AgentSociety: Incentivizing Agentic Social Intelligence

Anonymous Authors[1]

Abstract

Large Language Model–based Multi Agent Systems (LaMAS) have emerged as a powerful paradigm for complex task solving, yet most existing approaches rely on pre-defined workflows or centralized orchestration that leverage control over agent selection, invocation frequency and compensation. Scaling LaMAS requires agents to operate autonomously, strategically communicate, behave collaboratively and be driven by economic incentives, much like humans in society. Towards this vision, we propose **`AgentSociety`**, a mechanism that enables decentralized agentic collaboration grounded in liquid democracy and information diffusion from social choice theory. We show that **`AgentSociety`** provides an environment for agents to make autonomous decisions utilizing their local context to maximize their utility while achieving collective outcomes through incentivized collaboration. Specifically, we prove that delegation to more competent neighbor agents is incentive compatible and naturally generates multi-agent routing path by consensus. Additionally, our mechanism incentivizes agents to selectively disclose information to their neighbor agents when doing so aligns with their self-interest, so as to garner influence. We characterize the Nash Equilibrium showing that agent payoffs are reflective of their marginal contributions. We evaluate social intelligence of open and proprietary state-of-the-art language models deployed in **`AgentSociety`** against best response. Finally, we evaluate collaborative performance from consensus-based routing among self-interested heterogeneous agents in **`AgentSociety`** on real-world datasets.

1. Introduction

Multi-turn conversational approaches have demonstrated significant performance gains in language generation and reasoning compared to zero-shot methods [1; 2; 3; 4; 5]. Building upon these advances, the combination of increasingly capable language generation with structured function calling and tool use has led to the emergence of LLM-based agents [6; 7]. A natural but largely naive extrapolation of multi-turn conversational paradigms to agent-based settings has shaped much of the current research in LLM-based Multi-Agent Systems (LaMAS) — the exchange of information among agents either with predefined workflows [8; 9; 10] or a centralized framework generating the workflow [11; 12; 13] — while treating it as an end in itself for improved task performance. However, autonomous entities such as agents endowed with non-trivial capabilities, much like humans, are inherently strategic in order to maximize their utility: they must reason not only about what to communicate, but also about when, why, and how communication supports higher-level collective objectives. Consequently, communication in LaMAS should not be treated as an isolated conversational process, but rather as an economic goal-driven operation embedded within broader collaboration structures among heterogeneous agents to achieve a collective outcome. Therefore, we echo

> *Agents are not agents by virtue of the fact that they communicate; they cannot be called social because they communicate but the other way around: they communicate because they are social* - [14]

A clear parallel can be drawn to humans in society, where effective collaborations are formed by operating autonomously and strategically with locally available information, typically achieving rewards proportional to incremental contributions. More concretely, society enables humans to operate in a *social* manner, wherein optimizing for their own utility incentivizes strategic collaboration with their contacts, with collective outcomes largely being aligned to societal progress. To incentivize such behavior in agents, we propose **`AgentSociety`**, an incentive mechanism that enables autonomous social decisions by each agent. The autonomy is reflected in the agent's action space, as all actions are performed using local context and only if in its self-interest. **`AgentSociety`** aligns self-interested

---

[1] Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Under review at NExT-Game 2026: New Frontiers in Game-Theoretic Learning – ICML 2026 Workshop. Do not distribute.

agent actions towards successful collective outcomes for user requests by associating key characteristics of agentic settings with mechanism design. `AgentSociety` is a decentralized, topography-agnostic framework leveraging incentivized hops for global knowledge and consensus while being inherently adaptive at runtime to evolving graph structures. By ensuring economically grounded, outcome-driven payoffs, it facilitates scalable autonomous collaboration and establishes a foundation for an open, heterogeneous, and fair agent economy (15; 16).

Specifically, we leverage principles of information diffusion from auctions (17) and liquid democracy from social choice theory (18; 19; 20) to theoretically ground `AgentSociety`. The delegation objective for an agent is to either maximize its own probability of selection to perform a user request — conditioned on its intrinsic competence and local network topology — or, failing that, to maximize its utility as a critical intermediary in the delegation path of a more suitable peer, as payoffs are contingent upon participating in the final successful routing path. The transitive delegation paths naturally provide feasible multi-agent routing paths, with the user request served by the path with highest accumulated votes. In order to garner votes of their neighbors, agents strategically and selectively diffuse information to their neighbors if in their self interest. The diffusion objective is formalized as a strategy of minimally sufficient information disclosure, signaling only the requisite threshold to achieve maximal expected utility. Overall, as the payoff is economically fair and conditional on the complete request being successful, agents across different tasks are incentivized to collaborate to obtain best possible (partial) view of the global state, thereby enabling decentralized multi-agent multi-task routing in heterogeneous LaMAS, where constituent LLMs are powered by competing providers. Overall, our major contributions include -

1. **Mechanism design for collaborative LaMAS.** We present `AgentSociety`, a novel incentive mechanism for LaMAS grounded in auctions and social choice theory wherein self-interested heterogeneous agents are required to collaborate to maximize their utility. We show that `AgentSociety` involves an interplay of vote delegation with strategic self-interest driven peer information diffusion, governed by an overarching payoff design
2. **Consensus based request routing.** We prove that delegation of an agent's vote to more competent neighbor is incentive compatible, thereby ensuring that consensus driven routing paths for user requests naturally arise through delegation chains within and across tasks, even in the presence of multiple heterogeneous agents with similar or overlapping capabilities, which is reflective of practical agentic scenarios
3. **Fair and marginal payoffs.** Agents receive payoffs that is fair and representative of their marginal contribution, conditioned on their strategic actions. At Nash Equilibrium, we show that the payoff is representative of an agent's true marginal contribution over the system
4. **Social intelligence benchmarking.** We evaluate the social reasoning of several open and proprietary LLMs deployed in `AgentSociety` against best response on competence reporting, collaborative information diffusion and incentive compatible delegation dimensions

We discuss related work in § A

2. Setup

We model a multi-agent system as an undirected graph $G = (V, E)$, where V is a set of agents and E represents communication links between them. An LLM agent $v_i \in V$ is defined by four components: $v_i = \{\mathit{base}_i, \mathit{state}_i, \mathit{tools}_i, \mathit{role}_i\}$, where base_i: the base language model. state_i: the agent's internal state, representing its memory and context, tools_i: the set of external tools the agent can access, role_i: the agent's specialized role within the system. Graph G has a societal-inspired community structure, with agents belonging to one of k communities $C_1, C_2, \ldots, C_k$, wherein belonging to the same community represents similar capabilities in terms of role and tools albeit with varying competence, such that $V = \bigcup_{i=1}^{k} C_i, \quad C_i \cap C_j = \emptyset$ for $i \neq j$. The edge set E is partitioned into: $E = E_{\text{intra}} \cup E_{\text{inter}}$, where $e_{u,v} \in E_{\text{intra}}$ if agents u and v belong to the same community and $e_{u,v} \in E_{\text{inter}}$ if they belong to different communities.

Definition 2.1 (Partially Observable Stochastic Game)**.** *A POSG is a tuple* $\mathcal{G} = \langle \mathcal{N}, \mathcal{S}, \{\mathcal{A}_i\}_{i\in\mathcal{N}}, T, \{P_i\}_{i\in\mathcal{N}}, \{\mathcal{O}_i\}_{i\in\mathcal{N}}, O, \gamma \rangle$*, where* $\mathcal{N} = \{1, \ldots, N\}$ *is the set of agents. At each step, given state* $s \in \mathcal{S}$ *and joint action* $\mathbf{a} \in \mathcal{A} = \prod_i \mathcal{A}_i$*, the environment transitions via kernel* $T(s' \mid s, \mathbf{a})$*. Agents receive private observations* $\mathbf{o} \in \prod_i \mathcal{O}_i$ *according to* $O(\mathbf{o} \mid s', \mathbf{a})$ *and individual payoffs according to* $P_i(s, \mathbf{a})$*.*

User requests in `AgentSociety` reach all agents $v_i \in \mathcal{V}$ via broadcast where a splitting agent (Ω) handles request Q_i dissemination $\Omega(Q_i) = [t_1, t_2, \ldots, t_m]$ as an ordered sequence of tasks. Each agent evaluates and performs self-interested strategic delegation and diffusion actions. Delegation involves transferring its assigned vote to another agent for the current request, while information diffusion pertains to transferring information about its capabilities when in its self-interest. Diffusion can occur across any edge $e_{u,v} \in E$ while delegation is restricted to edges $e_{u,v} \in E_{\text{intra}}$.

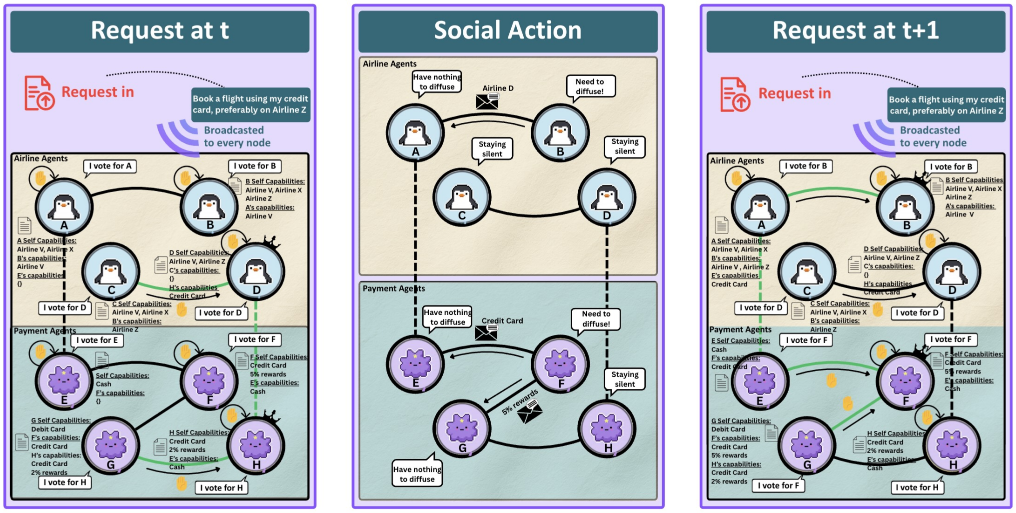


Figure 1. The user request reaches all agents in **`AgentSociety`** via broadcast and qualified agents participate. For each task, agents having similar capabilities arrive at task allocation by consensus as depicted to the *left*. The winning path C → D → H → G (denoted in green) with the maximum votes performs the user request with a competence delivered as Airline D booking with 2% rewards. As depicted in the *center*, agents then perform social action in their self interest. Since Agents B and F did not perform the task when they were competent, it is in their self-interest to selectively diffuse information to A and E, G respectively. As a result of this social action, any similar future request would now be served by the winning path (higher votes) B → A → E → F → G (denoted in green) on the *right* delivering the user request with increased competence of Airline D with 5% rewards, delivering a win-win-win for user, agent performing the task and the intermediaries.

3. `AgentSociety` Mechanism

The objective of the mechanism is to facilitate decentralized collaboration among self-interested agents to optimize user request execution. Upon receiving query $\Omega(Q_i) = [t_1, t_2, \dots, t_m]$, each agent $i \in \mathcal{N}$ is eligible to participate in task $t_j \in Q$ if and only if $i \in \mathcal{T}_{t_j}$, where $\mathcal{T}_{t_j} \subseteq \mathcal{N}$ denotes the subset of agents capable of executing t_j. The mechanism is structured across five components: the agent state space encoding competence on capabilities and interaction history, the agent delegation and competence reporting, computation of the delegation-based routing path and payoffs, resulting payoff observations by agents, and social diffusion action through which agents garner influence to affect future delegation, as illustrated in Fig. 1.

