INFLUENCE-SALIENT COORDINATION SHAPING FOR SCALABLE COOPERATIVE MARL

Wei Sheng
Department of Computer Science
Purdue University
shengw@purdue.edu

Rohan Paleja
Department of Computer Science
Purdue University
rpaleja@purdue.edu

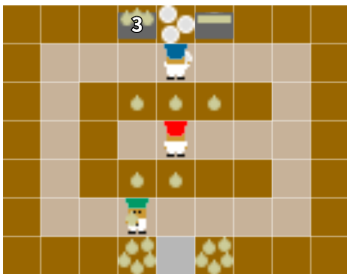
ABSTRACT

Interaction-driven coordination is central to real-world teamwork, yet cooperative multi-agent reinforcement learning often struggles to induce it. This difficulty compounds as agent populations scale, since decentralized learning and weak credit assignment under combinatorial interaction structures can yield brittle, loosely coupled routines with limited mutual responsiveness. To tackle these challenges, we propose Influence-Salient Coordination Shaping (ISCS), a scalable shaping mechanism for learning team coordination in cooperative multi-agent systems. ISCS identifies influence-salient choices by selecting actions that maximize expected transition displacement in a learned representation space, then computes a directed, baseline-adjusted uplift-based shaping bonus that rewards actions increasing the likelihood of subsequent teammate coordination beyond what the joint observation alone predicts. To reduce timing sensitivity, ISCS optimizes uplift over a short-horizon coordination event rather than requiring an immediate next-step response, improving robustness to delayed responses and reducing spurious attribution from state-induced correlations. Experiments on challenging cooperative benchmarks show that adding ISCS to standard CTDE methods improves sample efficiency and final performance over strong baselines. Code is available at: github.com/SCALE-Robotics-Lab/Influence-Salient-Coordination-Shaping/tree/AIMS-ICLR2026.

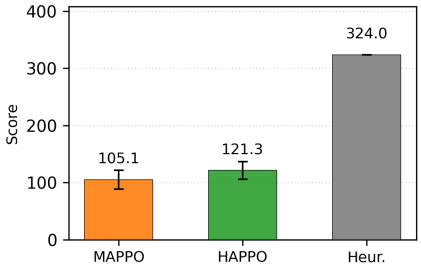
1 INTRODUCTION

Effective cooperation often relies on interaction-driven coordination (Lowe et al., 2017), in which each teammate's choices shape and adapt to the behaviors of others rather than simply contributing independent progress (e.g., transferring a resource, role switching, and timing complementary actions). Such coordination arises in human–human teams and in human–agent collaboration (Siu et al., 2021; Natarajan et al., 2023), and it is also required in fully autonomous multi-agent systems (Cao et al., 2012; Seraj et al., 2024), where agents must continually anticipate and respond to one another during execution. However, reliably learning and stabilizing mutually responsive teamwork remains difficult in cooperative multi-agent reinforcement learning (MARL) (Oroojlooy and Hajinezhad, 2023), even for strong baselines such as Multi-Agent Proximal Policy Optimization (MAPPO) (Yu et al., 2022) and Heterogeneous-Agent Proximal Policy Optimization (HAPPO) (Kuba et al., 2022). To further highlight this gap in teamwork, we deploy MAPPO and HAPPO in a simple 3-agent layout in Overcooked-AI (Carroll et al., 2019), where synced collaboration via passing objects up and down counters can result in high reward. Even with this simplistic collaborative scenario we term the "Pipeline Layout" (Fig. 1), we see that the performative MARL approaches of MAPPO and HAPPO fall very short of a simple heuristic (68% decrease and 63% decrease in reward, respectively). This gap raises a natural question: under standard MARL objectives, is the shared extrinsic reward alone sufficient to induce such obvious coordination behaviors and stabilize mutually responsive teamwork?

A key obstacle is that this form of coordination demands directed credit assignment (Foerster et al., 2018), i.e., identifying who caused what in multi-agent behavior. In practice, attribution is easily confounded because correlations explained by the (joint) observation can generate spurious apparent influence (Li et al., 2022), and coordination effects often materialize only after several steps, making one-step attribution timing-sensitive (Xiao et al., 2022). Non-stationary co-adaptation further drifts influence estimates (Foerster et al., 2017), moreover, as team size grows, interactions scale



(a) Pipeline layout (heuristic behavior)



(b) Standard MARL baselines underperform the heuristic

Figure 1: Case study showing that classical MARL baselines struggle to develop passing heuristics in a 3-agent setting.

combinatorially and the joint decision space expands accordingly (Yang et al., 2018; Qu et al., 2020), amplifying both the credit-assignment burden and the potential for spurious correlations. Together, these challenges motivate an intrinsic learning signal that separates directed interaction effects from observation-explained correlations while remaining robust to short-horizon delays.

We propose Influence-Salient Coordination Shaping (ISCS), a reward-shaping method for learning directed teammate responses. ISCS identifies influence-salient actions by selecting choices with the largest expected transition displacement in a learned representation space. It then assigns an uplift bonus, where uplift (Radcliffe, 2007) is the predicted increase in the probability of a teammate's short-horizon coordination response when conditioning on the source agent's action relative to an observation-only baseline. This quantity is used as a directed signal to address the credit assignment problem: actions receive more reward when they are predicted to create better opportunities for teammate follow-up than would be expected from the observation alone. To accommodate delayed teammate responses, we compute uplift over a K-step coordination event rather than an immediate reaction. This baseline adjustment reduces state-induced confounding and yields a more stable learning signal as team size grows. Intuitively, ISCS reinforces actions that create opportunities for teammates to follow up, encouraging coordination to emerge.

