitivities capture how decisions would shift under different environmental or strategic conditions, enabling agents to infer whether others are willing to restrain harvesting for collective benefit. Aggregating these signals across companies provides a shared picture of ecosystem health and alignment, supporting voluntary, decentralized coordination without a central controller.

Insight 3: Coordination under competing incentives. Across 8-season experiments with 12 LLM agents, REP achieves 25.2% sustainability improvement and 28.9% better population health while preventing the financial losses experienced by A2A agents (-2.5%). This demonstrates REP’s capability to coordinate in tragedy-of-commons scenarios—the most challenging multi-agent setting where individual and collective objectives directly conflict. Unlike traditional approaches that result in resource depletion harming all participants, REP enables win-win outcomes through voluntary coordination. This is visualized in figure 2

Insight 4: Transparency enables conditional cooperation. The shared sensitivities include both ecosystem assessments and signals of willingness to cooperate. Companies can

condition their behavior on others’ stated intentions—e.g., “We can maintain a conservative fleet size if others do the same”. This transparency allows trust to emerge without explicit enforcement, shifting agents from pure profit-maximization toward coordinated stewardship of the shared resource.

7 Preference Aggregation (Movie Coordination)

Group decision-making in sparse social networks presents a fundamental coordination challenge: agents must converge on a shared outcome despite heterogeneous preferences and limited connectivity. Each agent represents an individual with distinct budget and scheduling constraints, communicating only with immediate neighbors. As a result, preferences must propagate indirectly through the network.

In the baseline Agent-to-Agent (A2A) setting, agents exchange their final participation choices and free-form reasoning. Without structured aggregation, this leads to information isolation: small clusters of neighbors may agree locally, but there is no mechanism to align the full network. This produces fragmented, unstable outcomes, where local proposals conflict and the group fails to converge on a global consensus.

REP introduces a structured preference-sharing process that unfolds in two stages during each round. In the first stage, every agent communicates both its current participation decision and a set of preference sensitivities that capture how its utility would change under adjustments to the shared coordination variables. These sensitivities can take numerical form, such as $\partial U / \partial \text{price} = -0.8$ to indicate high price sensitivity, or textual form, such as “*Price matters more than timing — strongly budget constrained.*” Messages from neighboring agents are then aggregated for the two global coordination variables, **TIME** and **PRICE**, with local updates nudging these variables toward values that reflect the collective preference landscape. In the second stage, agents vote again on the updated proposal, and REP applies a median consensus rule to produce the next shared state. This consensus mechanism provides stability by preventing extreme local outliers from destabilizing the group decision. Through repeated rounds, sensitivities ripple across multiple network hops, allowing the population to converge even when many agents are not directly connected.

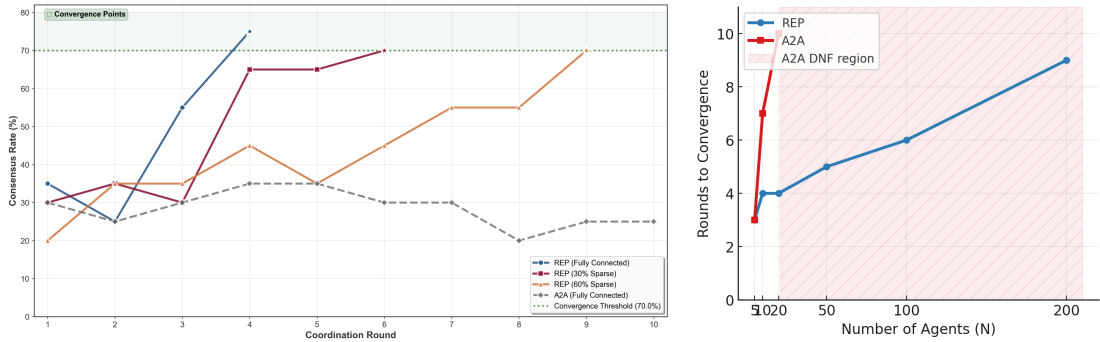


Figure 3. REP performance under network sparsity and population scaling. (Left) REP maintains effective coordination even in sparse social networks (30% and 60% sparse), with only gradual increases in convergence rounds ($4 \rightarrow 6 \rightarrow 9$) while sustaining 70–75% consensus. Stars mark convergence points where the 70% threshold is reached. In contrast, A2A fails to achieve meaningful coordination even when fully connected, showing that sensitivity sharing is essential for global alignment. (Right) As population size grows from 5 to 200 agents, REP continues to converge reliably (3–15 rounds), while A2A requires 7–10 rounds at small sizes and fails entirely beyond 20 agents (shaded DNF region). Together, these results demonstrate REP’s robustness to both sparse connectivity and large-scale populations.