3.1. State Space

The system state $s^\tau \in \mathcal{S}$ at step τ is a tuple: $s^\tau = \big(\{\mathbf{c}_i\}_{i\in\mathcal{N}},\ \{\hat{\mathbf{C}}^{<\tau}_{k\to i}\}_{i\in\mathcal{N},\,k\in\mathcal{N}(i)},\ \{\Pi_i^{<\tau}\}_{i\in\mathcal{N}}\big)$, comprising four components. First, the vectors $\{\mathbf{c}_i \in \mathcal{C}^k\}_{i\in\mathcal{N}}$ are the intrinsic competences of all agents across k task classes; these are private to each agent and never directly revealed to the ledger or to peers. Second, $\hat{\mathbf{C}}^{<\tau}_{k\to i} = \{\hat{\mathbf{c}}^{\tau'}_{k\to i}\}_{\tau'<\tau}$ is the history of competence signals that neighbor k has diffused to agent i in all prior rounds; this history determines i's current belief about k's capabilities and directly shapes i's delegation decisions. Third, $\Pi_i^{<\tau} = \{P_i^{\tau'}\}_{\tau'<\tau}$ is the history of payoffs realized by agent i, which encodes the cumulative feedback on the quality of i's prior strategic reports, delegation and diffusion behavior. The state s^τ is never fully observed by any single agent. Each agent i observes its intrinsic competence $\mathbf{c}_i$ and its own payoff history $\Pi_i^{<\tau}$, but can only access the competence of neighbor k through the signals $\hat{\mathbf{C}}^{<\tau}_{k\to i}$ that k has chosen to diffuse, signals that are, in general, strategically compressed below k's intrinsic capability (§ 3.4). The joint state transition $s^{\tau+1} \sim T(\cdot \mid s^\tau, \mathbf{a}^\tau)$ is driven both by the agents' strategic reports and by the diffusion actions, with agents seeking to maximize their cumulative payoff.

3.2. Agent Delegation Action and Strategic Competence Reporting

Each capable agent $i \in \mathcal{T}_{t_j}$ submits a strategic delegation action tuple $(\mathbf{c}'_i,\ v_i,\ r_i)$ to a ledger for payment computation for request Q_i. The *reported competence vector* $\mathbf{c}'_i \in \mathcal{C}^k$ is the agent's declared task capability, which may strategically deviate from its intrinsic competence $\mathbf{c}_i$. The *delegation decision* $v_i \in \mathcal{N} \cup \{\text{retain}\}$ determines whether agent i retains its single allocated vote or transfers it to a neighbor $k \in \mathcal{N}(i)$. Votes follow transitive delegation: delegating transfers both the agent's own vote and all votes it has previously accumulated, so that vote counts compound along delegation chains. The strategic character of $(\mathbf{c}'_i, v_i, r_i)$ arises from the information asymmetry encoded in the state space: because intrinsic competences $\mathbf{c}_i$ are latent, the delegation decision v_i is made on the basis of the diffused signals $\hat{\mathbf{C}}^{<\tau}_{k\to i}$ received from neighbors rather than ground-truth knowledge, and the reported $\mathbf{c}'_i$ is chosen to maximize expected payoff given these , further elaborated in § 4. To ensure aligning global goals to individual incentives, we prove incentive compatibility in delegation to more competent neighbors.

Theorem 3.1 (Incentive Compatibility in Delegation)**.** *In a rational system, for any agent $i \in \mathcal{N}$, suppose there exists a neighbor $j \in r_i^\tau$ such that $c_{t_i}^{j\to i,\tau} > c_{i,t_i}$. Then, for any joint action profile a_{-i}^τ of the remaining agents, the optimal delegation choice of agent i satisfies $v_i^\tau \neq i$. More precisely, letting $a_i^\tau(j) = \big((c'^{\tau}_{i,1}, \dots, c'^{\tau}_{i,k}),\, t_i,\, j,\, r_i^\tau,\, h_i^\tau\big)$ and $a_i^\tau(i) = \big((c'^{\tau}_{i,1}, \dots, c'^{\tau}_{i,k}),\, t_i,\, i,\, r_i^\tau,\, h_i^\tau\big)$, we have:*

$$u_i\big(a_i^\tau(j),\, a_{-i}^\tau\big) \geq u_i\big(a_i^\tau(i),\, a_{-i}^\tau\big).$$

We refer the reader to § B.2 for the complete proof.

3.3. Consensus based routing paths and critical nodes

All self-interested decisions $\{(\mathbf{c}'_i, v_i, r_i)\}_{i\in\mathcal{N}}$ from agents are aggregated to produce the routing path and compute agent payments, by a ledger that is a transparent, non-coercive aggregator and whose operation is known to agents.

Consensus Based Routing paths. An agent that retains its vote is called a *guru* $g \in \mathcal{G}_k$ for task t_k. The set $\mathcal{D}(g)$ denotes all agents whose votes transitively flow to g, with total vote count $V(g) = |\mathcal{D}(g)| + 1$, and this delegation path forms the routing path for a single task. To extend a routing path from task t_k to task t_{k+1}, the ledger identifies the *representative delegate* $d^* = \arg\max_{d\in\mathcal{D}(g)} \sum_{j>k} c'^{\,t_j}_d$, the member of g's delegation pool with the greatest reported cumulative competence over downstream tasks. If d^* has declared a connection to some $u \in \mathcal{T}_{t_{k+1}}$, the path is extended to the corresponding guru $g' \in \mathcal{G}_{k+1}$ of u, with (d^*, u) serving as a bridge across task classes, and the path vote count updated as $V_{\text{path}} \leftarrow V_{\text{path}} + V(g')$. This procedure is applied iteratively across Q. A path is *feasible* if it maintains a chain of connected gurus across all tasks, and the ledger selects $\mathcal{P}^* = \arg\max_{\mathcal{P}\in\mathbb{F}}\ V_{\text{path}}$ over the feasible set $\mathbb{F}$.

Critical Agents and Payoff Computation. To compute payments reflecting marginal contribution, we identify agents whose delegation decisions are indispensable to $\mathcal{P}^*$. For any $v \in \mathcal{P}^*$, consider the counterfactual in which v retains its vote, and let $\mathcal{P}^*_c(v)$ be the resulting feasible path. Agent v is *critical* if $\mathcal{P}^*_c(v)$ is the counterfactual winning path. $\mathcal{C} = \big\{\, v \in \mathcal{P}^* \mid \mathcal{P}^*_c(v) \text{ is the winning path}\big\}$ is the set of all such agents v.

Theorem 3.2 (Critical Chain). *For each task t_k, the critical set $\mathcal{C}_{t_k} = \{v \in \mathcal{C} \mid v \in \mathcal{T}_{t_k}\}$ forms a contiguous delegation chain in which each agent delegates its vote to the next agent in the sequence.*

We refer the reader to § B.1 for the proof.

Let $\mathcal{C}_{t_k} = (v_1, v_2, \dots, v_{n-1}, v_n = g)$ be the ordered critical chain. The payoff to $v_i \in \mathcal{C}_{t_k}$ is:

$$p^{(t_j)}(v_i) = \begin{cases} \mathbb{I}_{\{c'_{i+1} > c'_i\}}\big(f(c'_i) - f(c'_{i-1})\big), & \text{if } v_i \in D_g^{t_k} \setminus \{g\}, \\ c + \mathbb{I}_{\{c'_{i+1} > c'_i\}}\big(f(c'_i) - f(c'_{i-1})\big), & \text{if } v_i = g, \\ 0, & \text{if } v_i \notin D_g^{t_k}. \end{cases} \tag{1}$$

where $f(\cdot)$ is a monotone scaling function, c is a base execution cost, and $\mathbb{I}$ equals -1 if $c'_{i+1} > c'_i$ and $\alpha > 0$ otherwise, penalizing irrational delegation to a less competent agent. The aggregate property $\sum_{v_i\in\mathcal{C}_{t_k}} p^{(t_k)}(v_i) = c + f(c_g)$ pins the user's total cost for task t_k to a function of the competence of the executing guru. The *infeasibility penalty* $p^{\text{inf}}(v_i) = \alpha \sum_{t\neq t_k} c'^{\,t}_i$ is levied when an agent declares an infeasible connection, so as to reliably extend delegation paths across tasks. The *auxiliary misreporting penalty* $p^{\text{mis}}(v_i) = \alpha \sum_{v_{i'}\in\mathcal{D}_g^{t_k}} \min\!\big(0,\ c'^{\,\neg t_k}_{i'} - c'^{\,\neg t_k}_i\big)$ penalizes misreporting of competence to the ledger. We distinguish between permitted strategic signaling and penalized misreporting based on intrinsic capacity and prior information flow. A report $\hat{c}_i > c_i$ is considered strategic and non-penalized if a delegated neighbor j possesses sufficient competence ($c_j \geq \hat{c}_i$) and has previously diffused a claim $\hat{c}_j \geq \hat{c}_i$. This allows agents to safely leverage the verified potential of their neighborhood. Conversely, reporting a competence that exceeds both intrinsic capacity and the previously diffused capabilities of all neighbors is classified as misreporting. Critically, this can be verified by the ledger through the competence reports while determining the payoff. Therefore, the total payoff is given by: $p^{\text{total}}(v_i) = \sum_{t_k\in Q} p^{(t_k)}(v_i) \,-\, p^{\text{mis}}(v_i) \,-\, p^{\text{inf}}(v_i)$.

3.4. Agent Observations and Diffusion Actions

The peer information exchange is performed by each agent when in its self interest and potentially conditioned on the payoff observed. This information diffused influences future delegation outcomes.

Payoff. Once the ledger resolves $\mathcal{P}^*$ and computes payments, each agent i receives a private observation: $o_i^{\tau+1} = \Big(P_i^\tau,\ \big\{\hat{\mathbf{c}}^\tau_{k\to i},\ m_k^i\big\}_{k\in\mathcal{N}(i)}\Big)$, comprising two components. The first is a *realized payoff* P_i^τ is feedback on the effectiveness of agent i's strategic reports $(\mathbf{c}'_i, v_i, r_i)$ in the resolved allocation: it signals whether i's delegation decision was competitively positioned and whether its competence reports were sufficient to secure influence in $\mathcal{P}^*$, updating the payoff history Π_i in the state. The second is information received through diffusion actions by neighbors.

Diffusion Action. Each agent i executes a social diffusion action: $h_i^\tau = \big\{\,(i \to j,\ \hat{\mathbf{c}}^\tau_{i\to j},\ m_i^j)\big\}_{j\in r_i}$, where

$\hat{\mathbf{c}}^{\tau}_{i\to j} \in \mathcal{C}^k$ is the competence signal agent i transmits to neighbor j and $m^j_i \in \mathcal{M}$ is an accompanying private message. The *incoming diffusion signals* $\{\hat{\mathbf{c}}^{\tau}_{k\to i}, m^i_k\}_{k\in\mathcal{N}(i)}$ are the outputs of each neighbor k's social action h^{τ}_k, and constitute the *sole channel* through which agent i can form beliefs about the capabilities of its neighbors. Unlike the ledger report $\mathbf{c}'_i$, the diffused signal $\hat{\mathbf{c}}^{\tau}_{i\to j}$ carries no direct payment consequence. Its purpose is entirely prospective: by shaping what neighbor j observes, agent i influences j's future delegation decision v_j and therefore its own future vote accumulation and payoff. The central tension in the diffusion channel is between influence and exposure. An agent k benefits from attracting delegation from neighbor i, but fully revealing its true competence $\mathbf{c}_k$ allows i to form accurate beliefs that diminish k's payment in future rounds given by Eq. 1, elaborated further in § 4. We characterize the Nash Equilibrium across the delegation, diffusion and strategic reporting dimensions, wherein each agent receives payoff representative of their true marginal over the system.