Across cooperative benchmarks, ISCS improves both sample efficiency and final performance, with gains that strengthen as team size increases. The main contributions of this work are summarized as follows:

- Influence-Salient Coordination Shaping (ISCS), a scalable shaping method that selects influence-salient actions via representation-space impact and assigns directed credit using a baseline-adjusted uplift bonus relative to an observation-only predictor.
- An event-level uplift objective that uses short-horizon coordination events instead of next-step labels, improving robustness to delayed teammate responses.
- Scalable multi-agent Overcooked-AI benchmark settings for more than two agents, including designed 3- and 4-agent layout/task variants implemented on the extensible many-world simulator Madrona (Shacklett et al., 2023), enabling higher-throughput training and systematic evaluation of large-team coordination.

2 Related Work

In this section, we review prior work on MARL methods, Overcooked-AI benchmarks, and influence-based shaping to position ISCS relative to existing approaches.

Coordination Structures in MARL. Decentralized control in Dec-POMDPs is inherently difficult (Bernstein et al., 2002; Oliehoek et al., 2008), motivating cooperative MARL to adopt Centralized Training with Decentralized Execution (CTDE) (Amato, 2024). In this paradigm, PPO-based actor-critic methods such as MAPPO/IPPO (Yu et al., 2022) and HAPPO/HATRPO (Kuba et al., 2022) have become particularly effective due to their ability to stablize learning under multi-agent non-

stationarity. Other CTDE variants emphasize robustness and adaptive coordination architectures (e.g., (Li et al., 2025b; ab Tessera et al., 2025)), while earlier centralized-critic methods such as MADDPG and COMA (Lowe et al., 2017; Foerster et al., 2018) use centralized critics and counterfactual baselines to address non-stationarity and credit assignment. Complementary value-based approaches learn coordination via structured factorization, including VDN(Sunehag et al., 2017), QMIX(Rashid et al., 2020), and QTRAN(Son et al., 2019). While these approaches learn interaction behavior insofar as it improves return, they do not explicitly target directed, short-horizon teammate responses, motivating our shaping-based approach.

Overcooked-AI benchmarks. Overcooked-AI (Carroll et al., 2019) simulates the Overcooked cooking game and is widely used for cooperative MARL evaluation because success depends on tightly coupled, interdependent coordination under spatial constraints. A common approach is to construct a population of simulated partners for training, with the goal of improving generalization to new teammates, rather than relying only on self-play with PPO (Schulman et al., 2017). Fictitious Co-Play (FCP) (Strouse et al., 2021) encourages robustness by optimizing collaboration across a partner population, and Other-Play (Hu et al., 2020) targets zero-shot coordination across conventions. Diversity can be further amplified via trajectory-level objectives in TrajeDi (Lupu et al., 2021), via population-based optimization in MEP (Zhao et al., 2023) built on the Population-Based Training (PBT) framework (Jaderberg et al., 2017), and via generative partner modeling in GAMMA (Liang et al., 2024) to better adapt to heterogeneous teammates. However, brittleness can persist even for learning-based teammates in this domain (Paleja et al., 2024), suggesting that partner diversity alone does not guarantee robust teamwork. Agents can become good at accommodating weak partners while failing to learn the interaction causes of high-quality teamwork. Together, these observations motivate approaches that more directly target synergistic coordination rather than merely coping with partner variability.

Behavior Diversity and Influence Shaping. Generalization in cooperative learning often improves when agents encounter varied partner behaviors and when training signals emphasize interaction-relevant consequences. Behavioral variety can be supplied externally, for example via heterogeneous demonstrations (Sreeramdass et al., 2025; Li et al., 2017) or semantically varied partners constructed with language-model-based agents (Park et al., 2023; Li et al., 2025a). Intrinsic motivation instead promotes exploration and behavioral diversity during learning, including information-maximization objectives (Mohamed and Jimenez Rezende, 2015) and exploration bonuses (Tang et al., 2017; Raileanu and Rocktäschel, 2020), with multi-agent extensions such as episodic curiosity (Zheng et al., 2021) and cooperation-promoting incentives like inequity aversion (Hughes et al., 2018). A particularly relevant direction uses influence-based intrinsic rewards that encourage an agent to affect teammates' behavior (Jaques et al., 2019; Wang et al., 2020; Du et al., 2024). These objectives encourage agents to seek controllable impact on others, providing a useful inductive bias for discovering interaction structure. Even so, raw influence is not necessarily aligned with task-relevant coordination outcomes, inspiring our later specialization toward directed coordination events.

3 PRELIMINARIES

Here, we establish the task setting and notation, and describe a representative influence-based intrinsic reward as a stepping stone toward our interaction-targeted shaping formulation.