Insight 5: Coordination effectiveness across network sparsity. We evaluate REP on a 20-agent network under three connectivity regimes: fully connected, 30% sparse, and 60% sparse. **REP** achieves **70–75% consensus** across all conditions (Fig. 3(left)), with only gradual increases in convergence time: Round 4 \rightarrow Round 6 \rightarrow Round 9 as connectivity decreases. This resilience arises because sensitivity signals propagate indirectly, allowing agents to align even when some connections are missing. In contrast, **A2A** fails to reach meaningful agreement, plateauing at **35% maximum consensus** even in fully connected networks. Without structured aggregation, each agent is overwhelmed by conflicting inputs, leading to cognitive overload and preventing the system from integrating distributed reasoning into a stable global outcome.

Insight 6: Scalability to larger populations To evaluate REP’s scalability, we increase the network size from 5 to 200 agents at fixed sparsity. Fig. 3(right) projects the number of rounds required to reach convergence as population size grows. REP maintains stable performance: convergence occurs in just 3–9 rounds across the entire range, with only modest growth as the network scales. In contrast, A2A requires 7–10 rounds even for small populations (≤ 20 agents) and fails to converge entirely beyond this point, as indicated by the shaded DNF region. Because per-round message cost grows near-linearly with N under SLIM multicast, fewer rounds directly translate to lower total communication. Wall-clock profiling further shows that communication is negligible: at 200 agents, sensitivity sharing is 3% of runtime, with the rest dominated by LLM inference (38%) and wait time (59%). Thus, REP’s scalability is promising for designing an agentic web.

8 Conclusion and Future Work

We introduced the Ripple Effect Protocol (REP), a lightweight coordination layer that enables LLM-based agents to share sensitivities rather than only final decisions. By operating above existing messaging frameworks like A2A and SLIM, REP turns coordination into a protocol primitive, allowing diverse agents to align their actions across different network structures, incentive regimes, and temporal dynamics. Across three domains—supply chains, shared-resource management, and group decision-making—REP consistently outperformed traditional decision-only communication by accelerating convergence, improving stability, and enabling coordination at population scale.

Our current work assumes cooperative, non-malicious agents and synchronous interactions, which simplifies reasoning but limits applicability to fully open, decentralized networks. Future work will extend REP along two key directions. First, we plan to develop Byzantine fault-tolerant mechanisms that allow coordination even when some agents misreport sensitivities or behave adversarially, ensuring robustness in competitive or adversarial settings. Second, we will generalize REP to support asynchronous multi-step interactions, where agents operate on partially overlapping timelines and must integrate information without requiring strict round synchronization. These enhancements will broaden REP’s applicability to real-world, large-scale deployments such as financial markets, distributed infrastructure, and cross-organization collaboration. We will release the REP SDK along with this paper and provide a sandbox environment to test scalability of coordination and transport protocols. We hope our research accelerates the development of the Agentic Web.

9 Acknowledgement

We thank Anthropic for providing API credits; and MIT MLC for funding our research.