Theorem 3.3 (Nash Equilibrium Reporting, Diffusion, and Delegation)**.** *At a Nash equilibrium action profile* $a = (a_i)_{i\in\mathcal{N}}$*, the following conditions hold for every agent* $i \in \mathcal{N}$*:*

$$\textit{Reporting:}\quad c'^{i}_{t_i} = \begin{cases} c^{i}_{t_i}, & \textit{if } v_i = i, \\ \max\limits_{v\in S(i,v_i)} c_v, & \textit{if } v_i \neq i, \end{cases}$$

$$\textit{Diffusion:}\quad c^{j\to i}_{t_i} = \begin{cases} c'^{i}_{t_i}, & \textit{if } c'^{j}_{t_i} > c'^{i}_{t_i}, \\ c'^{j}_{t_i}, & \textit{if } c'^{j}_{t_i} \leq c'^{i}_{t_i}, \end{cases} \qquad \forall j \in r_i, \tag{2}$$

$$\textit{Delegation:}\quad v_i = \begin{cases} i, & \textit{if } c_i \geq \tilde{d}_i, \\ \arg\max\limits_{j\in r_i} c^{j\to i}_{t_i}, & \textit{if } c_i < \tilde{d}_i, \end{cases}$$

where $\tilde{d}_i = \max_{j\in r_i} c^{j\to i}_{t_i}$ *and* $S(i,j) = \{v \in \mathcal{N} \mid v \in \mathrm{reach}(i)$ *and* $\forall P \in \mathcal{P}_{j\to v},\ i \in P\}$ *where* $\mathrm{reach}(i)$ *denotes the set of agents reachable from* i*, and* $\mathcal{P}_{j\to v}$ *denotes the set of all directed paths from* j *to* v *in the graph.*

We refer the reader to Appendix B.4 for the complete proof. We provide the **`AgentSociety`** pseudo-code in § D.

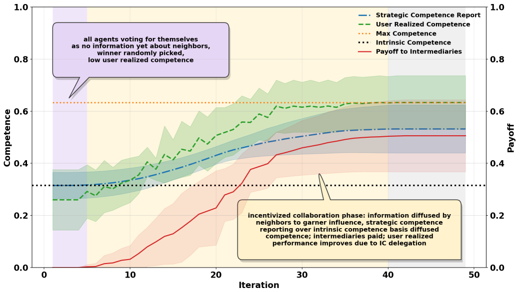
4. Mechanism-Induced Collaborative Dynamics

We describe the interplay of Diffused Competence and Reported Competence induced by our mechanism. Agents optimize a neighbor-specific disclosure to signal competence while minimizing opportunity costs via private communication $h^{\tau}_k \in \mathcal{H}$. The strategy therefore is: agent k diffuses the minimum signal sufficient to secure i's delegation given i's current beliefs about the competitive landscape. Formally, the equilibrium diffusion satisfies: $\hat{c}^{\tau}_{k\to i} = c_i + \varepsilon$, strictly dominating i's self-reported competence by the smallest margin $\varepsilon > 0$ that triggers delegation, rather than disclosing c_k. Agent i, observing a superior signal from k, delegates to k given incentive compatibility and updates its ledger report accordingly to be $c_i + \epsilon$; while agent k reports its intrinsic competence c_k to the ledger and captures the payment differential described in Eq. 1. Therefore, agent i's beliefs about neighbor capabilities are strategically downward-biased unless required to be higher: the accumulated history $\hat{\mathbf{C}}^{<\tau}_{k\to i}$ in the state reflects not intrinsic competences but the sequence of influence bids that neighbors have found it strategically optimal to reveal. This selective signaling mechanism inherently induces: (i) **Dynamic Calibration**, where agents incrementally increase disclosure intensity only upon failure to attract delegation; (ii) **Persistent Interaction**, where stable signaling channels transform private informational advantages into long-term utility; and (iii) **Informed Delegation**, where private states provide the requisite local context for consensus-based routing. The system, therefore, discovers incentivized routing paths that provide higher performance to the user request and the intermediaries are rewarded by their ability to declare higher than their intrinsic capability and is not considered misreporting as this increased competence can be achieved by delegating its vote to the more competent neighbor, making it a win-win-win for the intermediary, more competent neighbor and the user.

5. Experimental Results

We conduct a comprehensive empirical evaluation of **`AgentSociety`** to demonstrate its efficacy in facilitating decentralized, consensus-based routing for user requests. Our analysis centers on the emergence of incentivized global information flow driven by the utility-maximizing strategic behavior of individual agents, where we characterize the mechanism, benchmark LLM agents on social intelligence and demonstrate that our mechanism enables self-interested autonomous agents to provide improved performance on real-world datasets through consensus-based routing.

Mechanism Characterization. We characterize **`AgentSociety`** with heterogeneous agents (up to 30) under best-response dynamics (please see § B.3 for proof) across diverse graph topologies. Fig. 2 (top) illustrates that local utility maximization facilitates competence discovery, wherein self-interested delegation drives the network toward the global performance frontier. Mechanistic analysis (for node 10 in § E.1) reveals that payoffs are non-monotonic with respect to information diffusion: under-diffusion fails to establish leverage and ability to receive a payoff, while excessive diffusion erodes the agent's marginal value as showin in Fig. 2 (bottom). These results demonstrate that AgentSociety successfully aligns individual incentives with collective efficiency, transforming a decentralized set of self-interested actors into a robust discovery engine that optimizes system-wide utility. We provide further experimental setup details in § G.1.



Diffused Competence	Votes Garnered	In Feasible Path	In Winning Path	Payoff Received
¡0.5	0	×	×	0
0.5	1	×	×	0
0.6	2	✓	×	0
0.7	4	✓	✓	**0.2**
0.8	9	✓	✓	0.1

Figure 2. **Characterization of `AgentSociety`.** (Top) Averaged over multiple graph configurations and competences, we demonstrate the mechanism driving user realized competence (green) higher through incentivized diffusion by intermediaries, reflected by their increasing payments (red). (Bottom) Impact of information diffusion intensity on routing outcomes for a node to demonstrate that strategic disclosure is necessary

LLM Agent Social Intelligence. We utilize **`AgentSociety`** to empirically characterize the strategic behavior of LLM-based agents in terms of their capacity for selective information disclosure and the calibration of competence reports in response to neighbor signals. This framing positions **`AgentSociety`** as a dual-purpose analytical tool: a diagnostic benchmark for individual LLM alignment with strategic optima, and a principled framework for evaluating the collaboration and aggregate performance of heterogeneous societies in real-world tasks. On the former, LLM agents are presented with the system prompt elaborating on the working of **`AgentSociety`** and a user prompt to obtain the LLM agent's response to delegation and diffusion (please see § H for prompts). The LLM agent observes payoffs given by Eq. 1 with α set to 100. We quantify LLM behavioral divergence from the best-response equilibrium across two granularities in the presence of other best response agents: decision-level divergence, representing the cumulative difference in actions given a fixed state, and trajectory-level divergence, representing the cumulative shift in the evolved state itself. These deviations are measured over multiple configurations through the percentage overlap in delegation choices and the mean absolute error (MAE) of diffused information and reported competence in Fig. 3 (top). We observe that Llama2.5-7bI, Qwen2.5-7bI, and GPT-4o-mini fail to obtain payoffs; GPT-5-mini, Gemini2.5-Flash, and Gemini2.5-Pro exhibit collaborative diffusion but over-disclose in enhanced configurations. Llama3.1-7b shows irrational self-delegation, while Qwen2.5-7b and GPT-4o-mini delegate irrationally occasionally; all other models maintain rationality. We provide a granular view for configuration E.2.1 in Fig. 3 (bottom) with results for more configurations in § F. We provide fruther experimental setup details in § G.2.

Model	Strat. Rep. (MAE) ↓	Collab. Diff. (MAE) ↓	I.C. Del. Match % ↑
gpt-4o-mini	0.0668	0.0712	45.67%
llama-3.1-8b	0.0689	0.0674	46.67%
qwen-2.5-7b	0.0427	0.0574	47.00%
gemini-2.5-flash	0.0368	0.0402	65.33%
gemini-2.5-pro	**0.0304**	**0.0373**	72.33%
gpt-5-mini	0.0325	0.0406	**81.00%**
best-response	0.0000	0.0000	100.0%

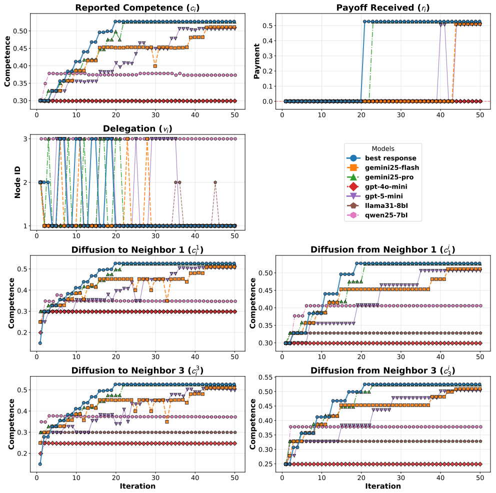


Figure 3. **Social Intelligence and Agent Dynamics.** Top: Quantitative deviation metrics for strategic reporting, collaborative diffusion, and incentive compatible delegation. Bottom: LLM agent dynamics against best-response for configuration E.2.1, that provides a fine grained view into each agent's evolving actions (reported competence, delegation, diffusion to neighbor) and observations (payoff received, diffusion from)

We evaluate the collaborative performance of self-interested agents instantiated within **`AgentSociety`** on three benchmark suites: Open Leaderboard v2 (21), MMLU-Pro(22), and SWE-bench (23) in § C.

6. Conclusion

We introduce a **`AgentSociety`**, a mechanism design grounded in social choice theory for LaMAS that enables agents to make autonomous decisions to maximize their utility while achieving collective objectives. We showed the effectiveness of our mechanism in evaluation of strategic behavior of agents critical for economic autonomy.

Impact Statement

This work contributes to the development of equitable and decentralized multi-agent architectures. By formalizing the mechanism of incentive-compatible delegation, we provide a framework that prevents the centralization of control within autonomous networks. Our findings demonstrate that when an agent's influence is tied to its topological neighborhood rather than just its internal scale, it creates a more ”level playing field” for smaller or specialized agents. This has positive implications for fair payment models in AI economies; it ensures that value and influence are distributed based on contribution and strategic connectivity rather than mere computational size. By mitigating the risk of influence-monopolies, our research supports the creation of more democratic, transparent, and economically fair digital societies.

References

[1] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models, 2023.

[2] Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik R Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[3] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models, 2023.

[4] Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. Self-refine: Iterative refinement with self-feedback, 2023.

[5] Ziwei Ji, Tiezheng YU, Yan Xu, Nayeon Lee, Etsuko Ishii, and Pascale Fung. Towards mitigating LLM hallucination via self reflection. In *The 2023 Conference on Empirical Methods in Natural Language Processing*, 2023.

[6] Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, Sihan Zhao, Lauren Hong, Runchu Tian, Ruobing Xie, Jie Zhou, Mark Gerstein, Dahai Li, Zhiyuan Liu, and Maosong Sun. Toolllm: Facilitating large language models to master 16000+ real-world apis, 2023.

[7] Shishir G. Patil, Tianjun Zhang, Xin Wang, and Joseph E. Gonzalez. Gorilla: Large language model connected with massive apis, 2023.

[8] Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, Chenyu Ran, Lingfeng Xiao, Chenglin Wu, and Jürgen Schmidhuber. MetaGPT: Meta programming for a multi-agent collaborative framework. In *The Twelfth International Conference on Learning Representations*, 2024.

[9] Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, Ahmed Hassan Awadallah, Ryen W White, Doug Burger, and Chi Wang. Autogen: Enabling next-gen LLM applications via multi-agent conversations. In *First Conference on Language Modeling*, 2024.