3.1 PROBLEM FORMULATION

A cooperative multi-agent task can be modeled as a decentralized partially observable Markov decision process (Dec-POMDP) (Bernstein et al., 2002; Oliehoek et al., 2016) with a finite set of agents $N = \{1, \ldots, n\}$, global environment states $s \in \mathcal{S}$, per-agent actions $a^i \in \mathcal{A}^i$ with joint action $\mathbf{a} = (a^1, \ldots, a^n) \in \mathcal{A} = \prod_i \mathcal{A}^i$, and per-agent observations $o^i \in \Omega^i$ drawn from an observation model conditioned on the underlying state. At each time step t, each agent selects a_t^i from local information, the environment transitions according to $\mathcal{T}(s_{t+1} \mid s_t, \mathbf{a}_t)$, and the team receives a shared reward $r_t = R(s_t, \mathbf{a}_t)$. The objective is to learn decentralized policies $\pi_i(a^i \mid o^i)$ that maximize $\mathbb{E}\left[\sum_{t=0}^{H-1} \gamma^t r_t\right]$, where H is the finite episode horizon and $\gamma \in [0, 1]$ is the discount factor.

3.2 INFLUENCE-BASED INTRINSIC MOTIVATION

Influence-based intrinsic motivation rewards an agent for changing how other agents act. A representative formulation is Social Influence (Jaques et al., 2019). Consider an ordered pair of agents (i, j) with $i \neq j$. During training, agent j can be modeled with a conditional action distribution that depends on agent i's action: $\pi_j\Big(a_t^j \mid o_t^j, a_t^i\Big)$. Social Influence defines a counterfactual marginal action distribution for j in which the dependence on i is removed by averaging over counterfactual actions $\tilde{a}_t^i$,

$$\bar{\pi}_j\Big(a_t^j \mid o_t^j\Big) = \sum_{\tilde{a}_t^i} \pi_j\Big(a_t^j \mid o_t^j, \tilde{a}_t^i\Big)\; p\Big(\tilde{a}_t^i \mid o_t^j\Big) \tag{1}$$

which can be approximated by sampling counterfactual actions. The intrinsic reward for agent i is then defined as the sum of divergences between the conditional and marginal distributions of other agents,

$$c_{i,t} = \sum_{j \neq i} D_{\mathrm{KL}}\Big(\pi_j(\cdot \mid o_t^j, a_t^i) \;\Big\|\; \bar{\pi}_j(\cdot \mid o_t^j)\Big) \tag{2}$$

The shaped reward is then defined as a weighted combination of extrinsic and intrinsic terms (Jaques et al., 2019). This objective encourages agent i to take actions that make other agents' behavior more dependent on i, which provides a useful reference point for interaction-based shaping, which we later adapt to emphasize task-relevant coordination outcomes rather than generic dependence.

4 METHODS

Building on this interaction-based reference point, ISCS augments CTDE with a post-hoc intrinsic shaping term computed from on-policy rollouts. ISCS defines short-horizon coordination events over a K-step window and assigns credit via a directed uplift bonus: the increase in a teammate's event probability under an influence-conditioned predictor relative to an observation-only baseline. Unlike Social Influence (Jaques et al., 2019), which rewards actions that shift a teammate's action distribution relative to a counterfactual baseline, ISCS first selects high-impact targets via a task-aware representation (Eq. (5)), then scores how much an agent increases the likelihood of subsequent teammate follow-ups on these targets. Complete training and shaping details are given in Alg. 1; the next subsections describe each component.

4.1 INFLUENCE-SALIENT ACTIONS

To identify influence-salient actions, we measure how strongly an agent's choice is expected to change the team's near-term state in a learned representation. Let $\tilde{o}_t = (o_t^i)_{i \in N}$ denote the joint observation at time t. We first map $\tilde{o}_t$ to a learned embedding $\phi(\tilde{o}_t) \in \mathbb{R}^d$ using an encoder f_θ followed by a normalization operator $\mathrm{Norm}(\cdot)$:

$$\phi(\tilde{o}_t) = \mathrm{Norm}\big(f_\theta(\tilde{o}_t)\big) \tag{3}$$

While $\phi(\tilde{o}_t)$ captures high-dimensional features of the joint observation, we additionally expose structured task state through an explicit task-variable extractor $g(\cdot)$. Specifically, $g(\tilde{o}_t)$ concatenates binary indicators and normalized real-valued progress variables:

$$g(\tilde{o}_t) = \big[\, g_b(\tilde{o}_t)\;;\; g_r(\tilde{o}_t) \,\big] \tag{4}$$

Here $g_b(\tilde{o}_t) \in \{0,1\}^{m_b}$, $g_r(\tilde{o}_t) \in \mathbb{R}^{m_r}$, and $m = m_b + m_r$. Binary components encode discrete task status such as resource possession, role or mode indicators, availability or connectivity states, and key object-state predicates, while real-valued components encode continuous progress variables normalized to $[0, 1]$.

Finally, we combine the learned embedding and structured task variables into a task-aware representation $\Phi(\tilde{o}_t) \in \mathbb{R}^{d+m}$:

$$\Phi(\tilde{o}_t) = \big[\, \phi(\tilde{o}_t)\;;\; g(\tilde{o}_t) \,\big] \tag{5}$$

This representation thus couples flexible learned features $\phi(\cdot)$ with explicit task-critical variables $g(\cdot)$, and some overlap is expected since ϕ may also encode predictive task cues while g provides a stable, structured channel for them.