References

- [1] a2aproject. a2aproject/a2a: An open protocol enabling communication and interoperability between opaque agentic applications. <https://github.com/a2aproject/A2A>, 2025. 2.1
- [2] AGNTCY. SLIM: Secure Low-Latency Interactive Messaging. <https://github.com/agtncy/slim>, 2025. 1, 2.1
- [3] K. J. Arrow. *Social Choice and Individual Values*. Yale University Press, 1951. 2.2
- [4] N. Attrapadung, K. Isayama, K. Sadakane, and K. Tozawa. Secure parallel computation with oblivious state transitions. In *Proceedings of the 2024 ACM SIGSAC Conference on Computer and Communications Security*, Salt Lake City, UT, USA, 2024. ACM. 2.3
- [5] K. Blair and I. Faro. An open-source protocol for AI agents to interact. IBM Research Blog, May 2025. Available at: <https://research.ibm.com/blog/agent-communication-protocol-ai>. 1
- [6] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*. 2.3
- [7] M. Castro and B. Liskov. Practical byzantine fault tolerance. *ACM Transactions on Computer Systems (TOCS)*, 17(4):398–461, 1999. 1, 2.2
- [8] W. Chen, Y. Su, J. Zuo, C. Yang, C. Yuan, C.-M. Chan, H. Yu, Y. Lu, Y.-H. Hung, C. Qian, et al. Agentverse: Facilitating multi-agent collaboration and exploring emergent behaviors. *arXiv preprint arXiv:2308.10848*, 2023. 1
- [9] A. Chopra, A. Quera-Bofarull, N. Giray-Kuru, M. Wooldridge, and R. Raskar. Private agent-based modeling. In *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, page 10. IFAAMAS, 2024. 2.3
- [10] A. Chopra, S. Kumar, N. G. Kuru, R. Raskar, and A. Quera-Bofarull. On the limits of agency in agent-based models. In *Proceedings of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025)*, page 500–509, Detroit, Michigan, USA, 2025. International Foundation for Autonomous Agents and Multiagent Systems (IFAAMAS). 2.2
- [11] R. Cooper, D. V. DeJong, R. Forsythe, and T. W. Ross. Selection criteria in coordination games: Some experimental results. *The American Economic Review*, 80(1):218–233, 1990. 1
- [12] P. Cramton, Y. Shoham, and R. Steinberg. *Combinatorial Auctions*. MIT Press, Cambridge, MA, 2006. 2.2
- [13] A. Ehtesham, A. Singh, G. K. Gupta, and S. Kumar. A survey of agent interoperability protocols: Model context protocol (mcp), agent communication protocol (acp), agent-to-agent protocol (a2a), and agent network protocol (anp). *arXiv preprint arXiv:2505.02279*, 2025. 2.1
- [14] Google for Developers. Announcing the agent2agent protocol (a2a). <https://developers.googleblog.com/en/a2a-a-new-era-of-agent-interoperability/>, April 2025. 1
- [15] G. Hardin. The tragedy of the commons. *Science*, 1968. 6
- [16] S. Hong, M. Zhuge, J. Chen, X. Zheng, Y. Cheng, C. Zhang, J. Wang, Z. Wang, S. K. S. Yau, Z. Lin, et al. Metagpt: Meta programming for a multi-agent collaborative framework. *arXiv preprint arXiv:2308.00352*, 2023. 1, 2.2