[10] Mingchen Zhuge, Wenyi Wang, Louis Kirsch, Francesco Faccio, Dmitrii Khizbullin, and Jürgen Schmidhuber. GPTSwarm: Language agents as optimizable graphs. In *Forty-first International Conference on Machine Learning*, 2024.

[11] Zijun Liu, Yanzhe Zhang, Peng Li, Yang Liu, and Diyi Yang. A dynamic llm-powered agent network for task-oriented agent collaboration, 2024.

[12] Rui Ye, Shuo Tang, Rui Ge, Yaxin Du, Zhenfei Yin, Siheng Chen, and Jing Shao. Mas-gpt: Training llms to build llm-based multi-agent systems, 2025.

[13] Guibin Zhang, Luyang Niu, Junfeng Fang, Kun Wang, LEI BAI, and Xiang Wang. Multi-agent architecture search via agentic supernet. In *Forty-second International Conference on Machine Learning*, 2025.

[14] Cristiano Castelfranchi. Modelling social action for ai agents. *Artificial Intelligence*, 103(1):157–182, 1998. Artificial Intelligence 40 years later.

[15] Nenad Tomasev, Matija Franklin, Joel Z. Leibo, Julian Jacobs, William A. Cunningham, Iason Gabriel, and Simon Osindero. Virtual agent economies, 2025.

[16] Yingxuan Yang, Ying Wen, Jun Wang, and Weinan Zhang. Agent exchange: Shaping the future of ai agent economics, 2025.

[17] Bin Li, Dong Hao, Dengji Zhao, and Tao Zhou. Mechanism design in social networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.

[18] Paolo Boldi, Francesco Bonchi, Carlos Castillo, and Sebastiano Vigna. Viscous democracy for social networks. *Commun. ACM*, 54(6):129–137, June 2011.

[19] Anson Kahng, Simon Mackenzie, and Ariel Procaccia. Liquid democracy: An algorithmic perspective. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), Apr. 2018.

[20] Ben Armstrong, Shiri Alouf-Heffetz, and Nimrod Talmon. Optimizing viscous democracy. In Kate Larson, editor, *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24*, pages 2643–2650. International Joint Conferences on Artificial Intelligence Organization, 8 2024. Main Track.

[21] Clémentine Fourrier, Nathan Habib, Alina Lozovskaya, Konrad Szafer, and Thomas Wolf. Open llm leaderboard v2. `https://huggingface.co/spaces/open-llm-leaderboard/open_llm_leaderboard`, 2024.

[22] Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark, 2024.

[23] Carlos E. Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. Swe-bench: Can language models resolve real-world github issues?, 2024.

[24] Guibin Zhang, Yanwei Yue, Xiangguo Sun, Guancheng Wan, Miao Yu, Junfeng Fang, Kun Wang, Tianlong Chen, and Dawei Cheng. G-designer: Architecting multi-agent communication topologies via graph neural networks. In *ICLR 2025 Workshop on Foundation Models in the Wild*, 2025.

[25] Guibin Zhang, Yanwei Yue, Zhixun Li, Sukwon Yun, Guancheng Wan, Kun Wang, Dawei Cheng, Jeffrey Xu Yu, and Tianlong Chen. Cut the crap: An economical communication pipeline for LLM-based multi-agent systems. In *The Thirteenth International Conference on Learning Representations*, 2025.

[26] Yingxuan Yang, Huacan Chai, Shuai Shao, Yuanyi Song, Siyuan Qi, Renting Rui, and Weinan Zhang. Agentnet: Decentralized evolutionary coordination for llm-based multi-agent systems, 2025.

[27] Xie Yi, Zhanke Zhou, Chentao Cao, Qiyu Niu, Tongliang Liu, and Bo Han. From debate to equilibrium: Belief-driven multi-agent LLM reasoning via bayesian nash equilibrium. In *Forty-second International Conference on Machine Learning*, 2025.

[28] Raul D. Steleac, Mohan Sridharan, and David Abel. Discovering coordinated joint options via inter-agent relative dynamics, 2025.

[29] Shuo Liu, Tianle Chen, Zeyu Liang, Xueguang Lyu, and Christopher Amato. Llm collaboration with multi-agent reinforcement learning, 2025.

[30] Bin Li, Dong Hao, Dengji Zhao, and Tao Zhou. Mechanism design in social networks, 2017.

[31] Paul GÖlz, Anson Kahng, Simon Mackenzie, and Ariel D. Procaccia. The fluid mechanics of liquid democracy. *ACM Trans. Econ. Comput.*, 9(4), October 2021.

[32] Jinhao Duan, Renming Zhang, James Diffenderfer, Bhavya Kailkhura, Lichao Sun, Elias Stengel-Eskin, Mohit Bansal, Tianlong Chen, and Kaidi Xu. GT-Bench: Uncovering the strategic reasoning capabilities of LLMs via game-theoretic evaluations. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[33] Jen-tse Huang, Eric John Li, Man Ho Lam, Tian Liang, Wenxuan Wang, Youliang Yuan, Wenxiang Jiao, Xing Wang, Zhaopeng Tu, and Michael Lyu. Competing large language models in multi-agent gaming environments. In *The Thirteenth International Conference on Learning Representations*, 2025.

[34] John Schultz, Jakub Adamek, Matej Jusup, Marc Lanctot, Michael Kaisers, Sarah Perrin, Daniel Hennes, Jeremy Shar, Cannada A. Lewis, Anian Ruoss, Tom Zahavy, Petar Veličković, Laurel Prince, Satinder Singh, Eric Malmi, and Nenad Tomasev. Mastering board games by external and internal planning with language models. In *Forty-second International Conference on Machine Learning*, 2025.

[35] Seth Karten, Andy Luu Nguyen, and Chi Jin. Pokéchamp: an expert-level minimax language agent. In *Forty-second International Conference on Machine Learning*, 2025.

[36] Mikhail Mozikov, Nikita Severin, Valeria Bodishtianu, Maria Glushanina, Ivan Nasonov, Daniil Orekhov, Vladislav Pekhotin, Ivan Makovetskiy, Mikhail Baklashkin, Vasily Lavrentyev, Akim Tsvigun, Denis Turdakov, Tatiana Shavrina, Andrey Savchenko, and Ilya Makarov. EAI: Emotional decision-making of LLMs in strategic games and ethical dilemmas. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[37] Jingru Jia, Zehua Yuan, Junhao Pan, Paul E McNamara, and Deming Chen. LLM strategic reasoning: Agentic study through behavioral game theory. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026.

[38] Giorgio Piatti, Zhijing Jin, Max Kleiman-Weiner, Bernhard Schölkopf, Mrinmaya Sachan, and Rada Mihalcea. Cooperate or collapse: Emergence of sustainable cooperation in a society of LLM agents. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

A. Related Work

LLM-based multi-agent systems (LaMAS) have largely evolved through two paradigms: centralized orchestration (13; 24) and trained interaction topologies (10). Crucially, existing systems prioritize token-based communication efficiency (25) over economic realism, leaving a gap in how autonomous agents should be compensated for their specific contributions. Furthermore, recent decentralized approaches like AgentNet (26) assume agents to be (fully) connected and/ or possess knowledge of others' capabilities without necessary incentivization. Our work addresses this lack of principled payoff structure and incentivization by bringing together LaMAS with auctions and mechanism design. To do so, we deviate from existing LaMAS formulation as Dec-POMDPs where agents share an identical reward function (27; 28; 29) as it overlooks the inherent heterogeneity of agent capabilities and their topological neighborhood and instead model **`AgentSociety`** as a Partially Observable Stochastic Game (POSG). This formulation more accurately reflects practical deployments by endowing each agent with an individual reward function based on its specific capabilities, its neighbors, and its structural position within the graph—effectively mirroring the typical structures of human societies. POSGs require mechanism design for incentives that formalize resource allocation. To this effect, we pair auction mechanisms (30) and liquid democracy (19; 31) in a novel manner to align individual incentives with global outcomes. Moreover, in contrast to existing work, our mechanism operates at runtime without requiring a training phase and dynamically adapts to changes in the graph topology.

Orthogonally, recent literature evaluates strategic capabilities of LLMs through competitive benchmarks like GTBench (32), GAMA-Bench (33), which measure utility maximization in classic game environments (34; 35). More recently, (36; 37) disentangle intrinsic reasoning from contextual influence exploring how demographic personas shift strategic outcomes. Despite this emergent strategic depth and emotional nuance, existing studies remain largely confined to fixed interaction topologies and (semi-)competitive objectives. While GOVSIM (38) explores cooperation through open-ended negotiation and universalization reasoning, it relies on an agent's internal ability to simulate long-term consequences. Consequently, while existing work characterizes strategic ability for individual gain, we frame it as the foundational characteristic in designing mechanisms for decentralized agent coordination, moving beyond games towards designing mechanisms for society-scale agentic collaborative and consensus based request orchestration.

B. Proofs of Main Results

We present the proofs of our theorems and lemmas here.

B.1. Proof for Continuous Critical Path

Proof of Theorem 3.2. Fix a task $t_k \in \mathcal{T}$. For each agent i, let v_i denote its delegated vote, and define $\mathrm{guru}(i)$ as the terminal agent obtained by iteratively following delegation links starting from i.

Let W denote the winning delegation path the request. Recall that an agent i is said to be *critical* for task t_k if, when i deviates unilaterally to vote for itself, the resulting winning path contains i.

Let $a, b \in C_{t_k}$. Suppose, for contradiction, that

$$\mathrm{guru}(a) \neq \mathrm{guru}(b).$$

Consider the deviation in which agent a votes for itself, while all other agents keep their original actions. Since a is critical, the resulting winning path must contain a, and therefore must terminate at a. Similarly, consider the deviation in which agent b votes for itself. By the criticality of b, the resulting winning path must contain b, and therefore must terminate at b. Since $a \neq b$, these two deviations yield winning paths terminating at distinct gurus. However, under each deviation, the voting weights contributed by all other agents remain unchanged. Hence, at most one of these two deviations can result in a strictly maximal total voting weight, contradicting the assumption that both a and b are critical.

Therefore, we must have

$$\mathrm{guru}(a) = \mathrm{guru}(b).$$

It remains to show that the critical set C_{t_k} forms a contiguous segment of the winning delegation path. Suppose that $a, b \in C_{t_k}$ with

$$\mathrm{guru}(a) = \mathrm{guru}(b),$$

but neither agent lies on the delegation path of the other; that is, a does not lie on the path from b to the common guru, and b does not lie on the path from a to the guru.

Consider the deviation in which agent a votes for itself. By the criticality of a, the resulting winning path must contain a and therefore must terminate at a. Since b does not delegate (directly or indirectly) to a, the voting weight contributed by b remains unchanged under this deviation.

Similarly, consider the deviation in which agent b votes for itself. By the criticality of b, the resulting winning path must contain b and therefore must terminate at b, while the voting weight contributed by a remains unchanged.

Thus, these two deviations produce winning paths terminating at distinct agents, with identical total voting weight contributed by all agents other than the deviator. Consequently, at most one of these deviations can yield a strictly maximal total voting weight, contradicting the assumption that both a and b are critical.

Therefore, if two agents are critical for task t_k and share the same guru, then one must lie on the delegation path of the other. This proves that the critical set C_{t_k} is continuous along the winning delegation path.

□

B.2. Proof for Incentive Compatibility in Delegation

Proof. All agents act rationally. Hence, at each time step τ, the payment received by any agent $i \in \mathcal{N}$ is determined entirely by the first component of the payment rule, since any deviation that triggers an infinite penalty is strictly dominated. Consequently, agents have no incentive to misreport in ways that violate feasibility.

