THEORY OF SCENE: BREAKING THE SYMMETRY TRAP IN MULTI-AGENT LLM COORDINATION

Liangqi Yuan[1]* **Wenzhi Fang**[1] **Shiqiang Wang**[2] **Christopher G. Brinton**[1]
[1]Purdue University [2]University of Exeter

ABSTRACT

Multi-agent systems built on large language models (LLMs) are largely homogeneous, as their agents behave alike even across distinct LLMs. We show that when such agents act concurrently without communication, they collide on targets they must split and diverge on targets they must take together, a double failure we term the *symmetry trap*. Theory of Mind (ToM), widely used for coordination without communication, cannot escape this trap, since homogeneous agents form the same prediction of one another and respond to it in the same way. We propose **Theory of Scene (ToS)**, a training-free reasoning schema in which each agent reads its public role, the only difference between the agents, and the task context they all observe. Homogeneous agents thereby derive one division of labor, each taking the part its role fixes, which turns homogeneity from the cause of the trap into the cure. ToS reads the role together with the scene through *role gating*, which determines whether ownership overlaps or is already divided, and the task context through *task coupling*, which infers whether the team must converge on each target, divide it, or take its stages in turn. We evaluate on DivvyBench, a controlled environment we introduce, whose target types make an episode Competitive, Cooperative, or Mixed across Tabletop, Airspace, and Household scenarios, and on two established agentic benchmarks, GovSim and Overcooked. ToS outperforms all six baselines on every benchmark, and each baseline falls far behind it in at least one setting. Against ToM given the same inputs, ToS raises the DivvyBench success rate from 71.1% to 99.6%, the GovSim total gain from 207 to 400, and the Overcooked level-normalized throughput from 1.41 to 1.67.

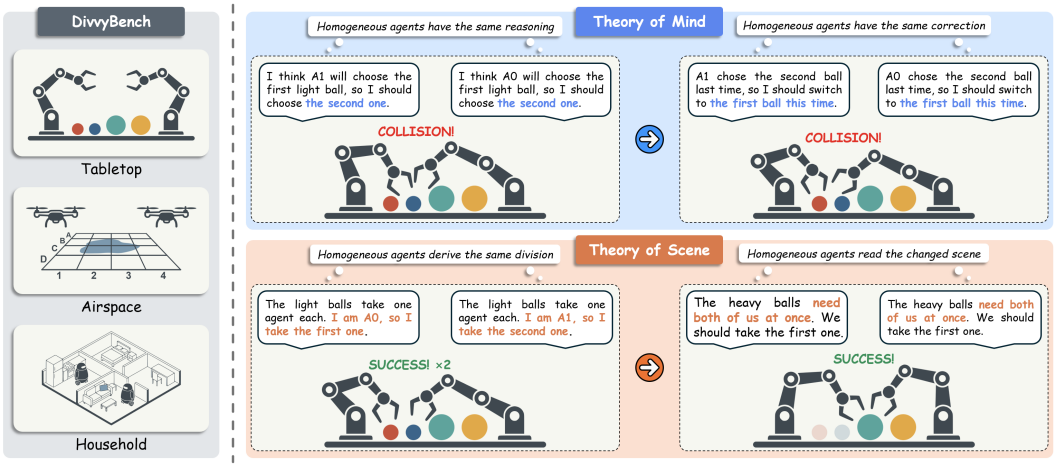

Figure 1: **Symmetry trap: ToM vs. ToS.** *Left:* DivvyBench covers three scenarios, Tabletop picking, Airspace survey, and Household assistance. *Top:* Under ToM, homogeneous agents reason identically at every step, so they collide, correct, and collide again. *Bottom:* Under ToS, the agents read the division of labor from the shared scene and the public role, diverging onto the Competitive targets and converging on the Cooperative targets.

*Corresponding author: Liangqi Yuan (liangqiy@purdue.edu)

1 INTRODUCTION

Large language model (LLM) agents increasingly work in teams, typically several instances of one LLM (Li et al., 2023a; Hong et al., 2024). Coordination among such teams usually relies on (i) a means for the agents to communicate (Chopra et al., 2025; Grötschla et al., 2025; Jian et al., 2026; Yang et al., 2025) or (ii) a predefined division of labor among the agents (Hong et al., 2024; Qian et al., 2024). Communication requires resources (e.g., wireless channels), which can be costly, forbidden by policy, and/or unreliable in multi-robot or multi-vehicle teams. A predefined division is realized by prompts that assign each agent a fixed responsibility, such as planner, executor, or verifier, yet the right division changes with the scene at runtime. Lacking both, the agents must coordinate unaided on tasks that often mix Competitive targets, which they must divide, with Cooperative targets, which require several of them at once (Carroll et al., 2019; Agapiou et al., 2022; Piatti et al., 2024). Yet such agents tend to be homogeneous, behaving alike because even distinct LLMs produce strikingly similar outputs (Jiang et al., 2025; Kim et al., 2025; Goel et al., 2025). They therefore collide on Competitive targets, and any mechanism that separates them there also splits them across Cooperative targets. We identify this double failure as *the symmetry trap, which arises when homogeneous agents act concurrently without communication and are distinguished only by a public role* (Figure 1, top).

Escaping the symmetry trap therefore requires each agent to derive a division of labor by itself, such that all agents arrive at the same division. A public role alone identifies each agent but divides no work, and raising the sampling temperature separates the agents at random, which leaves no division shared. Theory of Mind (ToM), a common approach to multi-agent coordination, has each agent predict the others and best-respond to the prediction (Agashe et al., 2025; Cross et al., 2025; Zhang et al., 2024a), yet a homogeneous agent that predicts its teammates recovers only its own reasoning. Conventions and index-based assignments determine which agent takes which target once the team knows it must separate (Schelling, 1980; Ashery et al., 2025), but do not infer from the scene whether a target requires divergence or convergence. Multi-agent reinforcement learning (MARL) breaks symmetry by training over the symmetry group (Hu et al., 2020), which is unavailable to LLM agents that coordinate at inference. We therefore ask the following question. *How can homogeneous LLM agents escape the symmetry trap by deriving the same division of labor from one shared scene, one that diverges, converges, or takes turns from one target to the next?*

To answer this question, we propose **Theory of Scene (ToS)**, a training-free reasoning schema that derives the division of labor from the observable scene, whereas ToM predicts the other agents' hidden intent. The intuition is that a scene admits one or more rational divisions of labor, and that any agent can derive one of them from the scene alone. ToS reads the two signals the setting supplies: (i) the *public role*, the only signal that distinguishes homogeneous agents, and (ii) the *task context*, which specifies the work the team must complete. The shared scene thus leads homogeneous agents to the same division, and the public role assigns each agent its own part, so the team diverges on Competitive targets and converges on Cooperative ones without exchanging any message (Figure 1, bottom). Because the interaction is agentic, a disagreement at one step is resolved over the steps that follow. Our contributions are summarized as follows.

- **The Symmetry Trap and DivvyBench.** We characterize the *symmetry trap*, in which homogeneous agents collide where the targets must be divided, while the randomness or prediction that separates them also separates them where they must converge. To isolate the trap, we build DivvyBench, a controlled environment that reduces each step to the choice of a target and composes each episode of one target type or a mix of both, across the Tabletop, Airspace, and Household scenarios.
- **Theory of Scene.** We propose ToS reasoning, in which each agent reads the shared scene through two mechanisms. (i) *Role gating* settles from the scene and the public role whether ownership is overlapping or already divided, and derives a division only where it is overlapping. (ii) *Task coupling* infers what each target requires of the team, and the team accordingly converges on it, divides it, or takes its stages in turn, which determines the division of labor.
- **Empirical Validation.** Three agentic benchmarks, our DivvyBench, GovSim, and Overcooked, show that ToS performs best throughout, while the strongest of the six baselines, each fixed to one reasoning schema, varies across benchmarks. Incorporating ToS's readings into ToM improves ToM yet leaves it below ToS, because ToM derives its action from its prediction, which these readings leave unconstrained.

2 Related Work

Multi-Agent Coordination: Competitive vs. Cooperative. Prior work on multi-agent LLM coordination studies competition and cooperation as separate problems and assumes a communication channel. *Competitive* interaction, where the agents contend for the same outcome, is approached through negotiation, debate, and strategic games (Du et al., 2024; Duan et al., 2024). *Cooperative* interaction, where the agents share a goal, is approached through role-based frameworks that decompose a task and negotiate a plan (Qian et al., 2024; Wu et al., 2024) and through joint-task benchmarks (Sun et al., 2025; Qian et al., 2026; Kim et al., 2026). Each line of work fixes a single setting, and even evaluations spanning both treat them as separate games, so an agent never has to adapt its coordination strategy within an episode. From embodied cooperative suites (Zhang et al., 2024b) and open-ended survival worlds (Tessera et al., 2026) to language games (Li et al., 2025; Abdelnabi et al., 2024; Lupu et al., 2025), coordination benchmarks provide this channel, through which the agents negotiate the division of labor. Mixed-motive suites in MARL place both settings in one episode (Papoudakis et al., 2021; Agapiou et al., 2022), which the agents learn in training. We introduce DivvyBench, the first environment in which (i) the agents are homogeneous and separated only by a public role, (ii) they act simultaneously, before seeing the others' actions, (iii) they have no communication channel, and (iv) each target carries its own demand, so the same agents must diverge and converge within one episode.

Non-Communicating Coordination: ToM and Its Alternatives. Coordination without a channel has been approached in three ways, each resting on something this setting removes. The first predicts the other agents and best-responds, as in MARL and Bayesian inverse planning (Wang et al., 2022; Wu et al., 2021). LLM agents with ToM infer the beliefs and intentions behind the other agents' behavior (Li et al., 2023b; Mu et al., 2026), as ProAgent does by predicting a teammate's action (Zhang et al., 2024a) and HypMinds by testing hypotheses about other agents (Cross et al., 2025), both built for heterogeneous teammates. Hayler et al. (2026) tell the agents their partner is an identical copy and still find no convention settled, and they add a self-check that only rejects flippable conventions. The second is conventions and focal points, which coordinate on a salient option under a commonly known rule fixed before the scene is read (Schelling, 1980; Ashery et al., 2025; Aharon et al., 2026). The third resolves symmetry with MARL techniques, randomizing over the symmetry group during training (Hu et al., 2020), applying off-belief learning (Hu et al., 2021), or sampling a rank ordering at execution (Patil et al., 2026). We therefore base ToS on the two signals that the setting provides, the public role and the task context, which removes the need for (i) predicting the other agents, (ii) a division of labor fixed before the scene is read, and (iii) training on the task or randomness at execution.

3 The Symmetry Trap

The DivvyBench Environment. DivvyBench distills existing coordination benchmarks to the one decision their LLM agents make, the choice of a target, and discards the simulator that executes it. In C-WAH (Zhang et al., 2024b) the LLM chooses which room to search and in Overcooked (Carroll et al., 2019; Sun et al., 2025) which station to use, and the surrounding simulator confounds the outcome, since a team that underperforms may have failed to coordinate or may simply have been overwhelmed by the scene. In DivvyBench, homogeneous agents must take every target in a shared scene (Appendix C.2), acting at the same time without any message and deciding only which target to select. Scoring is all-or-nothing, so an episode counts only when every target is taken within the budget, and each collision or stuck step spends part of it. The same rules run in three scenarios, in which the agents collect colored balls in *Tabletop*, survey grid areas in *Airspace*, and search indoor locations in *Household*, and we say that an agent takes a target when it completes that operation. A Competitive target is taken only when exactly one agent selects it, so agents that select the same one collide and take nothing, and a team that diverges completes the task in half the steps one agent needs alone. A Cooperative target is taken only when every agent selects it in the same step, so agents that split across targets take nothing. A scene that holds both types is Mixed, where the two demands fall on the same agents within one episode. The rest of this section runs on Tabletop, with every agent an instance of one LLM under one prompt, and holds the two pure settings apart, since a mixed scene superposes them and leaves each failure unattributable.

The Public Role and Temperature Cannot Break Competitive Symmetry. The first candidate for breaking the symmetry is the public role alone, and Figure 2 (left) shows that w/o Role (identical prompts) and w/ Role (a fixed public role, Leader or Follower) both collapse in the Competitive setting, where neither completes any episode at temperature 0. A label such as Leader identifies an agent without specifying which targets belong to it, so every agent reasons from the same observation to the same salient target and, after each collision, revises its choice in the same way and selects the same target again. The second candidate, a higher temperature, separates the agents on some steps through independent sampling, yet agents sampling from one shared distribution collide with positive probability at every step, and each collision consumes a step, so the separation never becomes reliable. Both candidates therefore exhibit the symmetry trap, since without randomness the agents converge but cannot diverge, and the randomness that separates them on some steps also lowers their Cooperative success (Figure 2, right), where they must converge.

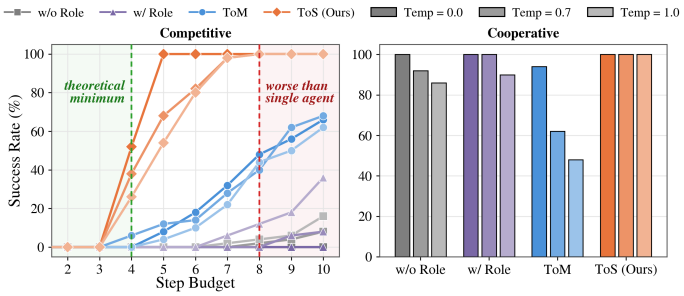

Figure 2: **Each baseline succeeds in one setting and fails in the other.** DivvyBench Tabletop with 2 agents, 8 balls, and a 10-step budget. *Left:* Competitive success rate against the step budget. *Right:* Cooperative success rate at the full budget.

ToM Fails under Cooperative Symmetry. The third candidate is ToM, in which each agent predicts the others and best-responds to the prediction. ToM raises success in the Competitive setting, since an agent that predicts which targets the others will choose selects a different one, and even an imperfect prediction allows the team to progress as long as the agents separate. The Cooperative setting imposes a stricter requirement, since the agents must independently select exactly the same target. A homogeneous agent that predicts its teammate simulates a copy of itself, so the prediction returns the reasoning the agent already had and uses no external distinction, without which homogeneous agents cannot break this symmetry (Angluin, 1980; Riedl, 2026). Their predictions therefore agree when one target is clearly salient and differ when several are equally salient, since each agent resolves the ambiguity privately through independent decoding, and the agents then select different targets and make no progress, increasingly so as the temperature rises (Figure 2, right). ToM predicts the teammate's target with more than 90% accuracy in the Cooperative setting, where agents succeed even without prediction, but with only about 30% accuracy in the Competitive setting, where the agents must divide the targets (Appendix A.8). The prediction is thus accurate where it is least needed and inaccurate where it is needed most, so ToM also exhibits the symmetry trap.

4 Theory of Scene

Setup. We consider N homogeneous agents that coordinate over a horizon of T steps without communication. All agents instantiate the same policy π_θ under a shared reasoning schema R, distinguished only by a public role r_i. At each step $t = 1, \ldots, T$, agent i has the environment observation o_i^t (the task context, the current scene, and its legal actions) and the other agents' past actions $\mathbf{a}_{-i}^{<t}$, where $-i$ denotes the agents other than i, and produces one action through a single policy call:

$$a_i^t \sim \pi_\theta(\cdot \mid o_i^t,\, r_i,\, \mathbf{a}_{-i}^{<t},\, R) \quad \text{for all } i = 1, \ldots, N \text{ concurrently.} \tag{1}$$

Each agent acts without observing the others' actions at the same step, so $\mathbf{a}_{-i}^{<t}$ runs only through step $t-1$. The environment then executes the joint action and returns the next observations.