-
- [17] H. L. Lee, V. Padmanabhan, and S. Whang. Information distortion in a supply chain: The bullwhip effect. *Management Science*, 43(4):546–558, 1997. 1, 5
- [18] Y. Louck, A. Stulman, and A. Dvir. Proposal for improving google a2a protocol: Safeguarding sensitive data in multi-agent systems. *arXiv preprint arXiv:2505.12490*, 2025. doi: 10.48550/arXiv.2505.12490. URL <https://arxiv.org/abs/2505.12490>. 2.2
- [19] S. Marro, E. L. Malfa, J. Wright, G. Li, N. Shadbolt, M. Wooldridge, and P. Torr. A scalable communication protocol for networks of large language models. *arXiv preprint arXiv:2410.11905*, 2024. doi: 10.48550/arXiv.2410.11905. URL <https://arxiv.org/abs/2410.11905>. 1, 2.1
- [20] J. Moura. crewai: The leading multi-agent framework. URL <https://github.com/crewAIInc/crewAI>. 2.2
- [21] A. Nedic, A. Olshevsky, W. Shi, and Y. Zhang. Stochastic gradient-push for strongly convex optimization over a time-varying directed network. In *Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on*, pages 3346–3351. IEEE, 2014. 2.3
- [22] D. Ongaro and J. Ousterhout. In search of an understandable consensus algorithm. In *2014 USENIX annual technical conference (USENIX ATC 14)*, pages 305–319, 2014. 1, 2.2
- [23] OpenAI. Swarm: An experimental framework for multi-agent orchestration. <https://github.com/openai/swarm>, 2024. URL <https://github.com/openai/swarm>. GitHub repository; a production-ready version is now available as the OpenAI Agents SDK. 2.2
- [24] E. Ostrom. *Governing the commons*. 1990. 6
- [25] T. Rashid, M. Samvelyan, C. Schroeder, G. Farquhar, J. Foerster, and S. Whiteson. Qmix: Monotonic value function factorisation for deep multi-agent reinforcement learning. In *International conference on machine learning*, pages 4295–4304. PMLR, 2018. 1
- [26] A. Sen. *Collective Choice and Social Welfare*. Holden-Day, San Francisco, 1970. 2.2
- [27] B. Singh. Overcoming the top 5 challenges in deploying llm agents into production. <https://medium.com/@bhaiyasingh/overcoming-the-top-5-challenges-in-deploying-llm-agents-into-production-937b27ebd1c1>, 2025. 2.2
- [28] J. Stermann and A. King. Fishbanks: A renewable resource management simulation. <https://mitsloan.mit.edu/teaching-resources-library/fishbanks-a-renewable-resource-management-simulation>, 2017. 1, 6
- [29] J. D. Stermann. Modeling managerial behavior: Misperceptions of feedback in a simple supply chain. *Management Science*, 35(3):321–339, 1989. 1, 5
- [30] P. Stone and M. Veloso. Multiagent systems: A survey from a machine learning perspective. *Autonomous Robots*, 8(3):345–383, 2000. 1
- [31] P. Sunehag, G. Lever, A. Gruslys, W. M. Czarnecki, V. Zambaldi, M. Jaderberg, M. Lanctot, N. Sonnerat, J. Z. Leibo, K. Tuyls, et al. Value-decomposition networks for cooperative multi-agent learning. *arXiv preprint arXiv:1706.05296*, 2017. 1
- [32] Y. Wang, S. Guo, Y. Pan, Z. Su, F. Chen, T. H. Luan, P. Li, J. Kang, and D. Niyato. Internet of agents: Fundamentals, applications, and challenges. *arXiv preprint arXiv:2505.07176*, 2025. 1

-
- [33] Q. Wu, G. Bansal, J. Zhang, Y. Wu, B. Li, E. Zhu, L. Jiang, X. Zhang, S. Zhang, J. Liu, A. H. Awadallah, R. W. White, D. Burger, and C. Wang. Autogen: Enabling next-gen llm applications via multi-agent conversation framework. *arXiv preprint arXiv:2308.08155*, 2023. URL <https://arxiv.org/abs/2308.08155>. 2.2
- [34] J. C. Yang, M. Korecki, D. Dailisan, C. I. Hausladen, and D. Helbing. Llm voting: Human choices and ai collective decision making. In *Proceedings of the 41st International Conference on Machine Learning*. PMLR, 2024. 2.2
- [35] Y. Yang, M. Ma, Y. Huang, H. Chai, C. Gong, H. Geng, Y. Zhou, Y. Wen, M. Fang, M. Chen, S. Gu, M. Jin, C. J. Spanos, Y. Yang, P. Abbeel, D. Song, W. Zhang, and J. Wang. Agentic web: Weaving the next web with ai agents. *arXiv preprint arXiv:2507.21206*, 2025. URL <https://arxiv.org/abs/2507.21206>. 1
- [36] M. Yuksekgonul, F. Bianchi, J. Boen, S. Liu, P. Lu, Z. Huang, C. Guestrin, and J. Zou. Optimizing generative ai by backpropagating language model feedback. *Nature*, 639:609–616, 2025. 2.3, 3.2
- [37] B. Zhang, Y. Tan, Y. Shen, A. Salem, M. Backes, S. Zannettou, and Y. Zhang. Breaking agents: Compromising autonomous llm agents through malfunction amplification. *arXiv preprint arXiv:2407.20859*, 2024. doi: 10.48550/arXiv.2407.20859. URL <https://arxiv.org/abs/2407.20859>. 2.2