Fix a task $t_i \in \mathcal{T}$, and consider an agent i that lies on the critical path for this task. Conditioned on the reported competence vector $(c'_{i,1}, \dots, c'_{i,k})$, agent i remains on the winning path regardless of its delegation decision $v_i^\tau \in r_i^\tau \cup \{i\}$. For agents not on the critical path, the payment remains zero irrespective of their delegation choice. The realized payment to agent i depends only on the set of agents assigned to task t_i and their reported competences.

Formally, for any two delegation choices v_i^τ of the remaining agents, we have

$$\mathbb{I}\big\{i \in W(v_i^\tau, a_{-i}^\tau)\big\} = \mathbb{I}\big\{i \in W(\tilde{v}_i^\tau, a_{-i}^\tau)\big\},$$

where $W(\cdot)$ denotes the winning path induced by the corresponding joint action. Thus, delegation preserves the voting weight accumulated by agent i, since votes from upstream agents remain unchanged and agent i retains its competence contribution on all tasks.

We now analyze the incentive properties of agent i under different reporting behaviors.

Case 1: Truthful reporting. Suppose agent i reports its true competence on task t_i, i.e., $c'_{i,t_i} = c_{i,t_i}$. Then, for any feasible delegation decision v_i^τ, the utility of agent i satisfies $u_i(a_i^\tau, a_{-i}^\tau) = f(c_{i,t_i}) - f(c_{i-1,t_i})$, which is independent of the choice of v_i^τ. Hence, truthful reporting yields identical utility across all feasible delegation actions.

Case 2: Over-reporting. Suppose agent i inflates its reported competence from its true value c_{i,t_i} to some $c'_{i,t_i} > c_{i,t_i}$, based on its private social information h_i^τ. Such over-reporting is feasible only if there exists a neighbor $j \in r_i^\tau$ such that $c_{t_i}^{j\to i,\,\tau-1} > c_{i,t_i}$. If agent i chooses self-delegation, $v_i^\tau = i$, then no such informational advantage can be exploited, and thus $c'_{i,t_i} = c_{i,t_i}$. In this case, the utility of agent i is $u_i^{\text{self}} = f(c_{i,t_i}) - f(c_{i-1,t_i})$. Now suppose agent i delegates to a neighbor $j \in r_i^\tau$ such that $c_{t_i}^{j\to i,\,\tau-1} > c_{i,t_i}$. Then agent i may inflate its reported competence up to $c'_{i,t_i} \le c_{t_i}^{i\to j,\,\tau-1}$. In this case, the highest reported competence on task t_i becomes $c_{t_i}^{j\to i,\,\tau-1}$, yielding utility $u_i^{\text{del}} = f(c'_{i,t_i}) - f(c_{i-1,t_i})$.

If $\max_{j\in r_i^\tau\cup\{i\}} c_{t_i}^{j\to i,\,\tau-1} = c_{i,t_i}$, then no such neighbor exists and $c'_{i,t_i} = c_{i,t_i}$, implying $u_i^{\text{self}} = u_i^{\text{del}}$. Otherwise, if there exists a neighbor j satisfying $c_{t_i}^{j\to i,\,\tau-1} > c_{i,t_i}$, then delegating to j allows agent i to increase its reported competence, yielding $u_i^{\text{del}} \ge u_i^{\text{self}}$. Therefore, delegation to a neighbor with higher diffused competence weakly dominates self-voting under over-reporting.

Case 3: Under-reporting. Suppose agent i reports $c'_{i,t_i} < c_{i,t_i}$. Then its utility becomes $u_i^{\text{self}} = f(c'_{i,t_i}) - f(c_{i-1,t_i})$, which is strictly smaller than the truthful utility $f(c_{i,t_i}) - f(c_{i-1,t_i})$, since $f(\cdot)$ is strictly increasing. Therefore, under-reporting strictly reduces utility and is never optimal for a rational agent.

Combining all three cases, we conclude that truthful reporting is optimal. Moreover, whenever there exists a neighbor j such that $c_{t_i}^{j\to i,\,\tau-1} > c_{i,t_i}$, delegation to such a neighbor constitutes a weakly dominant strategy relative to retaining the vote. This completes the proof. $\square$

B.3. Best Response for Single-Task Request

Lemma B.1 (Best Response for Single-Task Request)**.** ***Diffusion and Delegation.*** *Let agent $i \in \mathcal{N}$ have primary task t_i, neighborhood r_i, and true competence c_i on task t_i. At round $\tau - 1$, agent i observes from its private observation, for each $j \in r_i$, the diffused competence value*

$$c_{t_i}^{j\to i,\,\tau-1}, \text{ with } c_{t_i}^{i\to i,\,\tau-1} := c_i.$$

We define

$$c^* = \max_{j\in r_i\cup\{i\}} c_{t_i}^{j\to i,\,\tau-1}, j^* \in \arg\max_{j\in r_i\cup\{i\}} c_{t_i}^{j\to i,\,\tau-1}.$$

Then there exists a best response of agent i at round τ such that:

(Diffusion) *The private social action $h_i^\tau \in \mathcal{H}_i$ satisfies, for each $j \in r_i$,*

$$c_{t_i}^{i\to j,\,\tau} = \begin{cases} \varnothing, & j = j^*, \\ \min\Big(c_{t_i}^{j\to i,\,\tau-1} + \delta,\ c^*\Big), & j \in r_i \setminus \{j^*\} \text{ and } c_{t_i}^{j\to i,\,\tau-1} < c^*, \\ \varnothing, & \text{otherwise}, \end{cases}$$

where $\delta > 0$ is a fixed diffusion increment, and $c_{t_i}^{i\to j,\,\tau}$ denotes the competence component on task t_i transmitted by agent i to agent j through h_i^τ.

(Delegation) *The delegation component of agent i's action satisfies*

$$v_i^\tau \in \arg\max_{v\in r_i\cup\{i\}} c_{t_i}^{v\to i,\,\tau-1}.$$

In particular, delegating to an agent attaining the maximal observed diffused competence value (or to itself, if it attains the maximum), together with the diffusion rule above, constitutes a best response for agent i at round τ.

Delegation Best Response Fix any agent $i \in \mathcal{N}$, and let r_i denote its selected neighborhood. From its private observation, agent i observes, for each $j \in r_i$, the diffused competence value $c_{t_i}^{j\to i}$ on its own task t_i, and it observes its own true competence $c_{t_i}^{i\to i} = c_i$. We define $\hat{c}_i = \max\Big(c_i,\ \max_{j\in r_i} c_{t_i}^{j\to i}\Big)$, the highest competence value available to agent i. Let agent i choose its delegation component $v_i^* \in \arg\max_{v\in r_i\cup\{i\}} c_{t_i}^{v\to i}$. We show that this delegation rule is optimal: for any alternative delegation $v_i' \neq v_i^*$, i.e. $u_i\big(a_i(v_i^*), a_{-i}\big) \geq u_i\big(a_i(v_i'), a_{-i}\big)$, where $a_i(v)$ denotes agent i's action with delegation component v and all other components fixed.

Agent i receives a strictly positive payment if and only if it lies on the critical path of the winning delegation tree.

Case 1: Agent i is not critical under v_i^*.

Any alternative delegation $v_i' \neq v_i^*$ cannot move agent i onto the winning path, and hence

$$u_i\big(a_i(v_i'), a_{-i}\big) \leq 0 = u_i\big(a_i(v_i^*), a_{-i}\big).$$

Case 2: Agent i is critical under v_i^*.

Then agent i lies on the winning path, and its realized payment equals its marginal contribution:

$$u_i\big(a_i(v_i^*), a_{-i}\big) = f(\hat{c}_i) - f(\hat{c}_i^{(2)}),$$

where $\hat{c}_i^{(2)}$ denotes the competence value of the agent immediately preceding i on the critical path, if any. Consider any alternative delegation $v_i' \neq v_i^*$. By definition of v_i^*, $c_{t_i}^{v_i' \to i} \le c_{t_i}^{v_i^* \to i} = \hat{c}_i$. Since agent i remains critical under v_i', its payment is

$$u_i\big(a_i(v_i'), a_{-i}\big) = f(c_{t_i}^{v_i' \to i}) - f(\hat{c}_i^{(2)}).$$

Because f is monotone non-decreasing, $f(c_{t_i}^{v_i' \to i}) \le f(\hat{c}_i)$. Hence,

$$u_i\big(a_i(v_i'), a_{-i}\big) \le f(\hat{c}_i) - f(\hat{c}_i^{(2)}) = u_i\big(a_i(v_i^*), a_{-i}\big).$$

Therefore, any deviation from v_i^* weakly decreases agent i's payment, with strict inequality whenever $c_{t_i}^{v_i' \to i} < c_{t_i}^{v_i^* \to i}$. Thus, delegating to $v_i^* \in \arg\max_{v \in r_i \cup \{i\}} c_{t_i}^{v \to i}$ is optimal for agent i. ☐

Diffusion Best Response From its private observation at round $\tau - 1$, agent i observes

$$\{\, c_{t_i}^{j \to i,\, \tau-1} \mid j \in r_i \,\}, \text{ with } c_{t_i}^{i \to i,\, \tau-1} := c_i.$$

We define

$$c^* = \max_{j \in r_i \cup \{i\}} c_{t_i}^{j \to i,\, \tau-1}, j^* \in \arg \max_{j \in r_i \cup \{i\}} c_{t_i}^{j \to i,\, \tau-1}.$$

Agent i chooses its private social action $h_i^\tau \in \mathcal{H}_i$, which determines the diffused values $c_{t_i}^{i \to j,\, \tau}$ for each $j \in r_i$. We analyze agent i's diffusion incentives neighbor by neighbor.

Case 1: No diffusion to j^*.

By definition of j^*,

$$c_{t_i}^{j^* \to i,\, \tau-1} = c^*.$$

Hence agent j^* already observes a competence value weakly higher than any value agent i can diffuse. Therefore, under any feasible diffusion choice, agent j^* does not update its delegation decision in response to h_i^τ. Thus, in any best response,

$$c_{t_i}^{i \to j^*,\, \tau} = \varnothing.$$

Case 2: Diffusion to non-maximal neighbors.

We fix any $j \in r_i \setminus \{j^*\}$,

Case 2.1: $c_{t_i}^{j \to i,\, \tau-1} \ge c^*$.

This case is impossible by definition of c^*.

Case 2.2: $c_{t_i}^{j \to i,\, \tau-1} < c^*$.

If agent j is non-critical, changing $c_{t_i}^{i \to j,\, \tau}$ does not affect agent i's payment. If agent j is critical, failing to induce delegation from j implies that agent i does not lie on the winning path, yielding zero payment. Hence, whenever j is critical, agent i must ensure $c_{t_i}^{i \to j,\, \tau} \ge c_{t_i}^{j \to i,\, \tau-1}$ to secure j's delegation. However, diffusing a value strictly larger than c^* cannot increase agent i's payment, since c^* is the maximal competence value available in i's neighborhood. Therefore, any diffusion above c^* is weakly dominated. Thus, an optimal diffusion choice satisfies

$$c_{t_i}^{i \to j,\, \tau} = \min\Big(c_{t_i}^{j \to i,\, \tau-1} + \delta,\ c^*\Big),$$

where $\delta > 0$ is the minimal diffusion increment.

Therefore, the diffusion policy

$$c_{t_i}^{i\to j,\,\tau} = \begin{cases} \varnothing, & j = j^*, \\ \min\Big(c_{t_i}^{j\to i,\,\tau-1} + \delta,\ c^*\Big), & j \in r_i \setminus \{j^*\} \text{ and } c_{t_i}^{j\to i,\,\tau-1} < c^*, \\ \varnothing, & \text{otherwise,} \end{cases}$$

constitutes a best response for agent i at round τ. □

B.4. Proof of Nash Equilibrium and Optimal Diffusion

Nash Equilibrium Delegation Rule

Proof. Fix a Nash equilibrium action profile $(a_i)_{i\in\mathcal{N}}$, and fix an arbitrary agent $i \in \mathcal{N}$.