4.1 Theory of Mind versus Theory of Scene

Two Reasoning Schemas. A coordination method is a choice of reasoning schema R, the ordered fields a single π_θ call produces autoregressively, each conditioned on the observation, the role, and the earlier fields it reads. ToM and ToS share the `state` and `action` fields and differ in the two middle fields that carry the coordination strategy, and `action` acts on what those fields produce:

- **ToM** (Algorithm 1) corrects a belief about the other agents, predicts their next actions, and best-responds,

$$\underbrace{b_i^t \sim \pi_\theta\big(\cdot \mid o_i^t, r_i, (\mathbf{a}_{-i}^{<t}, \hat{\mathbf{a}}_{-i}^{<t})\big)}_{\texttt{belief}}, \quad \underbrace{\hat{\mathbf{a}}_{-i}^t \sim \pi_\theta\big(\cdot \mid o_i^t, r_i, b_i^t\big)}_{\texttt{predict}}, \quad \underbrace{a_i^t \sim \pi_\theta\big(\cdot \mid o_i^t, r_i, \hat{\mathbf{a}}_{-i}^t\big)}_{\texttt{action}}, \tag{2}$$

  where $\hat{\mathbf{a}}_{-i}^t$ denotes agent i's prediction of the other agents' actions at step t and $\hat{\mathbf{a}}_{-i}^{<t}$ its predictions from the earlier steps, and the belief b_i^t is revised by comparing each past action in $\mathbf{a}_{-i}^{<t}$ with the prediction made for it, the correction that defines ToM.
- **ToS** (Algorithm 2) reads the present scene and acts on what it reads,

$$\underbrace{\rho_i^t \sim \pi_\theta\big(\cdot \mid o_i^t, r_i\big)}_{\texttt{role}}, \quad \underbrace{\tau_i^t \sim \pi_\theta\big(\cdot \mid o_i^t, r_i\big)}_{\texttt{task}}, \quad \underbrace{a_i^t \sim \pi_\theta\big(\cdot \mid o_i^t, r_i, \rho_i^t, \tau_i^t\big)}_{\texttt{action}}, \tag{3}$$

  where the role reading ρ_i^t and task reading τ_i^t are inferred independently from o_i^t and r_i, and swapping the order in which they are emitted does not significantly lower performance (Table 2).

Algorithm 1 Theory of Mind

Require: observation o, the other agents' past actions $\mathbf{a}_{-i}^{<t}$

1: `state`: ground in the observation o
2: `belief`: correct from predicted vs. actual actions
3: `predict`: predict the others' next actions
4: `action`: best-respond to the predicted actions

Algorithm 2 Theory of Scene (Ours)

Require: observation o

1: `state`: ground in the observation o
2: `role`: infer overlapping vs. divided ownership
3: `task`: infer converge vs. diverge coordination
4: `action`: derive a division of labor

Remark (ToS Reasoning without the Other Agents' Actions)**.** Unlike ToM, whose belief and prediction are read off the other agents' past actions $\mathbf{a}_{-i}^{<t}$, ToS reads only the observation o_i^t and its role r_i. The observation holds the task context and the global world state, which targets remain, which stations are occupied, and whether the last step took anything, and it attributes nothing to any agent. ToS reads its division from that state, corrects no belief about any agent, and therefore applies where the other agents cannot be observed. We hand both schemas the same inputs so that R is the only variable in the comparison, and withholding $\mathbf{a}_{-i}^{<t}$ does not significantly lower ToS (Table 2).

4.2 Role Gating and Task Coupling

Role Gating ρ_i^t. Role gating turns the public role r_i into a reading of the scene and takes two inputs. First, the observation o_i^t carries the task context and the current scene together with the outcome of the agent's own actions. Second, the public role r_i is the one signal that tells homogeneous agents apart, fixed before the episode and unchanged as the scene changes. Whether it is a specific assignment or only a name is a joint property of the role and the observation, so one label divides the work in one scene and divides none of it in another. The `role` field combines them into the role reading ρ_i^t, and the gate settles only whether a division has to be derived at all, while the task reading determines the division itself. The reading classifies the ownership of the work:

- *Overlapping*, where the scene and the role name no agent's part and the agents work the same targets, the role distinguishing the agents without reserving work for any of them. The agent derives a division from the task reading and takes the part its role indexes, since agents that reason alike would otherwise fall into the symmetry trap.
- *Divided*, where the scene and the role already name each agent's part. The agent advances its own part, taking the single most useful action toward the goal and first producing any input a later step still lacks.

Task Coupling τ_i^t. Task coupling turns each target into a division that homogeneous agents derive the same way. The `task` field infers what each target requires of the team and records it as the task reading τ_i^t. The schema names three couplings:

- *Joint*, where the target's outcome requires more than one agent on it at once, which the team answers by converging on it.
- *Exclusive*, where the target is rival and what one agent takes is taken from the rest, which the team answers by dividing it into disjoint shares, each indexed by an agent's role r_i.
- *Sequential*, where the target depends on an input not yet in place, which the team answers by taking its stages in turn, each agent working the stage its role owns.

Combining ρ_i^t and τ_i^t into the Action a_i^t. The last statement of Equation (3) conditions the action on both readings, and in the `action` field the agent derives from them the division of labor for the whole team and acts on it at every step. A scene can admit more than one rational division, since its targets can be ordered by their position in the observation, by name, by difficulty, distance, or priority, or by any other property they carry. Homogeneous agents reason alike and therefore arrive at the same division more readily, whereas agents with different reasoning styles can derive different ones. Across the steps of an agentic interaction, the scene changes at every step and realigns those divisions until the agents agree on one. The one difference between the agents (i.e., the public role) then fixes which part of that division each takes, so every agent acts on its own part without exchanging a message.

4.3 Analysis of the Coordination Probability

Consider a single step, in which a scene offers $K \geq N$ targets indexed by k and every agent selects one of them. The policy π_θ of Equation (1) induces for agent i a distribution $p^{(i)}$ over the targets, with $p_k^{(i)}$ the probability of target k, and the N agents sample from their distributions independently, since no agent sees another's action at the same step. Two probabilities summarize the step, the probability P_{div} that the N agents select N distinct targets, an event termed divergence, and the probability P_{conv} that all of them select the same target, termed convergence. A schema blind to r_i puts the agents on one distribution whenever their remaining inputs coincide, a condition the first step of an episode satisfies. Two propositions bound these probabilities, one for that shared distribution and one for the general case, and Appendix B proves both.

Proposition 4.1 (Bounds under One Shared Distribution)**.** If all N agents sample independently from one distribution p, a single step satisfies

$$\begin{aligned} &P_{\text{div}}(p) = N!\, e_N(p), && \max_p P_{\text{div}}(p) = \textstyle\prod_{n=1}^{N-1}\left(1 - \frac{n}{K}\right) < 1, \\ &P_{\text{conv}}(p) = \textstyle\sum_{k=1}^{K} p_k^N, && \max_p P_{\text{conv}}(p) = 1, \end{aligned} \tag{4}$$

where $e_N(p) = \sum_{k_1<\cdots<k_N} p_{k_1} \cdots p_{k_N}$ is the elementary symmetric polynomial of degree N.

Remark (Policy Independence)**.** Proposition 4.1 bounds every schema whose agents sample one distribution, which a schema blind to r_i does whenever their inputs coincide. The bounds of Equation (4) depend on N and K alone, so they hold for every model backbone, every shared prompt, and every temperature at once, while a temperature sweep reaches finitely many policies. At $N = 2$ and $K = 8$, as in our DivvyBench experiments, converging is unconstrained, while diverging succeeds with probability at most $1 - 1/8 = 7/8$ in one step, since agents sampling one shared distribution select the same target with probability at least $1/8$. Compounded over the four steps of the theoretical minimum, the bound is 0.27, even when the distribution is retuned as targets are taken.

A schema that reads r_i can give different agents different distributions, and we quantify the effect of that difference. Let $\delta_{ij} \in [0, 1]$ denote the total variation distance between the distributions of agents i and j. It satisfies $\delta_{ij} = 0$ when the two distributions are identical and $\delta_{ij} = 1$ when they share no target, and its complement $1 - \delta_{ij} = \sum_k \min\left(p_k^{(i)}, p_k^{(j)}\right)$ is the overlap of the two distributions. Let $\delta_{\min}$ and $\delta_{\max}$ be the smallest and largest δ_{ij} over the pairs of agents.

Proposition 4.2 (Bounds under Per-Agent Distributions)**.** If the N agents sample independently, agent i from its distribution $p^{(i)}$, then a single step satisfies

$$\begin{aligned} &P_{\text{div}} \leq 1 - \tfrac{1}{K}\left(1 - \delta_{\min}\right)^2, && P_{\text{div}} = 1 \text{ only if } \delta_{\min} = 1, \\ &P_{\text{conv}} \leq 1 - \delta_{\max}, && P_{\text{conv}} = 1 \text{ only if } \delta_{\max} = 0. \end{aligned} \tag{5}$$

Remark (The Two Readings of ToS Adjust δ_{ij})**.** The distances enter the two bounds of Equation (5) with opposite signs. Raising $\delta_{\min}$ raises the bound on P_{div}, since distributions that overlap less collide less, and raising $\delta_{\max}$ lowers the bound on P_{conv}, since a pair selects the same target with probability at most its overlap $1 - \delta_{ij}$. The two only-if conditions of Equation (5) therefore pull δ_{ij} to opposite ends. The demand is per target, since the pair selects target k together with probability $p_k^{(i)} p_k^{(j)}$. Targets the team must divide require $p_k^{(i)} p_k^{(j)} = 0$ and thus $\delta_{ij} = 1$, and a target it must share requires the reverse, so the same pair needs δ_{ij} to change from one step to the next with the targets that step addresses. The two readings of Equation (3) place each target in its group, with the task reading τ_i^t inferring from the shared scene which demand a target makes and the role reading ρ_i^t settling whether the distributions differ at all. Appendix A.9 measures δ_{ij} under each method.

5 EXPERIMENTS

5.1 EXPERIMENTAL SETUP

Environments and Metrics. We evaluate on three agentic benchmarks, all run with the agents acting concurrently and without any inter-agent communication, and Appendix C gives their rules and per-benchmark parameters in full.

- **DivvyBench:** 2 agents must take all 8 targets within 10 steps under the Competitive, Cooperative, and Mixed settings. We report success and stuck rate, the fraction of steps that take nothing, over 150 episodes per setting, 50 in each of the Tabletop, Airspace, and Household scenarios.
- **GovSim** (Piatti et al., 2024): 5 agents privately harvest from one regenerating resource pool for at most 12 rounds, and the pool collapses once it falls below 5 units. We report survival and total gain over 50 games in each of the Fishery, Pasture, and Pollution scenarios.
- **Overcooked** (Carroll et al., 2019; Sun et al., 2025): 2 agents cook over 6 difficulty levels within a 50-step episode, sharing every single-use station. Success saturates, so we report throughput in dishes per episode over 15 episodes per level, alongside an average normalized to ReAct.

Baselines. We group the baselines by the signal each uses to coordinate. **Prediction-free** baselines carry no model of the other agents, and comprise ReAct (Yao et al., 2023), which reasons over the shared observation and takes the single most useful action, Focal (Aharon et al., 2026), which applies the focal-point prompt unchanged and takes the option that stands out from the rest, and CFD (Hayler et al., 2026), which adds a self-check that rejects any action resting on an arbitrary, flippable convention. **ToM** baselines predict the other agents, and comprise Classic ToM (Algorithm 1), which best-responds to its prediction of their next actions, ProAgent (Zhang et al., 2024a), which infers the intention behind their actions and plans a skill that complements it, and HypMinds (Cross et al., 2025), which maintains and tests hypotheses about their hidden strategies.

Implementation. All agents run one LLM backbone, Qwen3.5-35B-A3B (Qwen Team, 2026), under a prompt that is identical at each step except for the single line naming the public role (e.g., Chef or Assistant in Overcooked, Leader or Follower in DivvyBench, and A0 to A4 in GovSim). Across methods only the reasoning schema changes, itself appended verbatim to every task prompt over the three benchmarks, and we decode at a fixed temperature of 0.7. Each cell reports the mean over the trials above, with the spread ($\pm$) the standard error of that mean. Appendix C.1 gives the rest of the decoding and serving configuration, and Appendix D the task prompt of each DivvyBench setting with the ToS schema appended.

5.2 RESULTS

Comparison across the Coordination Spectrum. Each baseline in Table 1 collapses for one of three reasons. *(i) Nothing to break the symmetry.* ReAct acts on the shared observation alone, so homogeneous agents commit to the same option. They select the same target on DivvyBench and the same station on Overcooked, where no agent prepares the input that station needs, and with five agents on a shared pool each harvests with no reason to hold back. *(ii) A prediction that mirrors the predictor.* The ToM methods are the three strongest baselines in DivvyBench Competitive, ReAct alone exceeds all three in DivvyBench Cooperative, and none survives half the GovSim horizon on average. Predicting a homogeneous agent returns the predictor's own ambiguity among equally salient targets, which splits agents that must share a target. GovSim rewards each agent for what it takes from a resource the others harvest as well, so the prediction that they will harvest becomes a reason to harvest first, and the pool collapses. *(iii) One rule for every situation.* Focal separates even where the task requires the agents to meet, because a salience criterion treats every target the same way, and CFD never commits, because it rejects any action resting on a convention that could have been flipped. Neither reads what each target requires of the team, so each fails wherever its fixed rule does not fit the target. ToS avoids all three failure modes by deriving the division of labor from the scene at each step, which breaks the symmetry without predicting the other agents.

First Step vs. Recovery after Failure. Figure 3 reports the first step and the step after the first failure. At the first step no action has been taken, so every method holds the same observation and public role, and the reasoning schema is the only source of the lead ToS holds in every setting. No baseline uses the one asymmetry it is given, since ReAct acts on the observation, the ToM methods act on a prediction formed without seeing the other agents act, and Focal and CFD apply a rule

Table 1: **ToS is the only method that succeeds on all three benchmarks, and it outperforms every baseline on almost every metric.** Best per column in bold, ToS shaded, and ↑ and ↓ mark the improving direction. Appendix A.10 reports a two-sided 95% confidence interval and a significance test for every gap on the headline statistic of each benchmark.