Algorithm 1 ISCS Training with Event-Level (K-Step) Directed Uplift

Input: rollout length T, salience window L, event horizon K, ISCS bonus coefficient β
Initialize actors $\{\pi_i\}_{i=1}^n$, critic V_ψ, directed event predictors $\{q_{i\to j}\}$, observation-only baselines $\{\omega_j\}$
for each policy update **do**
  // Rollout (extrinsic only)
  for $t = 0$ to $T-1$ **do**
    Sample $\mathbf{a}_t$ and log-probs from $\{\pi_i(\cdot \mid o_t^i)\}$
    Evaluate $V_\psi(\tilde{o}_t)$; step env
    Store $(\tilde{o}_t, \mathbf{a}_t, r_t^{\text{env}}, d_t)$
  end for
  // Event uplift and post-hoc shaping (window-fixed targets)
  for $t_0 = 0$ to $T_{\text{eff}} - 1$ step L **do**
    for each agent $i \in \{1, \dots, n\}$ **do**
      Compute target $a_i^* \leftarrow a_i^{\text{sal}(L)}(t_0)$ (Eq. 8)
    end for
    for $t = t_0$ to $\min(t_0 + L - 1,\ T_{\text{eff}} - 1)$ **do**
      Compute $m_t^{(K)}$ to exclude terminations (Eq. 12)
      for each agent $i \in \{1, \dots, n\}$ **do**
        Compute $\{y_{j,t}^{(K)}\}_{j\neq i}$ from $\mathbf{a}_{t+1:t+K}$ (Eq. 10)
        Apply $m_t^{(K)}$ and compute $r_{i,t}^{\Delta p,(K)}$ (Eq. 11)
        $r_t^i \leftarrow r_t^{\text{env}} + \beta\, r_{i,t}^{\Delta p,(K)}$
      end for
    end for
  end for
  Set $r_{i,t}^{\Delta p,(K)} \leftarrow 0$ for all i and $t \geq T_{\text{eff}}$
  // Train predictors on valid windows
  Train $\{q_{i\to j}\}$ and $\{\omega_j\}$ with CE on $y^{(K)}$ and $m^{(K)}$
  Reduce per-agent shaped rewards to team reward
  Bootstrap $V_\psi(\tilde{o}_T)$; compute GAE/returns
  Update actors and critic (PPO or HAPPO per config)
end for

With the representation $\Phi(\tilde{o}_t)$ in place, we score each candidate action by its expected displacement in the embedding space, and treat the highest-impact choices as influence-salient. For agent i, the salience (impact) score of an action $a^i \in \mathcal{A}^i$ at time t is modeled as

$$\mathcal{I}_i(\tilde{o}_t, a^i) = \mathbb{E}\big[\|\Phi(\tilde{o}_{t+1}) - \Phi(\tilde{o}_t)\|_2 \;\big|\; \tilde{o}_t,\ a_t^i = a^i\big] \tag{6}$$

where the expectation marginalizes over teammates' actions sampled from their current decentralized policies and any environment stochasticity. An influence-salient action at time t is any maximizer

$$a_i^{\text{sal}}(t) \in \arg\max_{a^i \in \mathcal{A}^i} \mathcal{I}_i(\tilde{o}_t, a^i) \tag{7}$$

In practice, we do not recompute a distinct influence-salient action $a_i^{\text{sal}}(t)$ at every timestep because per-step impact estimates can be noisy. We instead select a dominant influence-salient action over a short window of length L, since salient choices can shape teammate behavior over multiple subsequent steps, by maximizing the average score

$$a_i^{\text{sal}(L)}(t) \in \arg\max_{a^i \in \mathcal{A}^i} \frac{1}{L} \sum_{\ell=0}^{L-1} \mathcal{I}_i(\tilde{o}_{t+\ell}, a^i) \tag{8}$$

Such impact-based definition highlights actions that induce large changes in the task-aware representation $\Phi(\cdot)$, which often correspond to choices that create opportunities for teammates to respond and coordinate.

4.2 Directed Baseline-Adjusted Uplift

With the salience constructed as Eq. (8), we now move on to a directed, baseline-adjusted uplift signal that assigns credit to agent i for increasing the likelihood that a teammate executes $a_i^{\text{sal}(L)}(t)$ beyond

what is explained by the joint observation alone. For brevity, we denote the (window-fixed) target action by a^*, taking $a^* = a_i^{\mathrm{sal}(L)}(t)$ and holding this label fixed over the length-L window starting at t.

Consider an ordered agent pair (i, j) with $i \neq j$. Let $\tilde{o}_t$ denote the joint observation at time t and let $a_t^i \in \mathcal{A}^i$ be agent i's action. We introduce two predictor families: an influence-conditioned predictor $q_{i\to j}\Big(a_{t+1}^j \mid \tilde{o}_t, a_t^i\Big)$ and an observation-only baseline $\omega_j\Big(a_{t+1}^j \mid \tilde{o}_t\Big)$. Then with these components, the one-step uplift intrinsic reward term for agent i at time t is defined as

$$r_{i,t}^{\Delta p} := \frac{1}{n-1}\sum_{j\neq i}\Big[q_{i\to j}\big(a^* \mid \tilde{o}_t, a_t^i\big) - \omega_j(a^* \mid \tilde{o}_t)\Big] \tag{9}$$

This reward assigns positive credit when agent i's action increases the predicted probability that teammates execute the target action, beyond what is explained by the joint observation alone.

Both $q_{i\to j}$ and ω_j are trained by supervised learning from rollouts collected under the current CTDE policies. During each PPO iteration we execute the decentralized actors $\pi_{\theta_i}(a_t^i \mid o_t^i)$ in the environment to generate on-policy trajectories, and record the resulting tuples $(\tilde{o}_t, \mathbf{a}_t, \tilde{o}_{t+1})$. The influence predictors then minimize cross-entropy for next-action prediction on these trajectories, aggregated over ordered pairs $i \neq j$, while the baseline predictors minimize cross-entropy for a_{t+1}^j conditioned only on $\tilde{o}_t$.