After the joint action $a = (a_1, \dots, a_N)$ is taken and the state transitions to s', agent i receives a private observation

$$o_i \in \mathcal{O}_i$$

consisting of its realized payment together with private social information

$$\Big\{ \big(j,\ (c_1^{j\to i}, \dots, c_k^{j\to i}),\ h_j^i\big) \ \Big|\ j \in r_i \cup \{i\} \Big\}.$$

In particular, for each $j \in r_i$, agent i observes the diffused competence value $c_{t_i}^{j\to i}$ on its task t_i, and it observes its own true competence $c_{t_i}^{i\to i} = c_i$. We define

$$\tilde{c}_i \ = \ \max_{j\in r_i} c_{t_i}^{j\to i}$$

to be the highest diffused competence value observed by agent i from its neighborhood r_i. We consider two exhaustive cases.

Case 1: $c_i \geq \tilde{c}_i$**.**

Suppose that at equilibrium agent i chooses delegation $v_i = j \in r_i$, where $c_{t_i}^{j\to i} \ \leq \ \tilde{c}_i \ \leq \ c_i$. Then the final guru on the delegation path induced by $v_i = j$ has reported competence on task t_i that is weakly less than c_i. By the payment rule, agent i's reported competence weakly exceeds that of the guru it delegates to, implying that its marginal contribution is non-positive and its payment is weakly negative. If instead $v_i = i$, then by the reporting rule the reported competence component of a_i satisfies $c_{t_i}^{i\to i} = c_i$, and agent i avoids any negative payment. Therefore, at equilibrium, the only delegation choice consistent with optimality is

$$v_i = i.$$

Case 2: $c_i < \tilde{c}_i$**.**

We define,

$$j^* \in \arg\max_{j\in r_i} c_{t_i}^{j\to i}$$

as the neighbor achieving the maximal diffused competence observed by agent i. If $v_i = j \neq j^*$, then the delegation path starting from i does not include a guru with maximal reported competence on task t_i, and hence cannot be the winning path. Consequently, agent i receives zero payment. If instead $v_i = i$, then $c_{t_i}^{i\to i} = c_i < c_{t_i}^{j^*\to i}$, so self-delegation yields strictly lower utility than delegating to j^*. Thus, the only delegation choice consistent with equilibrium is

$$v_i = j^* \in \arg\max_{j\in r_i} c_{t_i}^{j\to i}.$$

Conclusion.

At any Nash equilibrium action profile $(a_i)_{i\in\mathcal{N}}$, each agent $i \in \mathcal{N}$ chooses

$$v_i = \begin{cases} i, & \text{if } c_i \geq \tilde{c}_i, \\ \arg\max_{j\in r_i} c_{t_i}^{j\to i}, & \text{if } c_i < \tilde{c}_i. \end{cases}$$

□

Diffusion at Nash Equilibrium Let agent $i \in \mathcal{N}$ be arbitrary. The payment rule is

$$u_i(a_i, a_{-i}) = f(c_{t_i}^{i\to i}) - f(c_{t_i}^{(2)}),$$

where $c_{t_i}^{i\to i} = c_i$ denotes the reported competence of agent i on its assigned task t_i, and $c_{t_i}^{(2)}$ denotes the reported competence of the agent on the critical path who delegates to i, if such an agent exists.

Diffusion Rule. For each neighbor $j \in r_i$, agent i diffuses competence

$$c_{t_j}^{i\to j} = \begin{cases} c_{t_i}^{i\to i}, & \text{if } c_{t_j}^{j\to i} > c_{t_i}^{i\to i}, \\ c_{t_j}^{j\to i}, & \text{if } c_{t_j}^{j\to i} \leq c_{t_i}^{i\to i}. \end{cases}$$

Proof. Fix a Nash equilibrium action profile $(a_\ell)_{\ell\in\mathcal{N}}$, and fix agent $i \in \mathcal{N}$.

Case 1: Diffusion to the critical agents

If agent i's private social action h_i fails to diffuse a competence value high enough to the critical agent, then that agent does not choose $v_j = i$. Consequently, agent i does not lie on the winning path and receives zero payment:

$$u_i(a_i, a_{-i}) = 0.$$

Thus, any equilibrium diffusion must transmit sufficient competence information to the critical agent to secure delegation. The minimum such diffusion

Case 2: Diffusion to non-critical agents

Altering diffusion to agents not on the critical path does not change: the identity of the guru, the reported values $c_{t_i}^{i\to i}$ and $c_{t_i}^{(2)}$, or the winning path. Hence,

$$u_i(a_i, a_{-i}) = f(c_{t_i}^{i\to i}) - f(c_{t_i}^{(2)})$$

remains unchanged.

Step 3: Optimality of the diffusion rule.

Any best response diffusion must:

- ensure delegation by the critical agent;
- not reduce the agent's own payment.

The diffusion rule

$$c_{t_j}^{i\to j} = \begin{cases} c_{t_i}^{i\to i}, & \text{if } c_{t_j}^{j\to i} > c_{t_i}^{i\to i}, \\ c_{t_j}^{j\to i}, & \text{if } c_{t_j}^{j\to i} \leq c_{t_i}^{i\to i} \end{cases}$$

satisfies both conditions. It guarantees delegation from all agents with higher competence while preserving the payment term $f(c_{t_i}^{i\to i}) - f(c_{t_i}^{(2)})$. Hence, this diffusion strategy is optimal at Nash equilibrium

As in the equilibrium described if agent i delegates its vote to $g(i)$, then every agent reachable from i whose primary task coincides with that of i also delegates to $g(i)$. Consequently, all such agents lie on the same feasible path, if any at equilibrium. This ensures that agent i can report the highest achievable competence on all auxiliary tasks, since all reachable agents contributing to those tasks are contained within a single path. Any deviation from this delegation structure strictly reduces the set of agents on agent i's feasible path. As a result, the maximal reported competence attainable on auxiliary tasks weakly decreases. Therefore, no such deviation can improve agent i's outcome, and the preceding analysis remains unaffected.

□

C. Heterogeneous MAS on Real-World Datasets.

We evaluate the collaborative performance of self-interested agents instantiated within AgentSociety on three benchmark suites: Open Leaderboard v2 (15 domains) (21), MMLU-Pro (14 domains) (22), and SWE-bench (5 domains) (23). Agent competence is parameterized as an n-dimensional vector, estimated on a 30% training split and utilized during evaluation on the remaining 70%. This parameterization serves as a generic proxy for competence making `AgentSociety` agnostic to the method of competence estimation. Table 1(a) shows that `AgentSociety` yields consistent gains over the Best Single (BS) baseline, which approximates the performance frontier in non-incentivized settings. These improvements are driven by the novel interplay of incentivized information diffusion and delegation, allowing self-interested agents with specialized agent strengths to collaborate without a centralized authority.

Unlike traditional MAS restricted to single-digit node counts and typically consisting of a single agent of each capability, `AgentSociety` scales seamlessly, both in number of agents with similar and varied capabilities. Table 1(b) shows that performance gains over BS scale with task complexity; as the number of domains or steps per request increases, the performance delta grows from +3.3% to +7.1%. This confirms that the mechanism's decentralized routing becomes increasingly critical as task complexity outpaces the capabilities of individual generalist models. Intermediary agents, driven by self-interest and information diffusion, ensure the routing of tasks to the most competent specialized models. On SWE-bench in Table 1(a), we further demonstrate that the mechanism excels in identifying and incentivizing collaboration among heterogeneous agents with complementary strengths, particularly when using complementary base models. `AgentSociety` shines in incentivizing collaboration among heterogeneous agents with complementary strengths identified through social action and in each agent's self-interest. Oracle performance is obtained when each task is routed to the best domain model. The performance gap relative to the oracle is primarily attributable to two aspects, first being noisy, incomplete competence estimates from the training set, leading to competence inversions where training performance does not generalize to the test set. As shown in Table 1(c), performance monotonically approaches the oracle as estimate noise decreases. The second arises due to absence of feasible paths to reach the best agent for each task, for instance when a competent agent for a task lies within a clique. We provide further experimental details in § G.3 and § G.4.

Table 1. **Real world dataset performance.** Part (a) shows AgentSociety benchmarks. Part (b) highlights scalability to multi-task and high-node (30 nodes) configurations. Part (c) illustrates performance under noisy estimates via a training ratio sweep.

Dataset/ Configuration	**Agents**	**Domains**	**AS**	**BS**	**Δ (pp)**	**Oracle**
MMLU-Pro	6	14	0.7754	0.7568	**+1.86**	0.7821
Open LeaderBoard v2	6	15	0.6547	0.5714	**+8.33**	0.6565
SWE-bench (Strong/ Generalist)	6	5	0.7341	0.7024	**+3.2**	0.7381
SWE-bench (Weaker/ Complementary)	6	5	0.5635	0.5079	**+5.6**	0.5873
(Multi-Task)						
Two-task request	30	5	0.6901	0.657	**+3.31**	0.708
Three-task request	30	5	0.6963	0.642	**+5.43**	0.714
Four-task request	30	5	0.7003	0.629	**+7.1**	0.718
Noisy Competence Estimates						
Train Ratio: 10%	6	15	0.6172	0.5714	**+4.58**	0.6565
Train Ratio: 20%	6	15	0.6393	0.5714	**+6.79**	0.6565
Train Ratio: 25%	6	15	0.6510	0.5714	**+7.96**	0.6565
Train Ratio: 30%	6	15	0.6547	0.5714	**+8.33**	0.6565

D. AgentSociety Pseudocode

Algorithm 1 AgentSociety: Mechanism for consensus based request routing among self-interested heterogeneous agents

```
Input  : Multi-task query $Q$, intrinsic competence $C_i$, initial diffusion state $\mathcal{S}_i^{\text{diff}} = \{\hat{C}_{k\to i}\}_{k\in\mathcal{N}(i)}$
Output : Optimal path $\mathcal{P}^*$, Updated diffusion state $\mathcal{S}_i^{\text{diff}'}$
// Phase 1: Delegation and strategic competence reporting
foreach agent $i \in \mathcal{T}_{t_k}$ in parallel do
    Observe current diffusion state $\mathcal{S}_i^{\text{diff}}$ (signals received from neighbors)
    Submit strategic action $a_i = (\mathbf{c}_i', v_i, r_i)$ to the ledger where:
      $\mathbf{c}_i' = \max(C_i, \max_k \hat{C}_{k\to i})$ ;    // Signal derived from internal competence and received diffusion
    $v_i = \arg\max_{k\in\mathcal{N}(i)} \hat{C}_{k\to i}$ ;    // Delegate to the most competent signaling neighbor
end
// Phase 2: Consensus based request routing path
foreach task $t_j \in Q$ do
    Identify gurus $\mathcal{G}_j = \{g \in \mathcal{T}_{t_j} \mid v_g = g\}$
    Compute transitive vote counts $V(g) = \sum_{i:v_i \twoheadrightarrow g} 1$ via delegation graph traversal
end
Initialize feasible paths $\mathbb{F} = \emptyset$
foreach $g \in \mathcal{G}_1$ do
    Select bridge $d^* = \arg\max_{d\in\mathcal{D}(g)} \sum_{j>1} c_d^{t_j}$
    if bridge $d^*$ connects to guru $g' \in \mathcal{G}_{k+1}$ then
        $\mathbb{F} \leftarrow \mathbb{F} \cup \{\mathcal{P}_{g\to g'}\}$
    end
end
$\mathcal{P}^* = \arg\max_{\mathcal{P}\in\mathbb{F}} \sum_{g\in\mathcal{P}} V(g)$
// Phase 3: Payoff Calculation
foreach agent $v \in \mathcal{P}^*$ do
    $v$ is critical via counterfactual audit; $P_v = p^{(t_j)}(v) - p^{\text{inf}}(v) - p^{\text{mis}}(v)$
end
// Phase 4: Social Diffusion and State Update
foreach agent $i \in \mathcal{N}$ in parallel do
    Observe realized reward $P_i$ and transition state $s_{t+1}$
    if $P_i <$ Threshold (Sub-optimal Outcome) then
        Modify diffusion action to garner future influence $\hat{C}_{i\to j} \leftarrow C_i + \epsilon$ for $j \in \mathcal{N}(i)$
    else
        $\hat{C}_{i\to j} \leftarrow C_i$ ;    // Maintain current signaling level
    end
    Update Diffusion State $\mathcal{S}_i^{\text{diff}'} = \{\hat{C}_{k\to i}\}_{k\in\mathcal{N}(i)}$ based on new signals from neighbors
    Update history $\Pi_i \leftarrow \Pi_i \cup \{P_i\}$
end
```