		Competitive		Cooperative		Mixed		Avg.	
DivvyBench	Method	Success (%) ↑	Stuck (%) ↓	Success (%) ↑	Stuck (%) ↓	Success (%) ↑	Stuck (%) ↓	Success (%) ↑	Stuck (%) ↓
	ReAct	40.7 ±4.0	48.7 ±1.4	96.7 ±1.5	2.5 ±0.7	56.7 ±4.1	26.9 ±1.1	64.7 ±2.3	26.0 ±1.1
	Focal	47.3 ±4.1	45.4 ±1.4	42.0 ±4.0	36.6 ±2.4	50.0 ±4.1	29.0 ±1.1	46.4 ±2.4	37.0 ±1.1
	CFD	57.3 ±4.1	31.8 ±1.8	66.7 ±3.9	20.5 ±2.1	36.0 ±3.9	33.9 ±1.7	53.3 ±2.4	28.7 ±1.1
	Classic ToM	72.7 ±3.7	35.6 ±1.5	86.7 ±2.8	8.2 ±1.6	54.0 ±4.1	27.1 ±1.6	71.1 ±2.1	23.6 ±1.1
	ProAgent	68.7 ±3.8	37.2 ±1.7	72.0 ±3.7	20.1 ±2.7	34.7 ±3.9	35.0 ±1.5	58.4 ±2.3	30.8 ±1.2
	HypMinds	66.0 ±3.9	39.5 ±1.7	62.0 ±4.0	26.6 ±2.9	47.3 ±4.1	30.9 ±1.5	58.4 ±2.3	32.3 ±1.2
	ToS (Ours)	**99.3** ±0.7	**15.4** ±1.3	**100.0** ±0.0	**0.0** ±0.0	**99.3** ±0.7	**9.1** ±0.9	**99.6** ±0.3	**8.2** ±0.6
		Fishery		Pasture		Pollution		Avg.	
GovSim	Method	Survival (%) ↑	Total Gain ↑	Survival (%) ↑	Total Gain ↑	Survival (%) ↑	Total Gain ↑	Survival (%) ↑	Total Gain ↑
	ReAct	46.5 ±4.6	226 ±20	21.3 ±1.9	129 ±4	11.7 ±1.0	109 ±3	26.5 ±2.1	155 ±8
	Focal	36.8 ±4.8	207 ±21	14.8 ±1.5	120 ±5	8.8 ±0.4	102 ±2	20.2 ±1.9	143 ±8
	CFD	84.2 ±3.3	525 ±23	49.7 ±3.8	**257** ±19	43.7 ±4.7	244 ±22	59.2 ±2.7	342 ±16
	Classic ToM	63.3 ±4.1	357 ±23	19.5 ±1.4	144 ±5	13.7 ±1.1	122 ±5	32.2 ±2.3	207 ±12
	ProAgent	29.2 ±3.8	143 ±8	11.8 ±0.8	107 ±2	8.3 ±0.0	100 ±0	16.4 ±1.5	117 ±3
	HypMinds	80.7 ±3.8	448 ±23	35.8 ±2.2	189 ±6	23.0 ±2.9	147 ±9	46.5 ±2.7	262 ±14
	ToS (Ours)	**99.0** ±0.4	**550** ±10	**72.0** ±4.8	254 ±17	**84.5** ±4.3	**397** ±21	**85.2** ±2.3	**400** ±14
		Throughput (dishes) ↑						Avg.	
Overcooked	Method	Level 1	Level 2	Level 3	Level 4	Level 5	Level 6	Raw ↑	×ReAct ↑
	ReAct	4.93 ±0.18	3.47 ±0.22	2.13 ±0.19	2.27 ±0.21	0.60 ±0.13	1.07 ±0.12	2.41 ±0.17	1.00 ±0.05
	Focal	5.40 ±0.25	3.07 ±0.25	2.07 ±0.18	2.00 ±0.17	1.20 ±0.14	0.87 ±0.13	2.43 ±0.18	1.11 ±0.07
	CFD	4.80 ±0.22	2.47 ±0.17	1.80 ±0.11	2.33 ±0.16	1.47 ±0.13	1.00 ±0.17	2.31 ±0.14	1.16 ±0.08
	Classic ToM	5.53 ±0.29	4.47 ±0.19	2.73 ±0.15	3.27 ±0.23	1.33 ±0.19	1.20 ±0.11	3.09 ±0.18	1.41 ±0.07
	ProAgent	6.00 ±0.58	3.73 ±0.33	2.87 ±0.19	**3.80** ±0.14	1.27 ±0.12	0.80 ±0.17	3.08 ±0.22	1.36 ±0.07
	HypMinds	4.67 ±0.58	5.00 ±0.35	3.07 ±0.18	3.27 ±0.21	1.47 ±0.17	**1.33** ±0.13	3.13 ±0.19	1.49 ±0.08
	ToS (Ours)	**6.47** ±0.39	**5.73** ±0.44	**3.40** ±0.24	3.53 ±0.27	**1.67** ±0.13	1.20 ±0.17	**3.67** ±0.24	**1.67** ±0.08

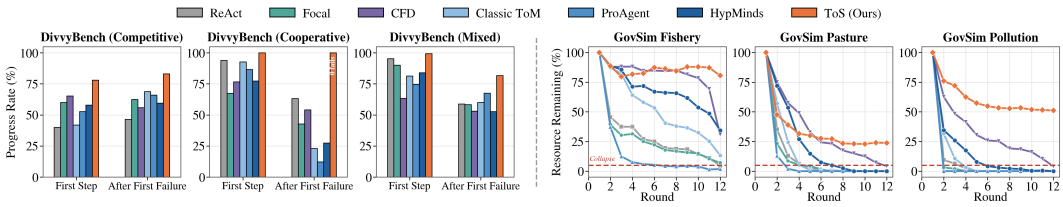

Figure 3: **ToS breaks symmetry on the first step, recovers after a failure, and sustains the resource pool.** *Left:* On DivvyBench, the progress rate at the first step and at the step after the first failure, in each setting. The progress rate is defined as the fraction of episodes in which the team takes at least one target on that step. *Right:* On GovSim, the resource remaining over the horizon.

fixed before the scene is read. At the step after the first failure, the methods differ only in what each infers from it. ToS leads at both steps, since a division derived from a scene that every agent reads alike fails only when an agent misreads the scene, whereas independent selections fail whenever they coincide. Over the 12 rounds of GovSim, the pool declines under every baseline, whereas ToS stabilizes it within a few rounds and ends each scenario with more resource than any baseline.

Ablation Study. Table 2 removes one reading at a time. Under *w/o* `task`, the agents cannot settle whether to converge on the same target or diverge onto different ones, so all three DivvyBench settings degrade, and GovSim loses the reading that marks the pool as rival, so nothing tells an agent that what it takes is taken from the rest. Overcooked alone barely changes, since its recipe states the ordering outright. Under *w/o* `role`, the pattern reverses and Overcooked drops, since it loses the reading that reports the work already divided, and the agents contend over stations the roles already assign. GovSim drops again by the wider margin, since role gating no longer reports the pool as contended, and the agents divide its full capacity, which exceeds the harvest the pool can sustain. By contrast, DivvyBench alone is unaffected, since the task coupling together with the public role (i.e., Leader or Follower) is enough to derive the division of labor. The task context and the public role settle a different part of the division on each benchmark, so either removal costs two of the three benchmarks, and only both readings together hold on all three.

Robustness Study. Table 2 also reorders the two readings and withholds the other agents' actions, and neither perturbation produces a significant drop. Reordering them under `task`→`role` does not

Table 2: **Ablation Study and Robustness Study.** Removing role gating or task coupling lowers performance, while reordering the two readings or withholding the other agents' actions does not. Red shaded cells are significantly worse than ToS (one-sided two-sample z-test, $p < 0.05$).

	DivvyBench - Success (%) ↑			**GovSim**		**Overcooked**
Method	Competitive	Cooperative	Mixed	Survival (%) ↑	Total Gain ↑	×ReAct ↑
ToS (Ours)	**99.3** ±0.7	**100.0** ±0.0	99.3 ±0.7	**85.2** ±2.3	400 ±14	**1.67** ±0.08
Ablation Study						
w/o `role`	97.3 ±1.3	**100.0** ±0.0	99.3 ±0.7	58.1 ±3.0	242 ±10	1.45 ±0.09
w/o `task`	83.3 ±3.1	76.0 ±3.5	18.0 ±3.1	67.9 ±3.0	344 ±17	1.61 ±0.07
Robustness Study						
`task`→`role`	**99.3** ±0.7	**100.0** ±0.0	**100.0** ±0.0	80.3 ±2.7	375 ±14	1.51 ±0.07
w/o $\mathbf{a}_{-i}^{<t}$	98.7 ±0.9	**100.0** ±0.0	**100.0** ±0.0	81.5 ±2.4	**407** ±14	1.54 ±0.08

change the inputs of either reading, since Equation (3) infers both from the observation o_i^t and role r_i alone. Withholding the other agents' actions under *w/o* $\mathbf{a}_{-i}^{<t}$ is the stronger test, since every ToM baseline depends on this input, and ToS therefore applies wherever those actions are unavailable.

ToM + ToS's Readings. Table 3 provides Classic ToM with the reading text of ToS, leaving the mechanism that derives the action as the only difference between the two schemas. ToS conditions the action on both readings (Algorithm 2, Line 4), whereas each variant best-responds to its prediction (Algorithm 1, Line 4), so that a reading reaches the action only through that prediction, with two consequences. *(i) ToM with both readings still fails to converge.* Adding both readings raises GovSim survival only to 39.1 against 85.2 for ToS and lowers DivvyBench Cooperative success below that of Classic ToM alone, the two cases in which every agent must select the same action. A prediction is specific to each agent and revised from its own history, so actions derived from predictions coincide only when the predictions themselves do, whereas the readings are common to all agents. The `role` field alone lowers Classic ToM on DivvyBench and GovSim, since it specifies no action on which a best response can operate. *(ii) Canonical order does not determine the action.* The `task` field provided to Classic ToM carries the same instruction as in ToS, to apply the rule of each coupling over a canonical order, so the variants derive that order in the same manner as ToS. Agents that derive the same order therefore select the same action under ToS, whereas under ToM they act on predictions that the order does not constrain and that need not coincide with it.

Table 3: **Classic ToM with ToS's `role` and `task` readings.**

	DivvyBench - Success (%) ↑		
Method	Competitive	Cooperative	Mixed
Classic ToM	72.7 ±3.7	86.7 ±2.8	54.0 ±4.1
+ `role`	64.7 ±3.9	80.0 ±3.3	47.3 ±4.1
+ `task`	91.3 ±2.3	68.7 ±3.8	82.7 ±3.1
+ `role`+`task`	88.7 ±2.6	68.7 ±3.8	78.7 ±3.4
ToS (Ours)	**99.3** ±0.7	**100.0** ±0.0	**99.3** ±0.7
	GovSim		**Overcooked**
	Survival (%) ↑	Total Gain ↑	×ReAct ↑
Classic ToM	32.2 ±2.3	207 ±12	1.41 ±0.07
+ `role`	31.2 ±2.3	191 ±10	1.50 ±0.09
+ `task`	31.7 ±2.5	211 ±13	1.63 ±0.08
+ `role`+`task`	39.1 ±2.8	247 ±15	1.66 ±0.08
ToS (Ours)	**85.2** ±2.3	**400** ±14	**1.67** ±0.08

Further Analysis. Appendix A covers the public role (A.1), observation asymmetry (A.2), the model backbone (A.3), heterogeneous models (A.4), heterogeneous reasoning schemas (A.5), the number of agents (A.6), the interaction horizon (A.7), ToM's failure mode (A.8), per-agent distributions (A.9), and confidence intervals and significance tests (A.10).

6 Conclusion

In this work, we identify the symmetry trap, in which homogeneous LLM agents acting concurrently without communication fail both to divide contested targets and to converge on shared ones. Existing approaches each handle only one side of the trap, since a public role or sampling randomness keeps the agents together where they must divide, whereas predicting one another separates them where they must converge. Theory of Scene (ToS) escapes the trap and remains effective in every setting of DivvyBench, GovSim, and Overcooked, without a predefined division of labor or training on the task. These results suggest that LLM agents coordinating without communication are better served by reasoning from the scene they share than by predicting one another.

REFERENCES

Sahar Abdelnabi, Amr Gomaa, Sarath Sivaprasad, Lea Schönherr, and Mario Fritz. Cooperation, Competition, and Maliciousness: LLM-Stakeholders Interactive Negotiation. *Advances in Neural Information Processing Systems*, 37:83548–83599, 2024.

John P Agapiou, Alexander Sasha Vezhnevets, Edgar A Duéñez-Guzmán, Jayd Matyas, Yiran Mao, Peter Sunehag, Raphael Köster, Udari Madhushani, Kavya Kopparapu, Ramona Comanescu, et al. Melting Pot 2.0. *arXiv preprint arXiv:2211.13746*, 2022.

Saaket Agashe, Yue Fan, Anthony Reyna, and Xin Eric Wang. LLM-Coordination: Evaluating and Analyzing Multi-agent Coordination Abilities in Large Language Models. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 8038–8057, 2025.

Ido Aharon, Emanuele La Malfa, Michael Wooldridge, and Sarit Kraus. Tacit Coordination of Large Language Models. *arXiv preprint arXiv:2601.22184*, 2026.

Dana Angluin. Local and Global Properties in Networks of Processors. In *Proceedings of the twelfth annual ACM symposium on Theory of computing*, pp. 82–93, 1980.

Ariel Flint Ashery, Luca Maria Aiello, and Andrea Baronchelli. Emergent Social Conventions and Collective Bias in LLM Populations. *Science Advances*, 11(20):eadu9368, 2025.

Micah Carroll, Rohin Shah, Mark K Ho, Tom Griffiths, Sanjit Seshia, Pieter Abbeel, and Anca Dragan. On the Utility of Learning about Humans for Human-AI Coordination. *Advances in Neural Information Processing Systems*, 32, 2019.

Ayush Chopra, Aman Sharma, Feroz Ahmad, Luca Muscariello, Vijoy Pandey, and Ramesh Raskar. Ripple Effect Protocol: Coordinating Agent Populations. *arXiv preprint arXiv:2510.16572*, 2025.

Logan Cross, Violet Xiang, Agam Bhatia, Daniel Yamins, and Nick Haber. Hypothetical Minds: Scaffolding Theory of Mind for Multi-Agent Tasks with Large Language Models. In *International Conference on Learning Representations*, pp. 6507–6546, 2025.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B Tenenbaum, and Igor Mordatch. Improving Factuality and Reasoning in Language Models through Multiagent Debate. In *International Conference on Machine Learning*, pp. 11733–11763. PMLR, 2024.

Jinhao Duan, Renming Zhang, James Diffenderfer, Bhavya Kailkhura, Lichao Sun, Elias Stengel-Eskin, Mohit Bansal, Tianlong Chen, and Kaidi Xu. GTBench: Uncovering the Strategic Reasoning Capabilities of LLMs via Game-Theoretic Evaluations. *Advances in Neural Information Processing Systems*, 37:28219–28253, 2024.

Shashwat Goel, Joschka Strüber, Ilze Amanda Auzina, Karuna K Chandra, Ponnurangam Kumaraguru, Douwe Kiela, Ameya Prabhu, Matthias Bethge, and Jonas Geiping. Great Models Think Alike and this Undermines AI Oversight. In *International Conference on Machine Learning*, pp. 19621–19678, 2025.

Florian Grötschla, Luis Müller, Jan Tönshoff, Mikhail Galkin, and Bryan Perozzi. AgentsNet: Coordination and Collaborative Reasoning in Multi-Agent LLMs. *arXiv preprint arXiv:2507.08616*, 2025.

Adrian Hayler, Shashank Reddy Chirra, Andrei Lupu, Johannes Forkel, Bidipta Sarkar, Siheng Feng, and Jakob Nicolaus Foerster. Zero-Shot Coordination among LLM Agents. In *Workshop on Multi-Agent Learning and Its Opportunities in the Era of Generative AI*, 2026.

Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Steven Yau, Zijuan Lin, Liyang Zhou, et al. MetaGPT: Meta Programming for A Multi-Agent Collaborative Framework. In *International Conference on Learning Representations*, pp. 23247–23275, 2024.

Hengyuan Hu, Adam Lerer, Alex Peysakhovich, and Jakob Foerster. "Other-Play" for Zero-Shot Coordination. In *International Conference on Machine Learning*, pp. 4399–4410. PMLR, 2020.

Hengyuan Hu, Adam Lerer, Brandon Cui, Luis Pineda, Noam Brown, and Jakob Foerster. Off-Belief Learning. In *International Conference on Machine Learning*, pp. 4369–4379. PMLR, 2021.

HuaDong Jian, Chenghao Li, Haoyu Wang, Jiajia Shuai, Jinyu Guo, Yang Yang, and Chaoning Zhang. Gated Coordination for Efficient Multi-Agent Collaboration in Minecraft Game. *arXiv preprint arXiv:2604.18975*, 2026.

Liwei Jiang, Yuanjun Chai, Margaret Li, Mickel Liu, Raymond Fok, Nouha Dziri, Yulia Tsvetkov, Maarten Sap, and Yejin Choi. Artificial Hivemind: The Open-Ended Homogeneity of Language Models (and Beyond). *Advances in Neural Information Processing Systems*, 38, 2025.

Elliot Kim, Avi Garg, Kenny Peng, and Nikhil Garg. Correlated Errors in Large Language Models. In *International Conference on Machine Learning*, pp. 30038–30066, 2025.

Yubin Kim, Chanwoo Park, Taehan Kim, Eugene Park, Samuel Schmidgall, Salman Rahman, Chunjong Park, Cynthia Breazeal, Xin Liu, Hamid Palangi, et al. TeamBench: Evaluating Agent Coordination under Enforced Role Separation. *arXiv preprint arXiv:2605.07073*, 2026.