The incremental intrinsic in Eq. (9) is inspired by influence-based intrinsic motivation (Section 3.2) but replaces counterfactual marginalization over full action distributions with a directed, baseline-adjusted probability lift on a salient coordination target a^*. Following standard intrinsic shaping (Wang et al., 2020), we optimize policies with the shaped reward $\hat{r}_{i,t} = r_t^{\mathrm{env}} + \beta\, r_{i,t}^{\Delta p}$, where r_t^{env} is the shared team reward and $\beta \geq 0$ controls the uplift strength.

4.3 Event-Level K-Step Uplift

The one-step uplift in Eq. 9 can be brittle when coordination responses occur after a short delay, e.g., when a teammate must first reposition before executing the target action. To reduce this timing sensitivity, uplift is generalized from next-action prediction to prediction of a short-horizon coordination event.

Remaining consistent with Section 4.2, we set the salient target action to $a^* = a_i^{\mathrm{sal}(L)}(t)$. For each target agent j, define the K-step event label

$$y_{j,t}^{(K)} := \mathbf{1}\Big\{\exists\, \tau \in \{1, \ldots, K\} \text{ s.t. } a_{t+\tau}^j = a^*\Big\}. \tag{10}$$

Thus $y_{j,t}^{(K)}$ indicates whether agent j executes the target action at least once within the next K steps. Keeping the same predictor families as Section 4.2, the influence-conditioned model predicts event likelihood $q_{i\to j}\big(y \mid \tilde{o}_t, a_t^i\big)$ and the observation-only baseline predicts $\omega_j(y \mid \tilde{o}_t)$, where $y \in \{0, 1\}$. The event-level uplift intrinsic is

$$r_{i,t}^{\Delta p,(K)} := \frac{1}{n-1}\sum_{j\neq i}\Big[q_{i\to j}\big(y{=}1 \mid \tilde{o}_t, a_t^i\big) - \omega_j(y{=}1 \mid \tilde{o}_t)\Big] \tag{11}$$

Event labels must not span episode terminations. Let $d_{t,e} \in \{0, 1\}$ denote the done flag for environment instance e at time t. We therefore mask out windows that contain any termination by defining

$$m_{t,e}^{(K)} \;:=\; \mathbf{1}\left\{\sum_{\ell=0}^{K-1} d_{t+\ell,e} = 0\right\}. \tag{12}$$

Only windows with $m_{t,e}^{(K)} = 1$ contribute to event supervision and to $r_{i,t}^{\Delta p,(K)}$. For a rollout segment of length T, event-level uplift is defined for $t \in \{0, \ldots, T - K - 1\}$.

Training follows Section 4.2, replacing next-action targets a_{t+1}^j with masked event targets $y_{j,t}^{(K)}$ in the cross-entropy objectives. The shaped reward $\hat{r}_{i,t}$ is defined analogously, substituting $r_{i,t}^{\Delta p,(K)}$ for $r_{i,t}^{\Delta p}$.

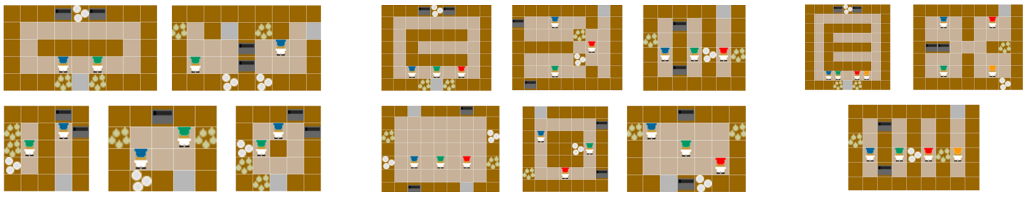


Figure 2: Layout overview for the 2-agent, 3-agent, and 4-agent settings. In the 2-agent setting, the top row (left to right) shows Pipeline and Asymmetric Advantages, and the bottom row (left to right) shows Forced Coordination, Cramped Room, and Coordination Ring. In the 3-agent setting, the top row (left to right) shows Pipeline, Asymmetric Advantages, and Forced Coordination, and the bottom row (left to right) shows Open Room, Coordination Ring, and Cramped Room. In the 4-agent setting, the top row (left to right) shows Pipeline, Asymmetric Advantages, and Forced Coordination.