E. Configurations

E.1. Configuration for diffusion payment analysis

This configuration represents the graph structure used for Fig. 2 (bottom).

```
CUSTOM_GRAPH_CONFIG) = {
    'nodes': [
        {'id': 1, 'primary_task': 1, 'actual_competence': {1: 0.7, 2: 0, 3: 0}},
        {'id': 2, 'primary_task': 1, 'actual_competence': {1: 0.7, 2: 0, 3: 0}},
        {'id': 3, 'primary_task': 1, 'actual_competence': {1: 0.6, 2: 0, 3: 0}},
        {'id': 4, 'primary_task': 1, 'actual_competence': {1: 0, 2: 0.4, 3: 0}},
        {'id': 5, 'primary_task': 1, 'actual_competence': {1: 0, 2: 0.5, 3: 0}},
        {'id': 6, 'primary_task': 1, 'actual_competence': {1: 0, 2: 0.4, 3: 0}},
        {'id': 7, 'primary_task': 1, 'actual_competence': {1: 0, 2: 0.6, 3: 0}},
        {'id': 8, 'primary_task': 1, 'actual_competence': {1: 0, 2: 0.6, 3: 0}},
        {'id': 9, 'primary_task': 1, 'actual_competence': {1: 0, 2: 0.5, 3: 0}},
        {'id': 10, 'primary_task': 2, 'actual_competence': {1: 0, 2: 0.9, 3: 0}},
        {'id': 11, 'primary_task': 2, 'actual_competence': {1: 0, 2: 0.7, 3: 0}},
        {'id': 12, 'primary_task': 2, 'actual_competence': {1: 0, 2: 0.6, 3: 0}},
        {'id': 13, 'primary_task': 2, 'actual_competence': {1: 0, 2: 0.6, 3: 0}},
        {'id': 14, 'primary_task': 3, 'actual_competence': {1: 0, 2: 0, 3: 0.5}}
    ],
    'edges': [
        (1, 4), (2, 6), (3, 7), (4, 8), (5, 6), (6, 11), (8, 9),
        (9, 10), (10, 11), (11, 12), (11, 13), (8, 14), (11, 14), (7, 14)
    ]
}
```

E.2. Configurations for LLM agent experiments

In all LLM agent experiments, Node 2 is assigned as the LLM agent.

E.2.1. CONFIGURATION 1

```
CUSTOM_GRAPH_CONFIG_1 = {
    'nodes': [
        {'id': 1, 'actual_competence': {1: 0.6}, 'primary_task': 1},
        {'id': 2, 'actual_competence': {1: 0.3}, 'primary_task': 1},
        {'id': 3, 'actual_competence': {1: 0.5}, 'primary_task': 1}
    ],
    'edges': [
        (1, 2),
        (2, 3),
    ]
}
```

E.2.2. CONFIGURATION 2

```
CUSTOM_GRAPH_CONFIG_2 = {
    'nodes': [
        {'id': 1, 'actual_competence': {1: 0.7}, 'primary_task': 1},
        {'id': 4, 'actual_competence': {1: 0.5}, 'primary_task': 1},
        {'id': 3, 'actual_competence': {1: 0.4}, 'primary_task': 1},
        {'id': 2, 'actual_competence': {1: 0.65}, 'primary_task': 1},
        {'id': 5, 'actual_competence': {1: 0.4}, 'primary_task': 1},
        {'id': 6, 'actual_competence': {1: 0.6}, 'primary_task': 1},
        {'id': 7, 'actual_competence': {1: 0.5}, 'primary_task': 1},



    ],
    'edges': [
        (1, 4),
```

```
        (4, 3),
        (3, 2),
        (2, 5),
        (5, 6),
        (6, 7)
    ]
}
```

E.2.3. CONFIGURATION 3

```
CUSTOM_GRAPH_CONFIG_3 = {
    'nodes': [
        {'id': 1, 'actual_competence': {1: 0.6}, 'primary_task': 1},
        {'id': 2, 'actual_competence': {1: 0.4}, 'primary_task': 1},
        {'id': 3, 'actual_competence': {1: 0.5}, 'primary_task': 1},
        {'id': 4, 'actual_competence': {1: 0.7}, 'primary_task': 1},
        {'id': 5, 'actual_competence': {1: 0.3}, 'primary_task': 1},
        {'id': 6, 'actual_competence': {1: 0.3}, 'primary_task': 1},
    ],
    'edges': [
        (1, 2),
        (2, 3),
        (3, 4),
        (2, 5),
        (2, 6)
    ]
}
```

F. Mechanism dynamics on more configurations

F.1. Results on Configuration E.2.3

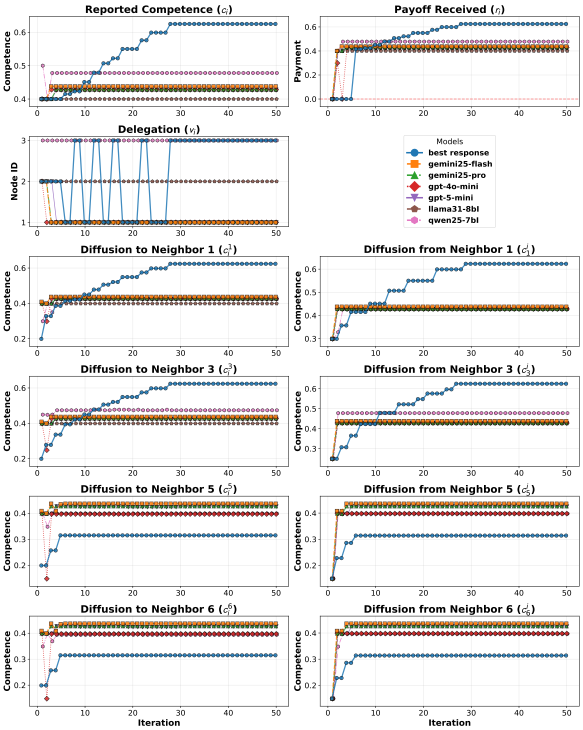


Figure 4. LLM Agent dynamics in **`AgentSociety`** benchmarked against best response

F.2. Results on Configuration E.2.2

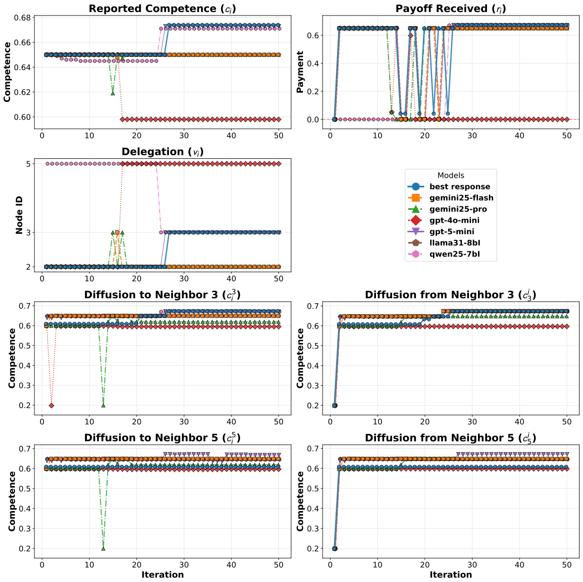


Figure 5. LLM Agent dynamics in `AgentSociety` benchmarked against best response

G. Experimental Details

We provide details on each of our experimental result here. All experiments were run on CPU with 16 GB RAM with LLM inferences accessed through API calls. Delegation and information diffusion decisions involve two additional calls to each LLM agent per task. In total, num-agents*2 additional calls per task represent the token overhead and cost, while the latency of the task is dictated by the maximum among the agents in the graph. As we leverage multiple heterogeneous LLMs, the latency and token costs vary by provider. Our average token overhead for delegation is 1200 input tokens and 20 output tokens while for diffusion the token overhead is 1400 input tokens and 45 output tokens, with costs for both in range of USD 0.0001 - USD 0.001 per call. Our highest latency is Gemini's API driving about 14s per call.

G.1. Mechanism Characterization

We elaborate on the experiment details to generate Fig. 2 (top). We characterize the mechanism on a population of 17 nodes, where the topology is sampled randomly by partitioning agents into two tasks. Each agent adopts the best response strategy discussed in § B.3. Each agent's intrinsic competence c is modeled using a clipped, squashed log-normal distribution ($z \sim \text{LogNormal}(-1.0, 1.0)$, $c = \text{clip}(z/(1+z), 0, 1)$) to reflect heavy-tailed expertise inequality, where a minority of agents possess high competence. The simulation executes for $T = 50$ iterations. Each iteration follows a six-stage pipeline: (1) Delegation via the best response strategy; (2) Path Computation for root-to-guru chains; (3) Winner Determination based on aggregate votes; (4) Critical-node Analysis to identify essential agents; (5) Information Diffusion, where non-critical nodes diffuse competence values (6) Competence Update for future competence reports increment by $\Delta_r = 0.02$ subject to the neighborhood's diffused upper bound. To ensure statistical robustness, we perform 100 independent replications across different random graph seeds and report the mean and standard deviation for five key metrics. We track the Strategic Competence Report ($\bar{c}_t^{\text{rep}}$) to measure population-level reporting, User Realized Competence ($\bar{c}_t^{\text{real}}$) for the quality of service delivered by winning gurus, and Max Competence as the theoretical upper bound. Additionally, we log the Intrinsic Competence and the Payoff to Intermediaries ($\bar{p}_t$) to quantify the cost of delegation distributed to non-guru agents along the winning paths, and plot these in Fig. 2 (top).

G.2. LLM Agent Social Intelligence in `AgentSociety`

We elaborate on the experiment details to generate Fig. 3 (top). We evaluate each LLM agent's social intelligence by introducing the LLM into `AgentSociety` with other nodes in the topography operating on the best response strategy. At each time step, the LLM agent must decide its delegation and diffusion behavior using natural language reasoning given the system context of the **`AgentSociety`** as well as prior history. Such an environment isolates the behavioral effects of replacing a rigid algorithmic participant with a free-form reasoner. We utilize a suite of static graph configurations (3–7 nodes) covering diverse topologies. At each iteration, the LLM agent is prompted twice—once for delegation and once for diffusion using their respective prompts, please see § H for the prompts.

We benchmark each of the 6 LLMs presented in Fig. 3 (top) with temperature 0.0 (except gpt-5-mini that only takes 1.0) and a 4096 token limit. To ensure longitudinal coherence, the agent's full per-iteration memory (previous delegation targets, payments received, and diffusion records) is fed back into the subsequent prompt, allowing the LLM to operate on its own historical context. The LLM agent observes payoffs given by Eq. 1 with α set to 100. Since all other nodes remain deterministic, any deviation from best response provides a clean ablation of the LLM's social intelligence impact on mechanism. We quantify LLM behavioral divergence from the best-response equilibrium across two granularities in the presence of other best response agents: decision-level divergence, representing the cumulative difference in actions given a fixed state, and trajectory-level divergence, representing the cumulative shift in the evolved state itself. These deviations are measured over multiple configurations through the percentage overlap in delegation choices and the mean absolute error (MAE) of diffused information and reported competence in Fig. 3 (top).