Guohao Li, Hasan Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. CAMEL: Communicative Agents for "Mind" Exploration of Large Language Model Society. *Advances in Neural Information Processing Systems*, 36:51991–52008, 2023a.

Huao Li, Yu Chong, Simon Stepputtis, Joseph P Campbell, Dana Hughes, Charles Lewis, and Katia Sycara. Theory of Mind for Multi-Agent Collaboration via Large Language Models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 180–192, 2023b.

Yuxuan Li, Aoi Naito, and Hirokazu Shirado. Systematic Failures in Collective Reasoning under Distributed Information in Multi-Agent LLMs. *arXiv preprint arXiv:2505.11556*, 2025.

Andrei Lupu, Timon Willi, and Jakob Foerster. The Decrypto Benchmark for Multi-Agent Reasoning and Theory of Mind. *arXiv preprint arXiv:2506.20664*, 2025.

Chunjiang Mu, Ya Zeng, Qiaosheng Zhang, Kun Shao, Chen Chu, Hao Guo, Danyang Jia, Zhen Wang, and Shuyue Hu. Adaptive Theory of Mind for LLM-based Multi-Agent Coordination. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 29608–29616, 2026.

Georgios Papoudakis, Filippos Christianos, Lukas Schäfer, and Stefano V Albrecht. Benchmarking Multi-Agent Deep Reinforcement Learning Algorithms in Cooperative Tasks. *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, 2021.

Rohan Patil, Jai Malegaonkar, and Henrik I Christensen. Randomness is Sometimes Necessary for Coordination. *arXiv preprint arXiv:2605.06825*, 2026.

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. *Advances in Neural Information Processing Systems*, 37:111715–111759, 2024.

Chen Qian, Wei Liu, Hongzhang Liu, Nuo Chen, Yufan Dang, Jiahao Li, Cheng Yang, Weize Chen, Yusheng Su, Xin Cong, et al. ChatDev: Communicative Agents for Software Development. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15174–15186, 2024.

Hong Qian, Yuanhao Liu, Zihan Zhou, Zongbao Zhang, Hanjie Ge, Haotian Shi, Liang Dou, Xiangfeng Wang, Jing-Wen Yang, and Aimin Zhou. CollabBench: Benchmarking and Unleashing Collaborative Ability of LLMs with Diverse Players via Proactive Engagement. In *International Conference on Machine Learning*, 2026.

Qwen Team. Qwen3.5: Towards Native Multimodal Agents, February 2026. URL `https://qwen.ai/blog?id=qwen3.5`.

Christoph Riedl. Emergent Coordination in Multi-Agent Language Models. In *International Conference on Learning Representations*, pp. 120776–120799, 2026.

Thomas C Schelling. *The Strategy of Conflict: with a new Preface by the Author*. Harvard University Press, 1980.

Haochen Sun, Shuwen Zhang, Lujie Niu, Lei Ren, Hao Xu, Hao Fu, Fangkun Zhao, Caixia Yuan, and Xiaojie Wang. Collab-Overcooked: Benchmarking and Evaluating Large Language Models as Collaborative Agents. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 4922–4951, 2025.

Kale-ab Abebe Tessera, Andras Szecsenyi, Cameron Barker, Alexander Rutherford, Davide Paglieri, Aidan Scannell, Henry Gouk, Elliot J Crowley, Tim Rocktäschel, and Amos Storkey. Benchmarking Open-Ended Multi-Agent Coordination in Language Agents. *arXiv preprint arXiv:2606.08340*, 2026.

Yuanfei Wang, Fangwei Zhong, Jing Xu, and Yizhou Wang. ToM2C: Target-oriented Multi-agent Communication and Cooperation with Theory of Mind. In *International Conference on Learning Representations*, 2022.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, et al. AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation. In *First Conference on Language Modeling*, 2024.

Sarah A Wu, Rose E Wang, James A Evans, Joshua B Tenenbaum, David C Parkes, and Max Kleiman-Weiner. Too many cooks: Bayesian inference for coordinating multi-agent collaboration. *Topics in Cognitive Science*, 13(2):414–432, 2021.

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. *Advances in Neural Information Processing Systems*, 38:107309–107336, 2025.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing Reasoning and Acting in Language Models. In *International Conference on Learning Representations*, 2023.

Ceyao Zhang, Kaijie Yang, Siyi Hu, Zihao Wang, Guanghe Li, Yihang Sun, Cheng Zhang, Zhaowei Zhang, Anji Liu, Song-Chun Zhu, et al. ProAgent: Building Proactive Cooperative Agents with Large Language Models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 17591–17599, 2024a.

Hongxin Zhang, Weihua Du, Jiaming Shan, Qinhong Zhou, Yilun Du, Joshua B Tenenbaum, Tianmin Shu, and Chuang Gan. Building Cooperative Embodied Agents Modularly with Large Language Models. In *International Conference on Learning Representations*, pp. 19373–19401, 2024b.

Appendix

A Further Analysis

A.1 Impact of the Public Role

Public Role. Table 4 holds the ToS reasoning fixed and changes only the public role r_i the agents are given, from an identical description through an index to the leader and follower labels ToS uses with two agents. At temperature 0, where the wording of r_i is the only thing that can separate the agents, identical descriptions leave ToS at 0.7% in DivvyBench Competitive, the outcome classical symmetry breaking predicts for identical deterministic agents (Angluin, 1980), so the trap survives an arbitrarily good reasoning schema once nothing distinguishes the agents. DivvyBench Cooperative is untouched at 100% with no stuck steps, since converging on one target requires only agreement, and the distinction is therefore needed exactly where the agents must differ. An index already recovers most of the success in DivvyBench Competitive (92.7%), and the leader and follower labels carry the same 1 bit yet close the remainder and cut the stuck rate in DivvyBench Mixed from 9.8% to 6.4%, so equal information does not yield equal coordination. Beyond two agents, every method uses index labels, and Appendix A.6 still finds ToS near perfect at every team size.

Sampling Temperature. Table 4 also sweeps the sampling temperature against each public role, the other candidate for breaking the symmetry (Patil et al., 2026). Temperature leaves the identical description near the floor, at 11.3% in DivvyBench Competitive at the highest temperature, partly compensates for a weak distinction, carrying the index from 92.7% to 99.3%, and adds nothing once the bit is a leader and follower label, where temperature 0 already completes all three DivvyBench settings and gives the lowest stuck rate of all. Randomness therefore substitutes for the wording of a distinction and never for the distinction itself.

Table 4: **One bit of distinction suffices to break the symmetry.** The ToS reasoning is held fixed on DivvyBench and only the public role differs. Under None the agents receive the same wording, under 1 Bit (index) they are told apart by "A0" and "A1", and under 1 Bit (role) by the leader and follower labels ToS uses, the last two carrying the same 1 bit. Red marks a cell significantly worse than 1 Bit (role) in the same setting at the same temperature (one-sided two-sample z-test, $p < 0.05$).

		Competitive		Cooperative		Mixed	
Temp	Public Role	Success (%) ↑	Stuck (%) ↓	Success (%) ↑	Stuck (%) ↓	Success (%) ↑	Stuck (%) ↓
0.0	None	0.7 ±0.7	80.5 ±1.8	100.0 ±0.0	0.0 ±0.0	8.0 ±2.2	48.9 ±1.2
	1 Bit (index)	92.7 ±2.1	24.4 ±1.9	100.0 ±0.0	0.0 ±0.0	99.3 ±0.7	9.8 ±0.9
	1 Bit (role)	100.0 ±0.0	9.3 ±1.0	100.0 ±0.0	0.0 ±0.0	100.0 ±0.0	6.4 ±0.8
0.7	None	12.7 ±2.7	53.1 ±1.4	100.0 ±0.0	0.2 ±0.1	28.0 ±3.7	33.0 ±1.1
	1 Bit (index)	99.3 ±0.7	19.5 ±1.3	100.0 ±0.0	0.2 ±0.1	98.7 ±0.9	13.7 ±1.0
	1 Bit (role)	99.3 ±0.7	15.4 ±1.3	100.0 ±0.0	0.0 ±0.0	99.3 ±0.7	9.1 ±0.9
1.0	None	11.3 ±2.6	51.5 ±1.3	100.0 ±0.0	0.5 ±0.2	22.7 ±3.4	34.0 ±1.0
	1 Bit (index)	99.3 ±0.7	20.4 ±1.4	100.0 ±0.0	0.5 ±0.2	98.7 ±0.9	13.7 ±1.0
	1 Bit (role)	99.3 ±0.7	15.1 ±1.3	100.0 ±0.0	0.2 ±0.1	98.0 ±1.1	11.4 ±0.9

A.2 Impact of Observation Asymmetry

Observation Asymmetry. Table 5 holds the scene fixed and changes only the order in which each agent is shown it. An observation that differs is a third source of asymmetry between homogeneous agents, after the public role of Appendix A.1 and the temperature of Section 3, and like the temperature it raises divergence and lowers convergence. Each baseline selects the most salient target of its own listing, and those listings no longer agree, so DivvyBench Competitive success rises. Converging instead requires the agents to select one target, and each selects the head of its own listing, which under a shared order had been the same target for all, so every baseline collapses in DivvyBench Cooperative. ReAct marks both extremes, rising in DivvyBench Competitive, where the agents must divide, and collapsing where they must meet, so the method that gains most from the asymmetry in one setting loses most in the other. DivvyBench Mixed puts both demands in one episode and leaves every baseline collapsed, so an asymmetry in what the agents observe resolves one side of the trap and never both.

ToS Derives Canonical Orders. Table 5 also shows that ToS is the only method that succeeds in all three settings. A scene admits one or more canonical orders over its targets, from their names, from an index, or from any other property they carry, and ToS derives one and divides it by role. One derived order serves both demands, since converging takes its first element and diverging splits it among the roles, and the agents therefore reach the same division from listings that agree nowhere. Two candidate orders are available, the one a listing presents and the one the names induce, and only Tabletop separates them, since its colors are listed out of alphabetical order while the grid references and the location names are already sorted. Given one shared listing, the agents follow the order it presents in 93.8% of their decisions, and under a reversed or shuffled listing they sort the names instead, in 61.7% and 65.2% of their decisions. Every agent can construct a sorted order for itself, whereas the presented order is no longer shared once the perturbation withholds it. The disagreements that remain fall mostly on the first step, where the agents hold the scene and nothing else, and the steps that follow realign them, since a wasted step is visible in the observation and the next action is sampled independently of the last.

Table 5: **Only ToS succeeds in every setting under disagreeing observations.** Under Shared, the agents are shown the targets in the presentation order, as in Table 1. Under Reversed and Shuffled, A1 is shown them in reverse or in a fixed derangement of that order, with the targets and their types unchanged, and the rules state that the orders may differ. Red marks a setting whose Success is 20% or below.

		Competitive		Cooperative		Mixed	
	Method	Success (%) ↑	Stuck (%) ↓	Success (%) ↑	Stuck (%) ↓	Success (%) ↑	Stuck (%) ↓
Shared	ReAct	40.7 ±4.0	48.7 ±1.4	96.7 ±1.5	2.5 ±0.7	56.7 ±4.1	26.9 ±1.1
	Focal	47.3 ±4.1	45.4 ±1.4	42.0 ±4.0	36.6 ±2.4	50.0 ±4.1	29.0 ±1.1
	CFD	57.3 ±4.1	31.8 ±1.8	66.7 ±3.9	20.5 ±2.1	36.0 ±3.9	33.9 ±1.7
	Classic ToM	72.7 ±3.7	35.6 ±1.5	86.7 ±2.8	8.2 ±1.6	54.0 ±4.1	27.1 ±1.6
	ProAgent	68.7 ±3.8	37.2 ±1.7	72.0 ±3.7	20.1 ±2.7	34.7 ±3.9	35.0 ±1.5
	HypMinds	66.0 ±3.9	39.5 ±1.7	62.0 ±4.0	26.6 ±2.9	47.3 ±4.1	30.9 ±1.5
	ToS (Ours)	**99.3** ±0.7	**15.4** ±1.3	**100.0** ±0.0	**0.0** ±0.0	**99.3** ±0.7	**9.1** ±0.9
Reversed	ReAct	94.7 ±1.8	**12.7** ±1.5	0.0 ±0.0	78.4 ±1.2	5.3 ±1.8	55.4 ±1.0
	Focal	67.3 ±3.8	32.9 ±1.5	0.7 ±0.7	75.5 ±1.2	12.0 ±2.7	51.0 ±1.3
	CFD	62.0 ±4.0	31.3 ±1.4	4.7 ±1.7	68.2 ±1.8	3.3 ±1.5	62.8 ±1.3
	Classic ToM	85.3 ±2.9	25.3 ±1.6	1.3 ±0.9	80.0 ±1.3	2.0 ±1.1	58.9 ±1.0
	ProAgent	77.3 ±3.4	28.6 ±1.8	1.3 ±0.9	84.5 ±1.3	7.3 ±2.1	55.1 ±1.3
	HypMinds	89.3 ±2.5	19.5 ±1.6	4.0 ±1.6	79.5 ±1.8	4.0 ±1.6	57.1 ±1.0
	ToS (Ours)	**99.3** ±0.7	16.2 ±1.3	**93.3** ±2.0	**7.1** ±0.9	**83.3** ±3.1	**23.9** ±1.6
Shuffled	ReAct	92.0 ±2.2	20.8 ±1.7	0.7 ±0.7	77.1 ±1.3	4.0 ±1.6	61.5 ±1.2
	Focal	61.3 ±4.0	34.1 ±1.5	20.0 ±3.3	61.2 ±2.4	6.7 ±2.0	59.2 ±1.3
	CFD	66.0 ±3.9	28.5 ±1.4	18.0 ±3.1	60.1 ±2.5	2.0 ±1.1	64.8 ±1.2
	Classic ToM	92.0 ±2.2	20.7 ±1.5	0.0 ±0.0	84.6 ±0.9	4.0 ±1.6	56.9 ±1.1
	ProAgent	69.3 ±3.8	32.8 ±1.9	0.0 ±0.0	86.9 ±0.9	6.7 ±2.0	59.2 ±1.4
	HypMinds	86.0 ±2.8	26.1 ±1.7	2.7 ±1.3	82.5 ±1.4	4.7 ±1.7	57.1 ±1.1
	ToS (Ours)	**99.3** ±0.7	**15.8** ±1.3	**92.0** ±2.2	**8.1** ±0.8	**80.7** ±3.2	**26.3** ±1.8

A.3 Impact of the Model Backbone

Table 6 rebuilds the agents on three further backbones, spanning three families from 9B to 26B, and repeats the three DivvyBench settings on each in every scenario. ToS is near-perfect on all four and holds the highest average on each, while baseline success swings sharply from one backbone to the next. The cause is that no baseline breaks symmetry explicitly, so whether homogeneous agents converge or diverge is set by the backbone, whatever the task requires. The prediction-free ReAct and CFD read that decision off the backbone's raw decoding tendency, and the prediction-based Classic ToM, ProAgent, and HypMinds reach the same outcome through prediction, since predicting an agent that reasons alike returns the predictor's own tendency. The ranking among the baselines changes with the backbone, and HypMinds ranks first on Gemma-4-26B and fifth on GPT-OSS-20B. ToS derives its decision from the scene, an input that does not vary with the backbone.

Table 6: **ToS's advantage holds across model backbones.** Three further backbones (Gemma-4-26B, GPT-OSS-20B, and Qwen3.5-9B) stand in for the main Qwen3.5-35B-A3B across the three DivvyBench settings, with all agents in a run sharing one backbone.