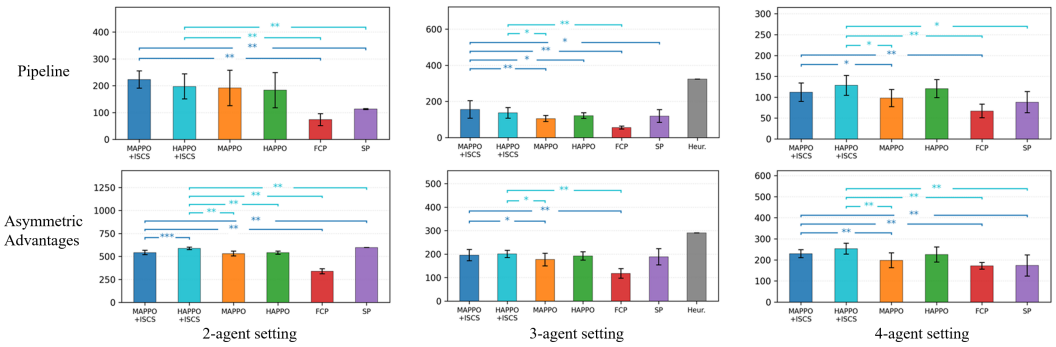


Figure 3: **Main results across team sizes on representative layouts.** Columns: 2-, 3-, and 4-agent settings. Rows: Pipeline (top) and Asymmetric Advantages (bottom). Bars show mean final episode return across 12 seeds; error bars indicate ± 1 std. Taken together, ISCS improves both CTDE backbones on these representative layouts, averaging +26.3% (MAPPO) / +8.9% (HAPPO) on Pipeline and +9.7% (MAPPO) / +8.5% (HAPPO) on Asymmetric Advantages across 2/3/4 agents, with the largest boost in 3-agent Pipeline (+48% over MAPPO). Brackets indicate paired Wilcoxon signed-rank tests with Holm correction (details in Appendix A).

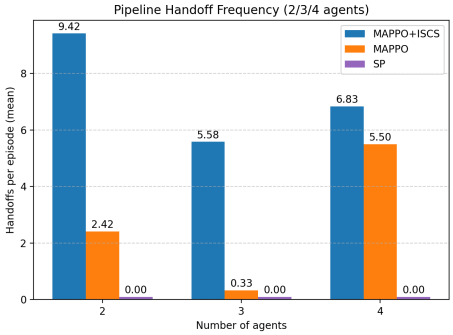
5 Experiments

We evaluate coordination in Overcooked with 2, 3, and 4 agents across layouts spanning passing-beneficial and congestion-dominated regimes (Fig. 2), and further include a hand-coded heuristic policy as a reference point for coordination performance; heuristic policy details are provided in Appendix C. All tasks require coordinated preparation and delivery of onion soup under standard Overcooked dynamics, using lightweight intermediate shaping to ease exploration; full layout enumeration and reward details are provided in Appendix A.1.

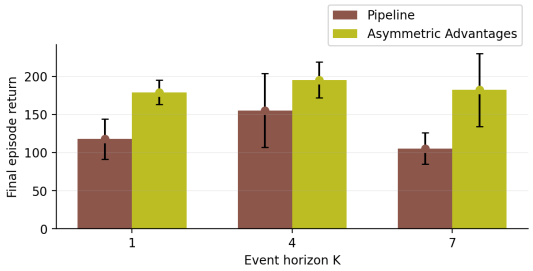
Comparisons span three common training paradigms in cooperative MARL: self-play PPO (SP) (Schulman et al., 2017), partner-diversity training via representative FCP (Strouse et al., 2021), and CTDE baselines MAPPO (Yu et al., 2022) and HAPPO (Kuba et al., 2022). ISCS is evaluated as an additive shaping bonus on top of MAPPO and HAPPO without changing decentralized execution. Results are reported as mean and standard deviation over multiple random seeds, with full training and evaluation details in Appendix A.2.

5.1 Results Across Team Sizes and Layouts

Figure 3 summarizes performance across team sizes on two representative layouts, Pipeline and Asymmetric Advantages. These layouts admit qualitatively different solutions, ranging from largely solo execution to clearly interaction-driven teamwork, and the benefits of coordination are directly observable. Performance is evaluated by the episode score (total game return), where higher scores indicate better cooperative performance. Complete results across all layouts are deferred to Appendix B.



(a) Counter-mediated handoff frequency



(b) Sensitivity to event horizon K (MAPPO+ISCS)

Figure 4: (a) Counter-mediated handoff frequency in Pipeline for 2, 3, and 4 agents (mean per episode), with handoffs counted using a $\delta{=}5$ step window (we use $\delta{=}K{+}1$ with $K{=}4$ under our step convention). (b) Sensitivity of MAPPO+ISCS to the event horizon K on Pipeline and Asymmetric Advantages (mean return over 12 seeds; error bars ± 1 std).

Across panels, ISCS consistently improves its CTDE backbone. On Pipeline, ISCS yields large backbone-relative gains: MAPPO+ISCS exceeds MAPPO by +16.5% (2 agents), **+48.0%** (3 agents), and +14.5% (4 agents) in mean final return (e.g., 155.5 vs. 105.1 at 3 agents), and HAPPO+ISCS exceeds HAPPO by +7.6%, +12.6%, and +6.4% respectively. On Asymmetric Advantages, improvements are more modest but still consistent, with MAPPO+ISCS improving over MAPPO by +2.3% (2 agents), +10.7% (3 agents), and +16.2% (4 agents), and HAPPO+ISCS improving over HAPPO by +8.4%, +4.7%, and +12.4%. Note that FCP underperforms SP under our self-play test protocol; this is not inconsistent with (Strouse et al., 2021), which evaluates robustness under partner shift with a fixed unseen partner population rather than self-play return (protocol details in Appendix B.4).

As team size increases, SP degrades, while ISCS continues to provide a stable gain relative to the same backbone, especially in Pipeline, where counter staging and role specialization are central. HAPPO is often more competitive than MAPPO at larger team sizes, consistent with the added stability of sequential policy updates (Kuba et al., 2022). Together, these trends support the view that ISCS alleviates a temporal attribution bottleneck that becomes more severe as cooperative interactions scale.