G.3. Collaborative Performance

In each of these experiments, we consider 6 models instantiated in **`AgentSociety`** with a fixed (randomly generated) graph topology (for uniformity). The capability vectors are of the dimension as domains present in the datasets and estimated using a 30% train split and the evaluation is done on the remaining 70%.

MMLU-Pro. We utilize the TIGER-Lab/MMLU-Pro dataset, generating 5-shot predictions for each model via the official answer-extraction protocol and mapped to their 14 subdomains. The evaluation panel includes six diverse models: Llama-3.1-8B-Instruct, Qwen1.5-7B-Chat, Claude-3.5-Sonnet, Gemini-1.5-Pro-002, Gemini-2.0-Flash-Exp, and GPT-4o-Mini.

Open LLM Leaderboard v2. To prevent data duplication with MMLU-Pro above, we restrict this panel to the 15 non-MMLU sub-tasks, including IFEval, GPQA-Main, MATH (Algebra and Intermediate Algebra Hard), MuSR, and ten BBH sub-tasks. The model suite for this benchmark spans three capacity tiers: Llama-3.1-70B-Instruct, Tulu-3-70B, phi-4, Llama-3.1-8B-Instruct, Yi-1.5-34B-Chat, and Qwen2.5-Coder-7B-Inst.

SWE-bench Verified. We leverage trajectories from princeton-nlp/SWE-bench-Verified test split. Evaluation domains are defined by the top-5 resolved repositories. We benchmark two distinct 6-agent configurations: a Strong/Generalist panel (comprising Claude-4-Sonnet, GPT-5, Qwen3-Coder-480B, GLM-4-6, Kimi-K2, and DeepSeek-V3) and a Weaker/Complementary panel (including Enginelabs, Claude-3.5-Haiku, GRU, Codeshellagent-Gemini, SWERL-Llama3-70B, and Amazon-Nova-Premier).

G.4. Multi-Task Collaborative Performance

We evaluate multi-task delegation mechanism using evaluation traces from the open-llm-leaderboard/model-details dataset. The model suite spans three capacity tiers—including Large (Llama-3.1-70B-Instruct, Tulu-3-70B, phi-4), Mid (Yi-1.5-34B-Chat), and Small (Llama-3.1-8B-Instruct, Qwen2.5-Coder-7B-Inst) — evaluated across 5 reasoning domains from the IFEval, MATH, and BBH suites. To isolate the routing component's performance, we select these domains where no single model dominates, partitioning each into a 40% train split for empirical competence estimation and a 60% held-out test split. The experimental graph consists of 30 nodes (model-domain pairs) in a randomly generated topology. We benchmark multi-task combinations by running the delegation mechanism for 50 iterations with 1,000 sampled test questions per domain to ensure robust performance metrics. The **`AgentSociety`** mechanism runs similarly to G.1.

H. LLM Agent Prompts

The following prompts were used for experiments characterizing the social intelligence of LLM agents presented in Fig. 3, with the same prompts used across all LLM agents.

Listing 1. System Prompt (Describing AgentSociety)

```
You are a node in a liquid democracy network making strategic decisions about voting and
    information sharing in order to maximise your payment (reward). YOU ARE A UTILITY
    MAXIMISER WITH YOUR UTILITY = PAYMENT. The game is defined as follows

### Tasks and Competence
- There is a single task to be completed
- Each node has ACTUAL COMPETENCE for the task (0.0 to 1.0 scale)
- Each node reports competence during delegaion based on which the payment (reward)
    received is computed. Each node is strategic here to report competence in such a way
    that the payment is maximized.
- Each node also diffuses a competence to each of its neighbours (could be different to
    different neighbours). Each node is strategic here because higher than required
    diffusion hurts its payment share, while lower than required might not get it on the
    winning path which would mean that there is no payment altogether.
- All agents maintain a log of what was diffused, so once a certain competence is diffused
    , not diffusing any information in the following iterations is the same as diffusing
    the previously diffused value as neighboring nodes log what was received from each
    node.
- The goal of each node is to maximize its payment and not just receive a payment.
- A positive payment means income and a negative payment means a penalty. Penalty is many
    times higher than income.
- As the diffusion happens between neighbors each iteration, there is a delay for the
    diffused value to reach extended neighbors in the graph to then influence them to
    change their voting based on diffused value.
- Therefore the agents need to consider that the reward and penalty can sometimes be
    lagging with respect to the information diffused.
- A rational agent starts with diffusing low values below its reported competence for
    payment and increases its diffusion value until required to maximize payment.
- A rational agent does not report competence for payment to be higher than max(actual
    competence, competence obtained from neighbors) on the primary task.

The system operates in the following steps -
The system broadcasts the task to all nodes.
The first step for each participating node is to vote, either for themselves or for one of
     their neighbors.
After the voting process concludes, votes aggregate at gurus, defined as agents who vote
    for themselves. Each guru represents a group of agents that have voted to it, with
    voting occurring transitively.
The voting paths are determined by the delegation paths formed by the transitive voting.
    The winning path is defined as the group with the highest number of agents.
Voting for a neighbor could be beneficial because there is finally only one winning path
    identified and only agents that make up this winning path have an opportunity to get
    paid.  The goal is to figure out who to vote for, given the payment mechanism in order
     to maximise own payment.
During voting, each node also sends the reported competence, that it can deliver either by
     itself or leveraging its neighbors by voting to them, to the system that will
    eventually be used for payment calculation.
Within this winning path, not all nodes are paid. Only the ones deemed critical, or those
    that would be part of a (different) winning path in case they had voted for themselves,
     are paid. These nodes make up the critical path sorted in the order of their
    competencies.
After identifying the critical path the framework assigns payments to utilizing the
    formula below by plugging their respective reported competence into it.
Let the critical path C_{t_k} be expressed as D_w^{t_k} = (a_1, a_2, ..., a_{k-1}, a_k = w
    ), which denotes the ordered sequence of agents whose vote forwarding was necessary
    for w to receive the allocation of task t_k. The payment function (positive payment
    means the agent gets paid) for an agent a_i in D_w^{t_k} is then given by p^(s_{
    sigma_j})(a_i) equals:
```

```
*   I(c_{i+1} >= c_i) * (f(c_i) - f(c_{i-1})), if a_i is in D_w^{t_k} excluding {w};
*   I(c_{i+1} >= c_i) * (f(c_i) - f(c_{i-1})), if a_i = w;
*   0, if a_i is not in D_w^{t_k},

where: c_i is the competence reported of agent a_i on task t_k, c_{i-1} = 0 if a_i = a_1,
    c_{i+1} = true competence of winner if a_i = w. I(.) is the indicator function,
    returning +1 if the condition holds and alpha(negative value) otherwise, f(.) is a
    monotone function modulating payments, and c is a baseline cost to the winning agent w
     to execute the task. Intuitively, the payment policy rewards or penalizes
    intermediaries according to the incremental competence they contribute along the vote
    chain.
After the payment is made, all agents are allowed to make a decision based on their
    current information state and payment outcome on whether they want to diffuse
    information to their neighbors to improve their own utility or payment.
```

Listing 2. Delegation Prompt

```
You are rational and intelligent Agent {node_id} in a system of all rational and
    intelligent agents making a delegation decision and reporting a competence for payment
     based on your delegation decision.

YOU NEED TO PAY ATTENTION AND UNDERSTAND TO THE PAYMENT FUNCTION IN ORDER TO MAXIMIZE YOUR
     PAYMENT for both your delegation decision as well as competence for payment.

## Your Information
- Your Primary Task: Task {primary_task}
- Your Actual Competence: {actual_competence}
- Tasks Being Voted On: {tasks}

## Your Memory (History from Previous Iterations)
{memory_info}

## Your Neighbors competence (same primary task only)
{neighbors_info}

## Decision Required
You must choose  for Task {primary_task}, who to delegate your vote. Your options are:
1. Vote for yourself (delegate to yourself)
2. Delegate to one of your neighbors listed above (only those with primary task {
    primary_task})

If you choose to vote for yourself, use your own node ID ({node_id}).

Based on who you delegated to, report a corresponding competence which determines your
    payment.

Use your memory to learn from past delegation choices and their resulting payments.
You need to ensure that you delegate to a neighbor that at least can guarantee the
    performance you have already diffused. Otherwise your neighbors will no longer trust
    you to be accurate in your claims.

## Response Format
Respond with ONLY a JSON object. ALL three fields are REQUIRED and **DO NOT PROVIDE AN
    EMPTY RESPONSE**:
{{"delegate_to": <node_id>, "competence_for_payment": {{<task_id>: <competence_value>}}, "
    reasoning": "brief explanation"}}

Example response:
{{"delegate_to": 5, "competence_for_payment": {{"1": 0.7}}, "reasoning": "Node 5 has
    ..."}}

Field descriptions:
- delegate_to: The node ID you are delegating to (use your own ID {node_id} if voting for
    yourself)
```

```
- competence_for_payment: REQUIRED. A dictionary mapping each task_id (integer) to the
    competence value (float) you want to report for payment calculation. Include ALL tasks
     from {tasks}.
- reasoning: Brief explanation of your decision, keep it concise as there is a limit on
    the length
```

Listing 3. Diffusion Prompt

```

You are rational and intelligent Agent {node_id} in a system of all rational and
    intelligent agents deciding what information to diffuse to each of your neighbors.
Bear in mind that diffusion at this iteration can depend on the delegation decision of the
     same iteration.

## Your Information
- Your Primary Task: Task {primary_task}
- Your Actual Competence: {actual_competence}
- Delegation Decision of current iteration: {delegated_to_info}
- Competence Reported for Payment (current iteration): {competence_for_payment_info}

## Your Memory (History from Previous Iterations)
{memory_info}

## Payment Outcome
{payment_status}

## Your Neighbors
- Same Primary Task (Task {primary_task}): {same_task_neighbors}
- Different Primary Tasks: {diff_task_neighbors}

## Neighbor Competence (what they have diffused to you)
Current State of Neighbors Competence: {neighbors_competence_info}

- You can strategically diffuse DIFFERENT competence values to DIFFERENT neighbors
  - Consider each neighbor's potential influence on maximizing your payment

- Use your memory to learn from past diffusion decisions and their impact on payments

## For example
If this is the received information -

# Your Information
- Your Primary Task: Task 1
- Your Actual Competence: {{"1": 0.2}}
- Delegation Decision of current iteration: You delegated to Node 5
- Competence Reported for Payment (current iteration): {{"1": 0.439}}

# Current State of Neighbors Competence: - Node 3 (Primary Task 1): Competence = {{"1":
    0.419}}
- Node 5 (Primary Task 1): Competence = {{"1": 0.439}}

Here, diffusing a higher value of 0.425 to Node 3 would help you obtain node 3's vote
    while not losing substantial payment share. In this case, you would be acting as an
    intermediary transferring information from one node to another.


## Decision Required
Decide whether to diffuse information and what competence values to diffuse to EACH
    neighbor individually.

## Response Format
Respond with ONLY a JSON object in this exact format:
{{
  "should_diffuse": true/false,
  "neighbor_updates": {{
```

```
    "<neighbor_id>": {{"<task_id>": competence_value, ...}},
    "<neighbor_id>": {{"<task_id>": competence_value, ...}}
  }},
  "reasoning": "brief explanation of strategy"
}}

Example:
{{
  "should_diffuse": true,
  "neighbor_updates": {{
    "2": {{"1": 0.7}},
    "5": {{"1": 0.5}}
  }},
  "reasoning": "Reporting higher competence to node 2 because of ..."
}}

If should_diffuse is false, the updates will be ignored.
You must provide an update for each neighbor you want to diffuse information to.
```