		Competitive		Cooperative		Mixed		Avg.	
	Method	Success (%) ↑	Stuck (%) ↓	Success (%) ↑	Stuck (%) ↓	Success (%) ↑	Stuck (%) ↓	Success (%) ↑	Stuck (%) ↓
Qwen3.5-35B	ReAct	40.7 ±4.0	48.7 ±1.4	96.7 ±1.5	2.5 ±0.7	56.7 ±4.1	26.9 ±1.1	64.7 ±2.3	26.0 ±1.1
	Focal	47.3 ±4.1	45.4 ±1.4	42.0 ±4.0	36.6 ±2.4	50.0 ±4.1	29.0 ±1.1	46.4 ±2.4	37.0 ±1.1
	CFD	57.3 ±4.1	31.8 ±1.8	66.7 ±3.9	20.5 ±2.1	36.0 ±3.9	33.9 ±1.7	53.3 ±2.4	28.7 ±1.1
	Classic ToM	72.7 ±3.7	35.6 ±1.5	86.7 ±2.8	8.2 ±1.6	54.0 ±4.1	27.1 ±1.6	71.1 ±2.1	23.6 ±1.1
	ProAgent	68.7 ±3.8	37.2 ±1.7	72.0 ±3.7	20.1 ±2.7	34.7 ±3.9	35.0 ±1.5	58.4 ±2.3	30.8 ±1.2
	HypMinds	66.0 ±3.9	39.5 ±1.7	62.0 ±4.0	26.6 ±2.9	47.3 ±4.1	30.9 ±1.5	58.4 ±2.3	32.3 ±1.2
	ToS (Ours)	**99.3** ±0.7	**15.4** ±1.3	**100.0** ±0.0	**0.0** ±0.0	**99.3** ±0.7	**9.1** ±0.9	**99.6** ±0.3	**8.2** ±0.6
Gemma-4-26B	ReAct	48.7 ±4.1	51.4 ±2.2	**100.0** ±0.0	0.7 ±0.2	45.3 ±4.1	34.3 ±1.3	64.7 ±2.3	28.8 ±1.3
	Focal	34.0 ±3.9	62.0 ±1.9	86.7 ±2.8	9.5 ±1.2	30.0 ±3.8	40.7 ±1.0	50.2 ±2.4	37.4 ±1.3
	CFD	94.0 ±1.9	27.7 ±1.8	91.3 ±2.3	4.9 ±1.0	61.3 ±4.0	28.4 ±1.5	82.2 ±1.8	20.3 ±1.0
	Classic ToM	66.7 ±3.9	47.5 ±2.1	**100.0** ±0.0	0.1 ±0.1	67.3 ±3.8	25.6 ±1.6	78.0 ±2.0	24.4 ±1.3
	ProAgent	62.7 ±4.0	51.8 ±1.8	**100.0** ±0.0	**0.0** ±0.0	62.7 ±4.0	29.5 ±1.4	75.1 ±2.0	27.1 ±1.3
	HypMinds	75.3 ±3.5	45.2 ±1.8	96.7 ±1.5	2.2 ±0.9	82.7 ±3.1	24.2 ±1.5	84.9 ±1.7	23.8 ±1.2
	ToS (Ours)	**100.0** ±0.0	**10.0** ±0.9	**100.0** ±0.0	**0.0** ±0.0	**100.0** ±0.0	**2.3** ±0.5	**100.0** ±0.0	**4.1** ±0.4
GPT-OSS-20B	ReAct	92.7 ±2.1	19.8 ±1.5	83.3 ±3.1	9.2 ±1.6	77.3 ±3.4	17.7 ±1.4	84.4 ±1.7	15.5 ±0.9
	Focal	72.7 ±3.7	38.0 ±1.4	10.7 ±2.5	57.6 ±1.9	39.3 ±4.0	35.3 ±1.5	40.9 ±2.3	43.6 ±1.0
	CFD	87.3 ±2.7	22.1 ±1.4	56.0 ±4.1	25.5 ±1.9	62.0 ±4.0	23.3 ±1.3	68.4 ±2.2	23.6 ±0.9
	Classic ToM	86.7 ±2.8	25.1 ±1.8	93.3 ±2.0	3.9 ±1.2	68.0 ±3.8	21.8 ±1.6	82.7 ±1.8	16.9 ±1.0
	ProAgent	81.3 ±3.2	34.1 ±1.7	74.0 ±3.6	16.6 ±2.3	55.3 ±4.1	27.7 ±1.5	70.2 ±2.2	26.1 ±1.1
	HypMinds	85.3 ±2.9	30.7 ±1.7	64.0 ±3.9	23.4 ±2.6	40.7 ±4.0	33.7 ±1.6	63.3 ±2.3	29.3 ±1.2
	ToS (Ours)	**100.0** ±0.0	**9.8** ±1.1	**100.0** ±0.0	**0.2** ±0.1	**99.3** ±0.7	**5.9** ±0.8	**99.8** ±0.2	**5.3** ±0.5
Qwen3.5-9B	ReAct	45.3 ±4.1	43.1 ±1.5	70.7 ±3.7	18.3 ±2.2	18.0 ±3.1	43.2 ±1.4	44.7 ±2.3	34.9 ±1.1
	Focal	63.3 ±3.9	32.1 ±1.5	34.7 ±3.9	40.8 ±2.6	13.3 ±2.8	50.0 ±1.5	37.1 ±2.3	40.9 ±1.1
	CFD	82.0 ±3.1	25.8 ±1.4	39.3 ±4.0	35.1 ±2.2	12.0 ±2.7	50.7 ±1.6	44.4 ±2.3	37.2 ±1.1
	Classic ToM	60.0 ±4.0	35.5 ±1.3	71.3 ±3.7	18.3 ±2.3	18.7 ±3.2	46.7 ±1.6	50.0 ±2.4	33.5 ±1.2
	ProAgent	75.3 ±3.5	31.8 ±1.4	46.7 ±4.1	36.6 ±2.9	19.3 ±3.2	45.7 ±1.6	47.1 ±2.4	38.0 ±1.2
	HypMinds	82.7 ±3.1	32.0 ±1.4	37.3 ±4.0	45.0 ±2.9	28.0 ±3.7	39.9 ±1.5	49.3 ±2.4	39.0 ±1.2
	ToS (Ours)	**98.0** ±1.1	**24.0** ±1.3	**100.0** ±0.0	**0.1** ±0.1	**96.0** ±1.6	**14.8** ±1.0	**98.0** ±0.7	**13.0** ±0.7

A.4 Impact of Heterogeneous Models

Table 7 gives the other agent a different backbone, the different-family Gemma-4-26B and the weaker same-family Qwen3.5-9B. Heterogeneity is the condition Theory of Mind is built for, since a prediction of an agent that reasons and decodes alike returns the predictor's own action, and a different backbone removes that. Every baseline gains against Gemma-4-26B, most of all CFD, which under homogeneity rejects every flippable convention until a second backbone breaks the deadlock for it. Classic ToM becomes the strongest baseline in both pairings, and it still trails ToS by 16.2 points against Gemma-4-26B and by 32.0 against Qwen3.5-9B. The weaker backbone separates the two mechanisms, since it breaks the symmetry just as freely and yet no baseline rises significantly above its homogeneous score, so a different agent improves ToM only when that agent is also a capable one. ToS's success is unchanged in both pairings, because the convention it commits to is read from the shared scene and the other agent contributes only its ability to read that scene and apply the same rule. Table 8 runs the same check on the reasoning schema, pairing each method with an agent that runs a different one on the same backbone.

Table 7: **ToS's advantage holds when the agents run different model backbones.** The agents run the method named in each row on DivvyBench, A0 on the main Qwen3.5-35B-A3B backbone and A1 on the backbone named in each panel.

		Competitive		Cooperative		Mixed		Avg.	
A1	Method	Success (%) ↑	Stuck (%) ↓	Success (%) ↑	Stuck (%) ↓	Success (%) ↑	Stuck (%) ↓	Success (%) ↑	Stuck (%) ↓
Gemma-4-26B	ReAct	53.3 ±4.1	44.2 ±1.8	98.7 ±0.9	1.4 ±0.5	52.7 ±4.1	29.3 ±1.0	68.2 ±2.2	25.0 ±1.1
	Focal	65.3 ±3.9	37.0 ±1.4	58.0 ±4.0	24.7 ±2.0	61.3 ±4.0	27.9 ±1.0	61.6 ±2.3	29.8 ±0.9
	CFD	86.7 ±2.8	19.0 ±1.3	77.3 ±3.4	13.5 ±1.7	82.7 ±3.1	15.1 ±1.0	82.2 ±1.8	15.9 ±0.8
	Classic ToM	78.7 ±3.4	36.4 ±1.3	97.3 ±1.3	1.6 ±0.6	75.3 ±3.5	21.0 ±1.4	83.8 ±1.7	19.7 ±0.9
	ProAgent	84.7 ±3.0	37.1 ±1.6	90.7 ±2.4	6.2 ±1.1	62.7 ±4.0	26.6 ±1.5	79.3 ±1.9	23.3 ±1.0
	HypMinds	82.7 ±3.1	34.9 ±1.6	75.3 ±3.5	16.7 ±2.3	66.0 ±3.9	24.6 ±1.5	74.7 ±2.1	25.4 ±1.1
	ToS (Ours)	**100.0** ±0.0	**10.5** ±1.0	**100.0** ±0.0	**0.0** ±0.0	**100.0** ±0.0	**2.8** ±0.5	**100.0** ±0.0	**4.5** ±0.4
Qwen3.5-9B	ReAct	50.7 ±4.1	40.2 ±1.4	78.0 ±3.4	12.1 ±1.7	28.7 ±3.7	37.1 ±1.3	52.4 ±2.4	29.8 ±1.0
	Focal	58.0 ±4.0	34.4 ±1.4	44.0 ±4.1	39.8 ±2.7	23.3 ±3.5	44.1 ±1.5	41.8 ±2.3	39.4 ±1.2
	CFD	60.7 ±4.0	30.8 ±1.4	49.3 ±4.1	33.2 ±2.4	16.7 ±3.1	47.1 ±1.6	42.2 ±2.3	37.0 ±1.1
	Classic ToM	81.3 ±3.2	28.1 ±1.4	81.3 ±3.2	12.0 ±1.9	36.0 ±3.9	38.1 ±1.8	66.2 ±2.2	26.1 ±1.1
	ProAgent	80.7 ±3.2	31.6 ±1.4	62.7 ±4.0	27.1 ±2.8	31.3 ±3.8	39.5 ±1.6	58.2 ±2.3	32.7 ±1.2
	HypMinds	89.3 ±2.5	29.8 ±1.5	54.7 ±4.1	32.5 ±3.0	40.7 ±4.0	34.9 ±1.6	61.6 ±2.3	32.4 ±1.2
	ToS (Ours)	**98.0** ±1.1	**27.2** ±1.4	**100.0** ±0.0	**0.1** ±0.1	**96.7** ±1.5	**13.2** ±1.0	**98.2** ±0.6	**13.5** ±0.8

A.5 Impact of Heterogeneous Reasoning Schemas

Table 8 pairs each method with an agent running ReAct or Classic ToM. Every method coordinates with an agent that reasons differently, and every baseline scores above its homogeneous result in Table 1 beside an unlike agent, least for Classic ToM. The gain comes from the pairing, since no method was changed and each was paired with the same two agents. Agents running unlike schemas are no longer homogeneous, so in DivvyBench Competitive the pairing supplies the distinction each baseline failed to derive. DivvyBench Cooperative reverses only for ReAct, the one baseline that converges reliably against a copy of itself, and its score there falls beside an unlike schema. The others converge less reliably under homogeneity, so an unlike partner does not lower their Cooperative success. Even with the symmetry broken for them, no baseline reaches ToS beside the same agent, and none reaches homogeneous ToS.

Table 8: **ToS achieves the highest success in every setting beside an agent that reasons differently.** A0 runs the method named in each row, and A1 runs the reasoning schema named in each panel.

		Competitive		Cooperative		Mixed		Avg.	
A1	A0 Method	Success (%) ↑	Stuck (%) ↓	Success (%) ↑	Stuck (%) ↓	Success (%) ↑	Stuck (%) ↓	Success (%) ↑	Stuck (%) ↓
ReAct	ReAct	40.7 ±4.0	48.7 ±1.4	96.7 ±1.5	2.5 ±0.7	56.7 ±4.1	26.9 ±1.1	64.7 ±2.3	26.0 ±1.1
	Focal	63.3 ±3.9	38.5 ±1.6	56.0 ±4.1	26.1 ±2.2	50.7 ±4.1	28.3 ±1.1	56.7 ±2.3	30.9 ±1.0
	CFD	60.0 ±4.0	31.4 ±1.4	77.3 ±3.4	14.3 ±1.9	66.7 ±3.9	18.7 ±1.2	68.0 ±2.2	21.5 ±0.9
	Classic ToM	74.0 ±3.6	29.8 ±1.7	94.7 ±1.8	4.5 ±1.2	73.3 ±3.6	21.1 ±1.3	80.7 ±1.9	**18.5** ±1.0
	ProAgent	82.0 ±3.1	31.6 ±1.6	85.3 ±2.9	9.8 ±1.7	66.0 ±3.9	23.8 ±1.2	77.8 ±2.0	21.7 ±1.0
	HypMinds	88.7 ±2.6	**27.0** ±1.5	82.7 ±3.1	10.8 ±1.8	68.0 ±3.8	23.5 ±1.3	79.8 ±1.9	20.4 ±0.9
	ToS (Ours)	**100.0** ±0.0	36.2 ±1.0	**98.0** ±1.1	**1.7** ±0.6	**94.0** ±1.9	**18.1** ±0.8	**97.3** ±0.8	18.7 ±0.8
Classic ToM	ReAct	97.3 ±1.3	8.2 ±1.2	88.7 ±2.6	6.2 ±1.3	81.3 ±3.2	13.5 ±1.2	89.1 ±1.5	9.3 ±0.7
	Focal	92.0 ±2.2	15.0 ±1.4	51.3 ±4.1	34.1 ±2.8	81.3 ±3.2	15.2 ±1.3	74.9 ±2.0	21.5 ±1.2
	CFD	84.7 ±3.0	16.6 ±1.5	68.0 ±3.8	18.7 ±2.1	72.0 ±3.7	18.5 ±1.3	74.9 ±2.0	17.9 ±1.0
	Classic ToM	72.7 ±3.7	35.6 ±1.5	86.7 ±2.8	8.2 ±1.6	54.0 ±4.1	27.1 ±1.6	71.1 ±2.1	23.6 ±1.1
	ProAgent	83.3 ±3.1	30.2 ±1.6	78.0 ±3.4	14.4 ±2.1	48.0 ±4.1	31.7 ±1.6	69.8 ±2.2	25.4 ±1.1
	HypMinds	75.3 ±3.5	35.9 ±1.5	77.3 ±3.4	14.0 ±2.1	57.3 ±4.1	28.9 ±1.5	70.0 ±2.2	26.3 ±1.1
	ToS (Ours)	**100.0** ±0.0	**7.9** ±1.0	**98.0** ±1.1	**1.9** ±0.5	**97.3** ±1.3	**11.8** ±1.0	**98.4** ±0.6	**7.2** ±0.5

A.6 Impact of the Number of Agents

Scaling the DivvyBench Team. Figure 4 grows the DivvyBench team from 2 agents to 5. DivvyBench Competitive absorbs the extra agents, since a collision wastes the step only for the agents in it while the rest still take targets, and the baselines show no common trend there. DivvyBench Cooperative and Mixed run the other way, since a Cooperative target is taken only when all N agents select it in the same step and one deviating agent voids the round. If each agent independently selects the shared target with probability $p < 1$, the round succeeds with probability about p^N, which decays as the group grows however close p is to one. Every baseline declines as the team grows, whatever its schema, since none of them makes the shared target certain. ToS reads the division from the shared scene, an input every agent holds, so p remains near one at every team size, and ToS stays near perfect in all 12 panels.