5.2 Case Study on Coordination Behavior

We continue using Pipeline as a diagnostic layout since its counter-staging and passing workflow makes coordination effects directly observable and typically reflected in return. However, Fig. 3 shows substantial headroom remains: in the 3-agent setting, the best-seed returns of MAPPO, HAPPO, SP, and FCP are approximately 50.7%, 44.9%, 25.0%, and 75.4% lower than MAPPO+ISCS, respectively. To contextualize the ceiling, a hand-designed Pipeline heuristic that approximates streamline passing reaches 324 return in the 3-agent setting (Appendix C). Even MAPPO+ISCS peaks around 272 in the same setting, motivating a focused analysis of policy behavior in Pipeline beyond aggregate return.

To quantify explicit passing, we measure *handoffs per episode* as counter staging followed by teammate retrieval within a short temporal window. We set $\delta = K + 1$ to align the handoff window with the ISCS event horizon: after staging at time t, retrieval by any teammate at $t + \tau$ with $\tau \in \{1, \ldots, K\}$ is counted as a handoff.

Fig. 4a shows clear behavioral separation, MAPPO+ISCS produces substantially more counter-mediated passing than MAPPO: 9.42 vs. 2.42 (2 agents), 5.58 vs. 0.33 (3 agents) handoffs/episode, and remains higher at 4 agents (6.83 vs. 5.50), while SP remains near zero. In Pipeline, shared-policy SP tends to produce symmetric routines (e.g., cyclic movement and sequential pot-filling) that accrue intermediate shaping but rarely induce prompt teammate exploitation of staged items. By contrast, ISCS explicitly rewards actions that increase short-horizon teammate follow-ups on salient interaction targets, which preferentially reinforces counter staging that becomes immediately useful to a particular teammate.

Table 1: Pipeline within-seed role diversity within a trained team. See Appendix A.4 for details.

AGENTS	MAPPO+ISCS	MAPPO	SELF-PLAY
2	**1.24**	1.21	1.20
3	**1.82**	1.80	1.47
4	**2.37**	2.23	1.85

In addition to handoff frequency, we summarize within-team role differentiation using a determinant-based behavioral diversity score computed from per-agent event-rate and visitation features (Appendix A.4). Table 1 shows that MAPPO+ISCS attains the highest role diversity across 2, 3, and 4 agents. Relative to MAPPO, it is higher by about +2.5% (2 agents), +1.1% (3 agents), and +6.3% (4 agents). Relative to self-play, the gap is larger at scale, reaching about +23.8% (3 agents) and +28.1% (4 agents), where role allocation is most underdetermined.

Collectively, higher handoff frequency and higher within-team role diversity reflect that ISCS promotes both more reliable passing and a more differentiated division of labor in Pipeline.

5.3 SENSITIVITY TO EVENT HORIZON K

This subsection ablate the event horizon K in the event-level uplift label while holding the MAPPO+ISCS pipeline fixed. We compare $K \in \{1, 4, 7\}$, ranging from immediate next-step labeling to a longer timing tolerance.

Fig. 4b shows that a medium horizon improves performance over an immediate label in both layouts.

In Pipeline, moving from $K{=}1$ to $K{=}4$ yields a +31.7% gain in mean return, while extending to $K{=}7$ produces a -32.1% drop relative to $K{=}4$. In Asymmetric Advantages, $K{=}4$ improves mean return by +9.1% over $K{=}1$, and $K{=}7$ reduces performance by -6.7% relative to $K{=}4$.

These trends match the timing structure of Overcooked coordination. With $K{=}1$, many legitimate follow-ups cannot occur on the next step due to travel time, approach-tile contention, or collision avoidance, making the label brittle. However, larger horizons can dilute attribution: as K increases, positives become less selective and more likely under typical dynamics even when the initiating action did not meaningfully influence the follow-up, which weakens directed shaping signal and matches the drop at $K{=}7$ in Pipeline. We therefore use a short window $K{=}4$ as the default in main experiments, balancing tolerance to short delays with keeping the event label tied to the initiating action.

Overall, these results suggest that ISCS primarily helps in interaction-forward settings by improving short-horizon action-to-response linkage as team size and coordination ambiguity increase.

6 DISCUSSION & FUTURE WORKS

We introduced Influence-Salient Coordination Shaping (ISCS), a directed shaping mechanism for interaction-driven coordination. Rather than relying solely on sparse team returns, ISCS converts delayed teammate-specific follow-up responses into a usable training signal through influence-salient action selection and a baseline-adjusted uplift bonus over a short-horizon K-step coordination event. This perspective is especially useful in settings where coordinated progress depends on one agent creating opportunities that others must subsequently realize.

Several directions remain important for future work. First, the current formulation relies on pairwise directed predictors, which introduces an $O(n^2)$ scaling cost as the number of agents grows; more scalable parameterizations, sharing schemes, or sparsity-aware approximations may be needed for larger teams. Second, the shaping signal depends on learned uplift predictors trained on non-stationary on-policy data, so understanding how prediction error, miscalibration, and drift affect the stability and reliability of the reward signal is an important next step. Finally, broader empirical validation would strengthen the picture further, including evaluation in additional multi-agent domains with denser action spaces and larger teams, as well as comparison against adjacent social-influence and KL-based shaping baselines. Together, these directions would clarify both the practical scope and the robustness of directed coordination shaping beyond the present study.

IMPACT STATEMENT

This work advances cooperative multi-agent reinforcement learning by introducing a shaping method for interaction-driven coordination. While such methods may benefit applications that require reliable teamwork among agents, real-world deployment should include robustness testing and safety oversight to mitigate risks from reward misspecification and unintended emergent behaviors.