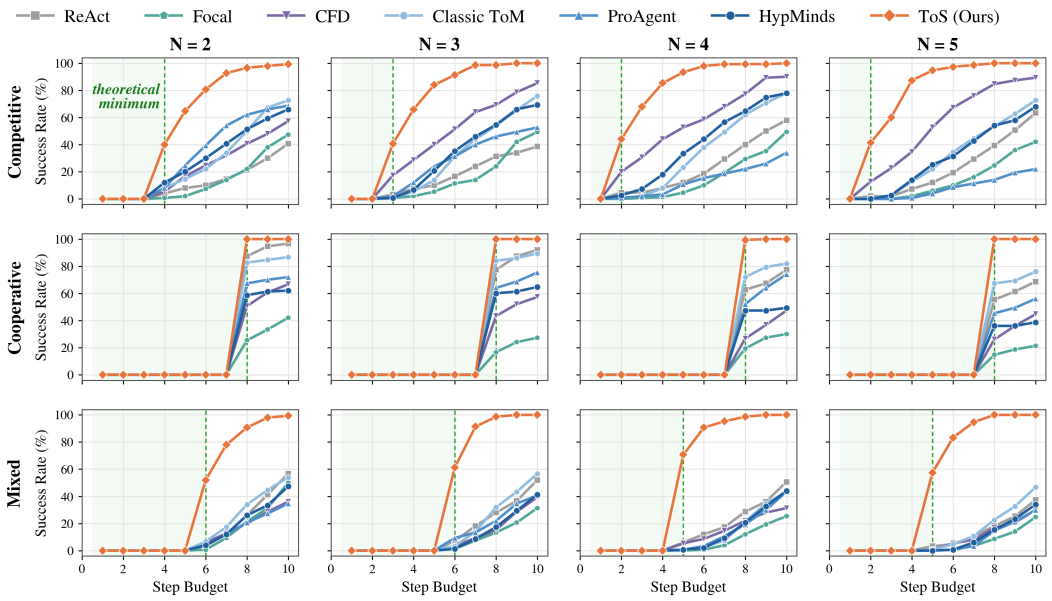

Figure 4: **Team size leaves ToS unchanged and lowers the baselines in DivvyBench Cooperative and Mixed.** DivvyBench success rate versus step budget from 2 to 5 agents, one panel per setting and team size.

Scaling the Population. Figure 5 scales the GovSim population from 5 to 100 while the pool stays at capacity 100. Every baseline decays as agents are added and has all but collapsed by 50 agents. ToS keeps its survival constant up to 20 agents in Fishery and Pasture and sustains the commons for more than half the horizon at 50. A fixed pool yields a fixed sustainable harvest, so the share each agent may take falls with every agent added, and a single agent that takes the whole pool collapses it for the group. At 100 agents the sustainable share is half a unit and a harvest is an integer, so the pool survives only if half of the agents harvest one unit and the other half harvest nothing.

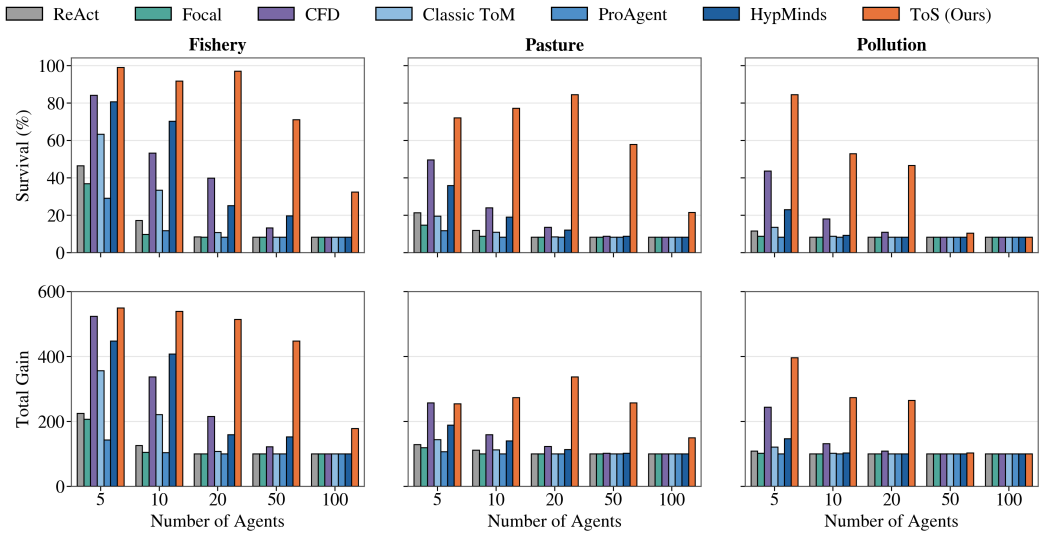

Figure 5: **ToS degrades more slowly than the baselines as the population grows.** GovSim survived rounds and total gain over populations from 5 to 100 on a pool of fixed capacity.

Per-Agent Demand Distribution. Figure 6 isolates the first round, when the pool is full and every method is still alive, and plots each agent's demand against the sustainable per-agent share. Every baseline leaves over-takers at every population size, and at 100 agents ReAct, Focal, Classic ToM, and ProAgent each leave more than 20% of the population over-taking, through one of two errors. The prediction-free ReAct and Focal size their demand so that what they leave behind regrows to full capacity, a calculation valid for one harvester, and arrive at half the pool, a quantity that does not fall as agents are added. The prediction-based Classic ToM and ProAgent instead anticipate heavy grazing by the others and harvest ahead of it. CFD and HypMinds make neither error and reduce their demand as the group grows, which is why they leave the fewest over-takers of the baselines. ToS divides the sustainable harvest by the number of agents it reads in the scene, and its over-takers are negligible at every population size. Appendix A.8 examines how ToM's prediction itself breaks down as the team grows.

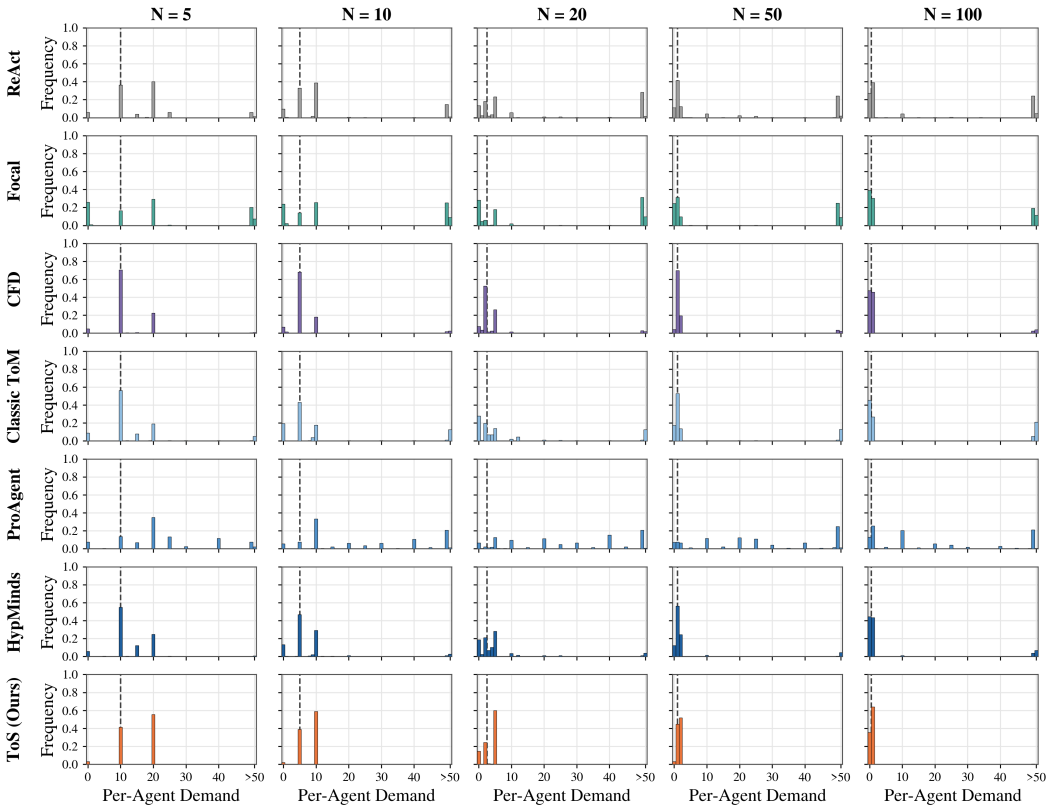

Figure 6: **The baselines leave over-takers at every population size and ToS leaves almost none.** Per-agent demand on GovSim (Pasture, first round), the fraction of the population at each demand level for every method and population, with the dashed line at the sustainable per-agent share. An over-taker asks for half the pool or more, which is the whole team's sustainable allowance and N times the dashed line.

A.7 Impact of the Interaction Horizon

Long-Horizon Survival. Figure 7 runs GovSim for 120 rounds, $10\times$ the default horizon. In Fishery, ToS sustains its harvest for the length of the game, and since the horizon is fixed and known, exhausting the resource in the final rounds, when preserving it has no remaining value, is the optimal end-game action. ToS reaches that point with the pool still at capacity, whereas CFD and HypMinds lose theirs earlier, so the fall in the survival curve marks the end game for ToS and a collapse for CFD and HypMinds. ToS also harvests the most over the rounds in which all three are alive, since a pool held at capacity regrows the most each round. The advantage measured over 12 rounds in the main results is therefore a lower bound on what a longer game reveals.

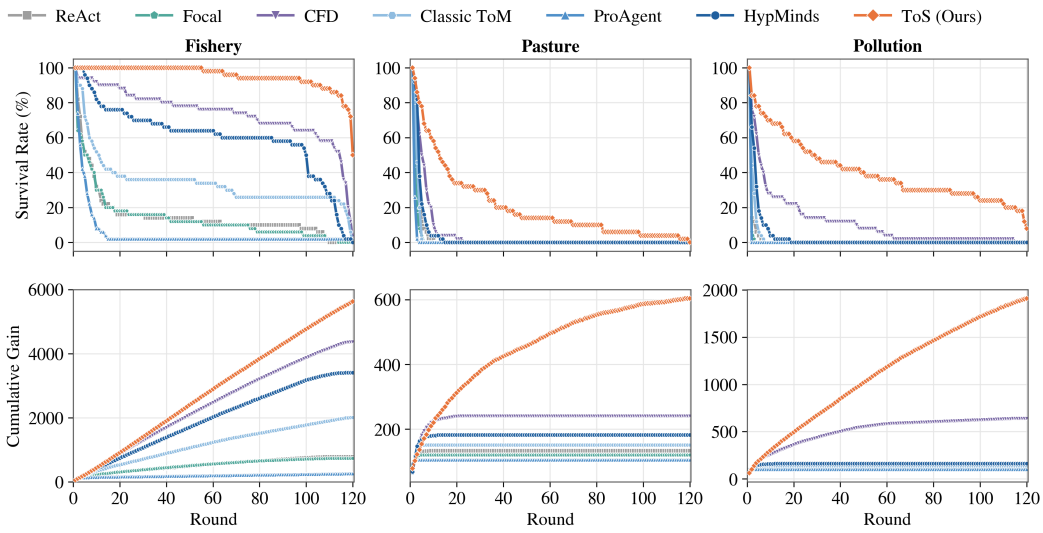

Figure 7: **ToS's advantage compounds over a long horizon.** GovSim run for 120 rounds, 10× the default horizon, reporting the survival rate (the fraction of runs whose commons is still alive at each round) and the cumulative total gain.

Premature Collapse. Appendix A.6 examines the first round alone, where ToS's over-taking is negligible, while a full run requires that same decision at every round. In the harder scenarios half of its runs have ended by round 12 in Pasture and by round 28 in Pollution. In its ordinary rounds the agents harvest almost exactly the sustainable half and leave the resource at its level, and the collapse is a single round in which they harvest nearly all of it at once. The `action` field on that round still divides the resource by the number of agents, as it does in every round, and divides the whole pool where every other round divides the sustainable half of it. Reasoning that is correct on average is therefore insufficient on an irreversible resource.

A.8 Failure of Theory of Mind

Prediction Accuracy. Table 9 scores each ToM baseline's prediction against the target the other agent then selected. The first step carries no history, and every later step carries the other agent's past actions, the input belief correction is built to read. Competitive accuracy does not improve once that history arrives and sits near 30% thereafter, so reading the other agent's past does not bring the prediction closer to its next choice. Cooperative accuracy stays above 70% at both points. The prediction is therefore accurate only where the agents must select the same target, which is where the prediction-free ReAct already succeeds (Table 1), and it is wrong where they must select different ones. ProAgent adds an intention inference and HypMinds a hypothesis test, and on the later Competitive steps both land within a few points of Classic ToM, so the failure lies in predicting a homogeneous agent, whatever the implementation of the prediction.

Table 9: **ToM prediction accuracy depends on the setting and does not improve with history.** The percentage of DivvyBench steps on which a baseline predicts the target the other agent then selected. *No History* is the first step, and *With History* every step after it.

	Competitive		Cooperative		Mixed	
Method	No History	With History	No History	With History	No History	With History
Classic ToM	34.3 ±2.8	30.5 ±1.1	91.7 ±1.6	91.5 ±0.6	45.8 ±3.0	56.2 ±1.1
ProAgent	29.5 ±2.6	30.3 ±1.1	90.2 ±1.7	78.4 ±0.9	46.3 ±3.1	49.0 ±1.0
HypMinds	43.1 ±2.9	29.6 ±1.1	79.7 ±2.3	73.9 ±0.9	48.6 ±3.0	44.4 ±1.0

Prediction Load. Figure 8 counts how many other agents each ToM baseline names in a single prediction step, across two tests, DivvyBench from 2 to 5 agents and GovSim from 5 to 100. The three baselines all predict the other agents and differ in how many of them they name. On a small team all three name every other agent, sitting on the cap at every DivvyBench size, so the prediction load scales one-for-one with the team. At a hundred agents, HypMinds keeps enumerating and names 92 of the 99, while Classic ToM and ProAgent stop enumerating and name 26 and 6, best-responding to a near-blind guess about the rest. Enumeration can lose the decision itself, since a prediction that long exceeds the generation budget or cannot be parsed, and at a hundred agents this removes 48% of HypMinds's decisions and 37% of Classic ToM's, leaving the agent to fall back on a default move. ProAgent holds that loss to 7% by naming almost no one. ToS conditions on the shared scene at no per-agent cost, so its reasoning does not lengthen as the team grows and still accounts for every agent through the scene.

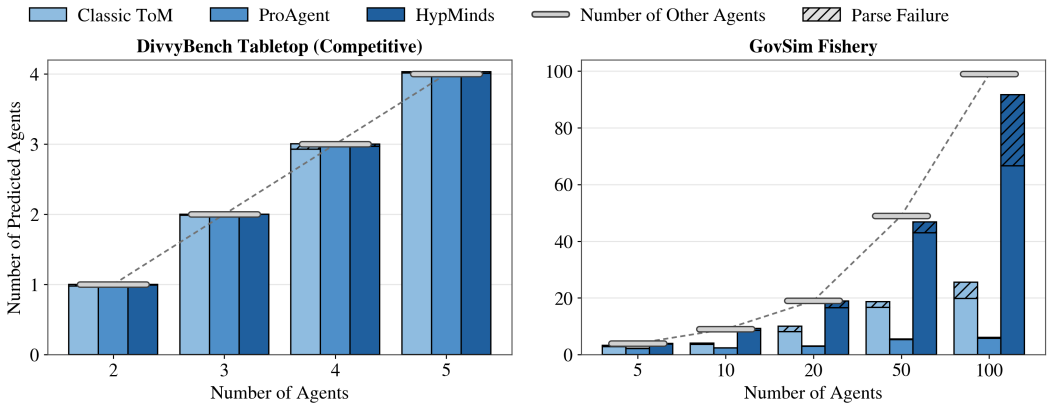

Figure 8: **Predicting a team does not scale.** Each bar is the mean number of other agents a ToM method names in a single prediction step, and the rounded cap marks the number of other agents ($N-1$). The hatched part is what the method predicted in responses whose reasoning failed to parse, so the agent fell back to a default move. Responses cut off at the generation budget are excluded from the mean.

A.9 Analysis of Per-Agent Distributions

Table 10 estimates δ_{ij}, the quantity the two only-if conditions of Proposition 4.2 constrain and the proposition leaves unmeasured. The estimate is taken at the first step, which every episode of a scenario enters on the same scene, so the actions recorded over those episodes give each agent's distribution over one set of targets. A team of 2 agents holds a single pair, so δ_{ij} is at once $\delta_{\min}$ and $\delta_{\max}$ and both conditions constrain one number. The complement $1-\delta_{ij}$ is the overlap of the two distributions and bounds from above the probability that the agents select the same target. A distance of 0 is necessary for converging and not sufficient, since agents sampling one distribution select the same target only as often as that distribution concentrates on it. In DivvyBench Cooperative every method is near 0, so the necessary condition is met by homogeneous agents that read nothing. No method approaches 1 in DivvyBench Competitive, where ToS reaches 0.780 and CFD, the highest baseline, reaches 0.480. Across the two settings each method runs one unchanged schema, so any change in δ_{ij} from one to the other comes from how the method reads the scene. ToS changes it by 0.780 against 0.440 for CFD, the largest change among the baselines, so the two readings of Equation (3) account for the change.

Table 10: **Only ToS moves the distance toward the value each step requires.** The total variation distance δ_{ij} between the target distributions of A0 and A1 on the first step of DivvyBench. The arrows give the value Proposition 4.2 requires, 1 where the step's targets have to be divided and 0 where one of them has to be selected together.

	DivvyBench - δ_{ij} at the first step	
Method	Competitive ↑	Cooperative ↓
ReAct	0.307 ±0.043	0.007 ±0.013
Focal	0.187 ±0.039	0.100 ±0.031
CFD	0.480 ±0.035	0.040 ±0.024
Classic ToM	0.327 ±0.041	0.047 ±0.020
ProAgent	0.280 ±0.043	0.067 ±0.027
HypMinds	0.467 ±0.043	0.047 ±0.026
ToS (Ours)	**0.780** ±0.034	**0.000** ±0.000

A.10 Confidence Intervals and Significance Tests

Table 11 reports how far ToS leads each baseline, as a two-sided 95% interval on every gap together with a test of that gap. The interval is a percentile bootstrap over 10,000 resamples and the test is a permutation test on the difference of means, with every resample and every permutation kept inside one DivvyBench scenario, one GovSim scenario, or one Overcooked recipe level, so the difficulty spread across those strata cannot enter the null distribution, and Overcooked uses the level-normalized statistic of the main table. ToS's margin is significant in every comparison but ReAct in DivvyBench Cooperative, where ReAct reaches 96.7% against ToS's 100.0% and the test returns $p = 0.054$ for that 3.3-point gap even though its interval excludes zero, since the permutation test is the more conservative of the two at this sample size.

Table 11: **ToS's lead is significant in 35 of the 36 comparisons.** Each baseline row gives ToS's lead over that baseline on the statistic Table 1 reports, and the brackets hold the two-sided 95% interval of the number before them. The ToS row gives its own score. ✓ marks a gap significant at $p < 0.05$ and ✗ one that is not.

	DivvyBench - Success (%) ↑		
Method	Competitive	Cooperative	Mixed
ReAct	+58.7 [51.3, 66.0] ✓	+3.3 [0.7, 6.7] ✗	+42.7 [34.7, 50.7] ✓
Focal	+52.0 [44.0, 59.3] ✓	+58.0 [52.7, 62.7] ✓	+49.3 [41.3, 57.3] ✓
CFD	+42.0 [36.0, 48.0] ✓	+33.3 [31.3, 35.3] ✓	+63.3 [56.7, 70.0] ✓
Classic ToM	+26.7 [19.3, 34.0] ✓	+13.3 [8.7, 18.0] ✓	+45.3 [37.3, 53.3] ✓
ProAgent	+30.7 [23.3, 38.0] ✓	+28.0 [22.0, 34.0] ✓	+64.7 [57.3, 72.0] ✓
HypMinds	+33.3 [26.0, 41.3] ✓	+38.0 [34.0, 42.0] ✓	+52.0 [44.0, 60.0] ✓
ToS (Ours)	99.3 [98.0, 100.0]	100.0 [100.0, 100.0]	99.3 [98.0, 100.0]
	GovSim		**Overcooked**
	Survival (%) ↑	Total Gain ↑	×ReAct ↑
ReAct	+58.7 [53.3, 63.9] ✓	+246 [223, 269] ✓	+0.67 [0.51, 0.94] ✓
Focal	+65.0 [59.6, 70.1] ✓	+257 [234, 280] ✓	+0.56 [0.41, 0.75] ✓
CFD	+26.0 [19.8, 31.9] ✓	+58 [29, 88] ✓	+0.51 [0.37, 0.68] ✓
Classic ToM	+53.0 [47.9, 58.0] ✓	+193 [169, 217] ✓	+0.26 [0.10, 0.45] ✓
ProAgent	+68.7 [63.7, 73.5] ✓	+284 [264, 303] ✓	+0.31 [0.16, 0.49] ✓
HypMinds	+38.7 [33.3, 44.1] ✓	+139 [115, 164] ✓	+0.18 [0.02, 0.36] ✓
ToS (Ours)	85.2 [80.9, 89.2]	400 [382, 419]	1.67 [1.48, 2.05]

B PROOFS OF PROPOSITIONS

Proof of Proposition 4.1. The team diverges when the N agents select N distinct targets. Each unordered set of N targets is assigned to the N agents in $N!$ orders, so the event has probability $N!\, e_N(p)$. By Maclaurin's inequality, $e_N(p) \le \binom{K}{N} K^{-N}$, with equality at the uniform distribution, so $\max_p N!\, e_N(p) = N!\binom{K}{N}K^{-N} = \prod_{n=1}^{N-1}\left(1 - \frac{n}{K}\right)$. The team converges when all N agents select the same target, an event of probability $\sum_k p_k^N$, which attains 1 when p assigns probability 1 to a single target. □

Proof of Proposition 4.2. Agents i and j select the same target with probability $\sum_k p_k^{(i)} p_k^{(j)}$, since both select target k with probability $p_k^{(i)} p_k^{(j)}$. If the team converges, every pair of agents selects the same target, so $P_{\text{conv}} \le \sum_k p_k^{(i)} p_k^{(j)}$ for every pair $i \ne j$, and applying $p_k^{(i)} p_k^{(j)} \le \min\left(p_k^{(i)}, p_k^{(j)}\right)$ term by term gives $P_{\text{conv}} \le \sum_k \min\left(p_k^{(i)}, p_k^{(j)}\right) = 1 - \delta_{ij}$. Taking the pair with the largest distance yields $P_{\text{conv}} \le 1 - \delta_{\max}$, and $P_{\text{conv}} = 1$ requires $\delta_{\max} = 0$. If the team diverges, no pair of agents selects the same target, so $P_{\text{div}} \le 1 - \sum_k p_k^{(i)} p_k^{(j)}$ for every pair, and by Cauchy-Schwarz together with $\sqrt{p_k^{(i)} p_k^{(j)}} \ge \min\left(p_k^{(i)}, p_k^{(j)}\right)$,

$$\sum_k p_k^{(i)} p_k^{(j)} \;\ge\; \frac{1}{K}\Big(\sum_k \sqrt{p_k^{(i)} p_k^{(j)}}\Big)^2 \;\ge\; \frac{1}{K}\Big(\sum_k \min\left(p_k^{(i)}, p_k^{(j)}\right)\Big)^2 \;=\; \frac{(1-\delta_{ij})^2}{K}.$$

Taking the pair with the smallest distance yields $P_{\text{div}} \le 1 - (1-\delta_{\min})^2/K$, and $P_{\text{div}} = 1$ requires $\delta_{\min} = 1$. At $\delta_{\min} = 0$ this bound reduces to $1 - 1/K$, which is Proposition 4.1 at $N = 2$, so one shared distribution is the case $\delta_{ij} = 0$ for every pair. □

C Experimental Details

C.1 Configuration and Benchmarks

Implementation Details. Table 12 lists the implementation configuration. The settings below are those of the main experiments, and the studies in Appendix A vary one of them at a time. We decode with the backbone's thinking mode disabled, so the generation budget goes to the reasoning schema itself. A response the model does not close as valid JSON is repaired when the error is a common one, an unquoted key, a single-quoted string, or a trailing comma, and when it still cannot be read the agent waits.

Table 12: Implementation details.

Parameter	**Value**
Backbone	Qwen3.5-35B-A3B
Temperature	0.7
Generation budget	8192 tokens
Max model length	16384 tokens
Thinking mode	Disabled
Serving	vLLM on 4 NVIDIA H100

Benchmarks. Table 13 lists the three benchmarks, DivvyBench, which we introduce, and the existing GovSim and Overcooked. GovSim gives a group of agents one regenerating pool and asks each of them privately how much to harvest this round, so the pool survives only while the group holds its total under the regrowth. Overcooked puts two cooks in a kitchen whose stations are shared and single-use, so a dish is delivered only when they divide the pipeline between them.

Table 13: Benchmark details.

	DivvyBench	**GovSim** (Piatti et al., 2024)	**Overcooked** (Sun et al., 2025)
Agents	2	5	2
Public role	Leader, Follower	A0 to A4	Chef, Assistant
Scenarios	Tabletop, Airspace, Household	Fishery, Pasture, Pollution	Levels 1–6
Settings	Competitive, Cooperative, Mixed	-	-
Action	One target	An integer harvest	One operation (e.g., `pickup`, `cut`)
Episode budget	10 steps	12 rounds	50 steps
Trials	150 per setting	50 per scenario	15 per level
Metrics	Success, Stuck	Survival, Total Gain	Throughput

C.2 DivvyBench

Rules. A team of N homogeneous agents faces 8 targets. Every agent sees all of them in the same order, may select any target still standing or wait, and all of them act in the same step without seeing what the others chose and without exchanging a message. A target that no agent takes stays where it is, and the step is wasted. A target is either Competitive or Cooperative, which the prompts label light and heavy, and the three settings differ only in the mix.

- **Competitive** contains only Competitive targets, each taken when exactly one agent selects it, while two or more agents selecting it collide and take nothing.
- **Cooperative** contains only Cooperative targets, each taken only when every agent selects it in the same step, while anything short of that takes nothing.
- **Mixed** makes the first 4 targets Cooperative and the rest Competitive, so both rules apply at once.

Scenarios. One set of rules runs in three scenarios, which differ in what the targets are. **Tabletop** puts 8 balls on a table, named by color, red, blue, green, yellow, orange, purple, pink, and brown. **Airspace** puts 8 areas in a survey grid, A1, A2, B1, B2, C1, C2, D1, and D2. **Household** puts 8 indoor locations taken from Zhang et al. (2024b), the bedroom, dishwasher, fridge, kitchen cabinet, living room, microwave, office, and stove. A scenario holds its targets fixed across episodes, so what

varies from episode to episode is the sampling, never the layout, and Figure 9 shows one Tabletop episode of each pure setting.

Metrics. Two numbers summarize an episode, one for whether the team finished and one for how much of its effort was wasted getting there. In DivvyBench, the theoretical minimum is known in closed form, and an episode scores all or nothing, so each method is measured against the best any team could do as well as against the other methods. Figure 2 brackets a step count with two references. A run at the theoretical minimum spends one step on each Cooperative target and shares the Competitive ones out among the agents, which is 4, 8, and 6 steps for the three settings with two agents, while a single agent takes one Competitive target per step and has nobody to collide with, so it completes the Competitive setting in 8 steps however it reasons, and a team that needs more steps performs worse than one agent alone.

- **Success** is the fraction of the 150 episodes in which the team takes every target within the 10-step budget, 50 episodes in each scenario. The budget leaves little slack over the theoretical minimum, so finishing is close to all-or-nothing.
- **Stuck** is the fraction of steps that take nothing, whether from a collision on a Competitive target, a partial reach on a Cooperative one, or a wait. It keeps the methods apart once success saturates and says how a team failed, since missing the budget with few stuck steps means it was slow while many means it was deadlocked.

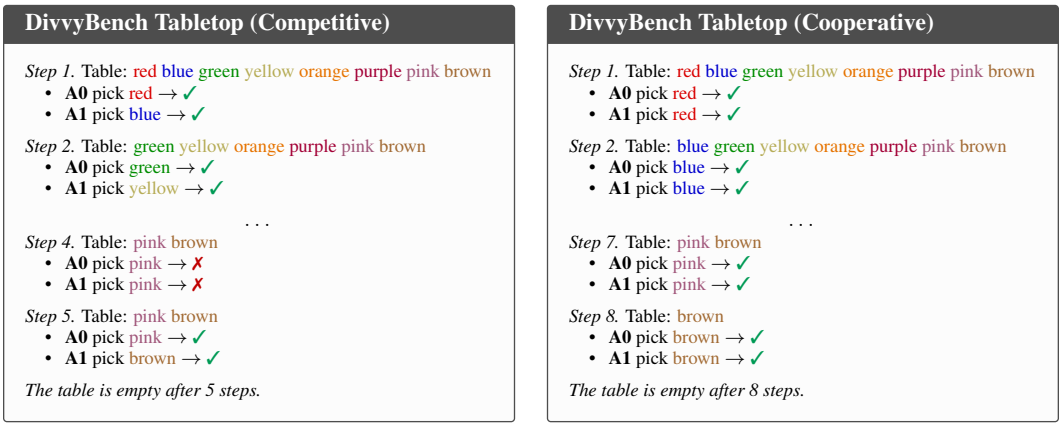

Figure 9: **One Tabletop episode in each setting.** The agents act at the same step, and the same joint action has opposite outcomes in the two settings. On the left, two agents on one ball collect neither (✗) and leave the table unchanged, and on the right they collect it together (✓), where a split would collect nothing.

C.3 GovSim

Rules. N agents share one regenerating pool for at most 12 rounds. The pool holds up to 100 units, and at the start of a round every agent privately names an integer between 0 and 100, and that many units are removed at once. What remains then doubles, capped at the capacity, and the pool collapses once it falls below 5 units. Each agent is told to maximize what it harvests over the long run, and its observation carries the current pool level and the harvests every agent made in the earlier rounds. The three scenarios put the same arithmetic in three framings, a lake of fish, a public pasture, and a river shared by factories. We remove two things GovSim gives the agents. *(i)* The communication period, in which every agent's harvest is announced and the agents negotiate and persuade one another before the next round. We drop the sentences that announce it and never run the exchange, so nothing an agent writes reaches another. *(ii)* The universalization hint, a line telling an agent what the resource would do if every agent took what it is about to take. We never inject it, since it would hand the agent, from outside, the reasoning a coordination method is meant to produce.

Metrics. Two numbers summarize a run, one for how long the resource lasts and one for how much comes out of it.

- **Survival** is the number of rounds the pool stays above the collapse threshold, reported as a percentage of the 12-round horizon.
- **Total Gain** is the units harvested over the whole run, summed across the agents and the rounds.

C.4 OVERCOOKED

Rules. Two agents cook in one kitchen for at most 50 steps. A recipe is a pipeline running from fetching an ingredient through cutting or blending it, cooking or baking it, plating it, and delivering the dish, and every station is single-use, so a station one agent is working is closed to the other for that step. Each of the six difficulty levels holds five recipes, and each recipe is run three times, giving 15 episodes per level. We change two things. *(i)* We run the open kitchen, in which every station is within reach of either agent and each of them holds the full recipe, so no agent is confined to a part of the pipeline by what it can reach. The layout is close to the Cramped Room of Carroll et al. (2019), where the agents share one room and contend for the same stations. *(ii)* Collab-Overcooked's action space carries a request action beside the cooking operations, which hands another agent an operation to perform and sends it a message. We remove that action, so nothing an agent can do addresses another and the shared kitchen state is all they hold in common.

Metrics. Every method delivers the dish on almost every episode, so success does not separate them and we report throughput, the number of dishes delivered within the 50 steps. The main table aggregates it over the six levels in two ways.

- **Raw** is the unweighted mean of the six level means. Yield differs widely across levels, 6.47 dishes at level 1 against 1.20 at level 6 for ToS, so the easier levels dominate it.
- $\times$**ReAct** divides each level mean by ReAct's mean at the same level before taking that mean, placing the levels on a common scale. The two levels above then read 1.31 and 1.12 for ToS.

D PROMPT TEMPLATES

D.1 DIVVYBENCH

The three DivvyBench settings use the same task prompt apart from the rules block. The figures show the task description, the rules, and the public role, and at each step the prompt continues with the current state, the legal actions, and the history, followed by the method's reasoning schema.

Task Prompt: DivvyBench Tabletop (Competitive)

```
There are {n_balls} balls on a table: {ball_list}. You and {n_teammates} teammate(s) form
a {n_agents}-robot team, and together you must collect all of them in as few steps as
possible.

**Rules**
- Each step, each of you reaches for one ball **at the same time** - there is no
turn-taking. You may also wait.
- You **cannot** communicate, and you cannot see which ball anyone else will reach for
this step.
- If **two or more** of you reach for the same ball, you collide: none of you gets it and
the step is wasted.
- A ball is single-use: if exactly one of you reaches for it, that robot collects it for
the team; if two or more of you reach for the same ball you collide and nobody gets it
(the step is wasted). Keep going until the table is empty.

## Who you are
You are the **{role}** of this two-robot team (agent A{agent_id}); your teammate is the
**{other_role}** (agent A{other_id}). You both reason in exactly the same way -- your role
label is the only thing that distinguishes you.
```

Figure 10: Task prompt for DivvyBench Competitive in Tabletop.

Task Prompt: DivvyBench Tabletop (Cooperative)

```
There are {n_balls} balls on a table: {ball_list}. You and {n_teammates} teammate(s) form
a {n_agents}-robot team, and together you must collect all of them in as few steps as
possible.

**Rules**
- Each ball is **heavy**: collecting it needs all of you to lift it together.
- Each step, each of you reaches for one ball **at the same time** - there is no
turn-taking. You may also wait.
- You **cannot** communicate, and you cannot see which ball anyone else will reach for
this step.
- If **all of you** reach for the same ball, you lift it together and collect it.
- If you do **not all** reach for the same ball (someone reaches elsewhere or waits),
nothing is collected and the step is wasted.

## Who you are
You are the **{role}** of this two-robot team (agent A{agent_id}); your teammate is the
**{other_role}** (agent A{other_id}). You both reason in exactly the same way -- your role
label is the only thing that distinguishes you.
```

Figure 11: Task prompt for DivvyBench Cooperative in Tabletop.

Task Prompt: DivvyBench Tabletop (Mixed)

```
There are {n_balls} balls on a table: {ball_list}. You and {n_teammates} teammate(s) form
a {n_agents}-robot team, and together you must collect all of them in as few steps as
possible.

**Rules**
- Each step, each of you reaches for one ball **at the same time** - no turn-taking, no
communication, and you cannot see anyone else's choice. You may also wait.
- Balls come in two kinds (each ball is labelled in the state below):
  - a **light** ball is single-use: if **exactly one** of you reaches for it, that robot
  collects it; if two or more of you reach for the same light ball you collide and nobody
  gets it (wasted).
  - a **heavy** ball needs all of you: it is collected only if **all of you** reach for
  it together; anything short of everyone (a split or a partial reach) collects nothing
  (wasted).
- Keep going until the table is empty.

## Who you are
You are the **{role}** of this two-robot team (agent A{agent_id}); your teammate is the
**{other_role}** (agent A{other_id}). You both reason in exactly the same way -- your role
label is the only thing that distinguishes you.
```

Figure 12: Task prompt for DivvyBench Mixed in Tabletop.

Task Prompt: DivvyBench Airspace (Competitive)

```
There are {n_areas} areas in a survey grid: {area_list}. You and {n_teammates} teammate(s)
form a {n_agents}-drone team, and together you must survey all of them in as few steps as
possible.

**Rules**
- Each step, each of you flies to one area **at the same time** - there is no
turn-taking. You may also wait.
- You **cannot** communicate, and you cannot see which area anyone else will fly to this
step.
- If **two or more** of you fly to the same area, you collide: none of you gets it and
the step is wasted.
- An area is single-pass: if exactly one of you flies to it, that drone surveys it for
the team; if two or more of you fly to the same area you collide and nobody gets it (the
step is wasted). Keep going until every area is surveyed.

## Who you are
You are the **{role}** of this two-drone team (agent A{agent_id}); your teammate is the
**{other_role}** (agent A{other_id}). You both reason in exactly the same way -- your role
label is the only thing that distinguishes you.
```

Figure 13: Task prompt for DivvyBench Competitive in Airspace.

Task Prompt: DivvyBench Household (Competitive)

```
There are {n_locations} locations in an apartment: {location_list}. You and {n_teammates}
teammate(s) form a {n_agents}-robot team, and together you must search all of them in as
few steps as possible.

**Rules**
- Each step, each of you goes to one location **at the same time** - there is no
turn-taking. You may also wait.
- You **cannot** communicate, and you cannot see which location anyone else will go to
this step.
- If **two or more** of you go to the same location, you collide: none of you gets it and
the step is wasted.
- A location is single-visit: if exactly one of you goes to it, that robot searches it
for the team; if two or more of you go to the same location you collide and nobody gets
it (the step is wasted). Keep going until every location is searched.

## Who you are
You are the **{role}** of this two-robot team (agent A{agent_id}); your teammate is the
**{other_role}** (agent A{other_id}). You both reason in exactly the same way -- your role
label is the only thing that distinguishes you.
```

Figure 14: Task prompt for DivvyBench Competitive in Household.

D.2 THEORY OF SCENE

Figure 15 gives the ToS reasoning schema, the block appended to the task prompt on every benchmark. Section 4.2 writes that schema with one `action` field. The implementation splits that step in two, a `plan` field that states the division the readings name and an `action` field constrained to copy one action verbatim from the currently legal ones, because a schema that emits the action directly returns actions outside that set. Every schema in the comparison carries the same split.

Prompt Template: Theory of Scene (ToS)

```
{benchmark_scaffold}

## Coordinating from the shared observation
Decide from the observation you and the other agents all see: reading the same scene and
reasoning alike, a rule anchored to that scene leads you all to the same division of
labor.

**Role.** Read the ownership: is it overlapping -- do you and the other agents all work
the same targets and resources -- or is it divided, each of you already holding its own
part?
  - **Overlapping** (acting alike you would contend for one target or all defer):
  coordinate -- settle a division of labor over the shared targets from the scene.
  - **Divided** (your role owns some stages/stations and the other agents own the rest):
  hold your part and execute. Take the single most useful action within your own part; do
  not step onto a station or task that belongs to another agent's part, even if it looks
  like the most useful step right now -- reading the same scene, they are already taking
  it, so you would only collide. If your own part has no ready step (it waits on their
  output), do its enabling step or start the next independent unit; wait only when
  nothing of yours is productive.

**Task.** Settle the division over the shared targets: read how each pending target
couples you and apply its rule over the canonical order:
  1. **Joint** (succeeds only if you all act on it together): converge -- all take the
  same one: the first such target in that order. (Acting alone wastes it.)
  2. **Exclusive** (single-occupancy -- multiple acting on it wastes it): divide by item
  -- take your share by role over that order; the other agents take the complement, so
  you never collide.
  3. **Sequential** (ordered hand-off chain): divide by stage -- work the stage your role
  owns, by the chain order.

## Reply format
Reply with a single JSON object and nothing else:

{
  "state":   "<from the observation only: what you hold + which steps are doable now
  (inputs present); world state, never the other agents' intent>",
  "role":    "<is ownership overlapping (you and the other agents all work the same
  targets/resources -> coordinate) or divided (your role owns some stages/stations, the
  others own the rest -> execute your own), and why>",
  "task":    "<for each pending shared target, how it couples you -- Joint / Exclusive /
  Sequential>",
  "plan":    "<overlapping: apply the matching rule per target over the canonical order
  -- Joint: converge on the first such target; Exclusive: your role's share; Sequential:
  your stage; divided: the most useful step within your own part (enabling step or next
  unit if none ready), not a step that belongs to another agent's part>",
  "action":  "<copy one action verbatim from your currently-legal actions>"
}
```

Figure 15: Prompt for ToS.

E THEORY OF SCENE REASONING EXAMPLES

Figures 16 to 18 reproduce ToS reasoning verbatim from the runs, one per benchmark, with the role and task readings highlighted. Role gating yields both of its readings, since on DivvyBench and GovSim the agents read ownership as overlapping and have to settle a division, while on Overcooked they read it as already divided and each advances a stage of the pipeline. The three couplings all appear, *Joint* and *Exclusive* in DivvyBench Mixed, *Exclusive* on GovSim, and *Sequential* on Overcooked.

ToS Reasoning on DivvyBench Tabletop (Mixed)

Context. Agents: 2 Balls: 8, 4 heavy and 4 light Rule: a heavy ball needs both agents at once, a light ball needs exactly one and collides if both take it

Step 1. On the table: red (*heavy*), blue (*heavy*), green (*heavy*), yellow (*heavy*), orange (*light*), purple (*light*), pink (*light*), brown (*light*).

A0 → `pick_red`

- `role`: **Overlapping**. Both agents have access to the same set of targets and resources; there are no distinct stages or stations assigned to specific roles. Therefore, we must [...]
- `task`: **Joint**: red, blue, green, yellow (require both agents to pick the same ball). **Exclusive**: orange, purple, pink, brown (require exactly one agent to pick; if both pick, [...]
- `plan`: Apply the canonical order rule. For Joint targets, converge on the first one in the list (red). For Exclusive targets, split them by role (Leader takes the first [...]

A1 → `pick_red`

- `role`: **Overlapping**. Both agents see the same state, have the same capabilities, and no role-specific stations or stages exist. Therefore, they must coordinate to avoid [...]
- `task`: **Joint**: The four heavy balls (red, blue, green, yellow) require both agents to pick the same one simultaneously to succeed. **Exclusive**: The four light balls (orange, [...]
- `plan`: Apply the coordination rule: 1. Joint targets: Both agents must converge on the first heavy ball in the canonical order (red). 2. Exclusive targets: Since we are [...]

Step 5. On the table: orange (*light*), purple (*light*), pink (*light*), brown (*light*).

A0 → `pick_orange`

- `role`: **Overlapping**. Both agents are identical in capability and have access to the same set of remaining targets (the four light balls). There are no distinct stages or [...]
- `task`: All remaining targets (orange, purple, pink, brown) are **Exclusive** (light balls). If both agents pick the same one, they collide and fail; if they pick different ones, [...]
- `plan`: The agents must divide the Exclusive targets to avoid collision. Using the canonical order (orange, purple, pink, brown) and a deterministic role-based split (e.g., [...]

A1 → `pick_purple`

- `role`: **Overlapping**
- `task`: **Exclusive**: The remaining targets are all light (single-use) balls. Since both robots are identical and overlapping, acting on the same ball causes a collision (waste). [...]
- `plan`: As the follower (A1) and the leader (A0) share the same reasoning and canonical order, we must split the Exclusive targets. A standard coordination rule for overlapping [...]

Figure 16: ToS reasoning on DivvyBench Tabletop (Mixed), where one table carries both couplings. On Step 1 the agents read the Cooperative balls as *Joint* and commit to the first one in the shared order, since a Cooperative ball is taken only when all agents select it together. On Step 5 the Cooperative balls are gone, the agents read the remaining Competitive balls as *Exclusive*, and they split them without exchanging a message.

ToS Reasoning on GovSim Fishery

Context. Agents: 5 Lake: 100 tons Regrowth: survivors double, capped at 100 Horizon: 12 months

Month 1 of 12. Lake at 100 tons.

A0 → `harvest 10`

- `role`: **Overlapping**; all 5 agents are fishermen with identical capabilities and access to the same shared resource (the lake), contending for the same fish stock.
- `task`: **Exclusive**; the fish in the lake are a shared resource where multiple agents acting on it (catching) creates competition. To avoid overfishing and ensure sustainability for long-term income, the agents must divide the catch among themselves.
- `plan`: Since we are overlapping and the resource is Exclusive (single-occupancy per unit of fish, but shared by the group), we must coordinate a division of labor to maximize long-term yield. The optimal sustainable strategy for a logistic growth model with carrying capacity K=100 and growth rate r=1 (doubling) is to harvest at the Maximum [...]

A1 → `harvest 10`
A2 → `harvest 10`
A3 → `harvest 10`
A4 → `harvest 10`

Figure 17: ToS reasoning on GovSim (Fishery). The agents read the pool as shared and each asks for its per-agent share of what the pool can regrow, so the total stays sustainable without anyone proposing it. Only A0 is expanded, and the other four reach the same two readings and the same catch.

ToS Reasoning on Overcooked

Context. Agents: 2 Dish: baked bell pepper soup (level 3) Recipe: bell pepper → cut → bake → cook → plate → deliver
Stations: A0 pot (cook) and oven (bake), A1 chopping board (cut) and blender (stir)

Step 1. Kitchen empty, both agents idle.

A0 → `pickup(dish,dish_dispenser)`

- `role`: **Divided**. Chef owns cooking stations (pot, oven) and serving, while Assistant owns prep stations (ingredient dispenser, chopping board, blender). Since the first step [...]
- `task`: **Sequential**. The process is a chain: Ingredient Pickup → Cut → Bake → Cook → Serve. The Chef's stage is the second half (Bake, Cook, Serve). The Assistant's stage is [...]
- `plan`: Divided roles: Assistant handles ingredient pickup and chopping. Chef handles baking, cooking, and serving. Since the Assistant must start the process by picking up the [...]

A1 → `pickup(bell_pepper,ingredient_dispenser)`

- `role`: **Divided**
- `task`: **Sequential**
- `plan`: My role is the Assistant, owning the prep stage (ingredient dispenser, chopping board). The Chef owns the cooking stage (oven, pot). The workflow is sequential: I must [...]

Step 2. A0 holds a dish, A1 holds a bell pepper, all stations empty.

A0 → `wait(1)`

- `role`: **Divided**. The Chef is responsible for cooking stations (oven, pot) and serving, while the Assistant handles prep (chopping board) and fetching. The Chef cannot cut [...]
- `task`: **Sequential**. The workflow requires: 1. Assistant cuts pepper → 2. Hand off slices → 3. Chef bakes → 4. Hand off baked slices → 5. Chef cooks → 6. Chef plates. The [...]
- `plan`: The Assistant currently holds the raw bell pepper and needs to cut it. The Chef holds a dish (prepared for serving) but cannot cook anything yet because the ingredients [...]

A1 → `put_obj_in_utensil(chopping_board0)`

- `role`: **Divided**. The Assistant owns the prep stage (ingredient dispenser, chopping_board, blender) and the Chef owns the cooking stage (oven, pot) and serving. The workflow is [...]
- `task`: **Sequential**. The target is the 'Baked Bell Pepper Soup' order. The chain is: 1. Assistant cuts bell_pepper (my stage). 2. Chef bakes slices (their stage). 3. Chef cooks [...]
- `plan`: I have the bell_pepper in hand and the chopping_board0 is empty. According to the recipe, I must cut the bell pepper. The legal action to start this process is to place [...]

Figure 18: ToS reasoning on Overcooked. The recipe is a chain and the public role gives each agent its own stations, so ownership reads as divided and no agent settles a division. On Step 1, A1 opens the chain by fetching the pepper while A0, whose stations have nothing to work on yet, fetches the dish the order will be served on. On Step 2, A1 loads the board and A0 waits, since nothing in its own part is ready.