REFERENCES

Kale ab Tessera, Arrasy Rahman, Amos Storkey, and Stefano V. Albrecht. HyperMARL: Adaptive hypernetworks for multi-agent RL. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Christopher Amato. An introduction to centralized training for decentralized execution in cooperative multi-agent reinforcement learning. *arXiv preprint arXiv:2409.03052*, 2024.

Daniel S Bernstein, Robert Givan, Neil Immerman, and Shlomo Zilberstein. The complexity of decentralized control of markov decision processes. *Mathematics of operations research*, 27: 819–840, 2002.

Yongcan Cao, Wenwu Yu, Wei Ren, and Guanrong Chen. An overview of recent progress in the study of distributed multi-agent coordination. *IEEE Transactions on Industrial informatics*, 9(1): 427–438, 2012.

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.

Xiao Du, Yutong Ye, Pengyu Zhang, Yaning Yang, Mingsong Chen, and Ting Wang. Situation-dependent causal influence-based cooperative multi-agent reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 17362–17370, 2024.

Jakob Foerster, Nantas Nardelli, Gregory Farquhar, Triantafyllos Afouras, Philip HS Torr, Pushmeet Kohli, and Shimon Whiteson. Stabilising experience replay for deep multi-agent reinforcement learning. In *International Conference on Machine Learning*, pages 1146–1155. PMLR, 2017.

Jakob Foerster, Gregory Farquhar, Triantafyllos Afouras, Nantas Nardelli, and Shimon Whiteson. Counterfactual multi-agent policy gradients. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Hengyuan Hu, Adam Lerer, Alex Peysakhovich, and Jakob Foerster. "other-play" for zero-shot coordination. In *International Conference on Machine Learning*, pages 4399–4410. PMLR, 2020.

Edward Hughes, Joel Z Leibo, Matthew Phillips, Karl Tuyls, Edgar Dueñez-Guzman, Antonio García Castañeda, Iain Dunning, Tina Zhu, Kevin McKee, Raphael Koster, et al. Inequity aversion improves cooperation in intertemporal social dilemmas. *Advances in Neural Information Processing Systems*, 31, 2018.

Max Jaderberg, Valentin Dalibard, Simon Osindero, Wojciech M Czarnecki, Jeff Donahue, Ali Razavi, Oriol Vinyals, Tim Green, Iain Dunning, Karen Simonyan, et al. Population based training of neural networks. *arXiv preprint arXiv:1711.09846*, 2017.

Natasha Jaques, Angeliki Lazaridou, Edward Hughes, Caglar Gulcehre, Pedro Ortega, DJ Strouse, Joel Z Leibo, and Nando De Freitas. Social influence as intrinsic motivation for multi-agent deep reinforcement learning. In *International conference on machine learning*, pages 3040–3049. PMLR, 2019.

Jakub Grudzien Kuba, Ruiqing Chen, Muning Wen, Ying Wen, Fanglei Sun, Jun Wang, and Yaodong Yang. Trust region policy optimisation in multi-agent reinforcement learning. In *International Conference on Learning Representations*, 2022.

Jiahui Li, Kun Kuang, Baoxiang Wang, Furui Liu, Long Chen, Changjie Fan, Fei Wu, and Jun Xiao. Deconfounded value decomposition for multi-agent reinforcement learning. In *International Conference on Machine Learning*, pages 12843–12856. PMLR, 2022.

Lihe Li, Lei Yuan, Pengsen Liu, Tao Jiang, and Yang Yu. Llm-assisted semantically diverse teammate generation for efficient multi-agent coordination. In *Forty-second International Conference on Machine Learning*, 2025a.

Simin Li, Zihao Mao, Hanxiao Li, Zonglei Jing, Zhuohang bian, Jun Guo, Li Wang, Zhuoran Han, Ruixiao Xu, Xin Yu, Chengdong Ma, Yuqing Ma, Bo An, Yaodong Yang, Weifeng Lv, and Xianglong Liu. Empirical study on robustness and resilience in cooperative multi-agent reinforcement learning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025b.

Yunzhu Li, Jiaming Song, and Stefano Ermon. Infogail: Interpretable imitation learning from visual demonstrations. *Advances in Neural Information Processing Systems*, 30, 2017.

Yancheng Liang, Daphne Chen, Abhishek Gupta, Simon S Du, and Natasha Jaques. Learning to cooperate with humans using generative agents. *Advances in Neural Information Processing Systems*, 37:60061–60087, 2024.

Ryan Lowe, Yi I Wu, Aviv Tamar, Jean Harb, OpenAI Pieter Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. *Advances in neural information processing systems*, 30, 2017.

Andrei Lupu, Brandon Cui, Hengyuan Hu, and Jakob Foerster. Trajectory diversity for zero-shot coordination. In *International Conference on Machine Learning*, pages 7204–7213. PMLR, 2021.

Shakir Mohamed and Danilo Jimenez Rezende. Variational information maximisation for intrinsically motivated reinforcement learning. *Advances in Neural Information Processing Systems*, 28, 2015.

Manisha Natarajan, Esmaeil Seraj, Batuhan Altundas, Rohan Paleja, Sean Ye, Letian Chen, Reed Jensen, Kimberlee Chestnut Chang, and Matthew Gombolay. Human-robot teaming: grand challenges. *Current Robotics Reports*, 4(3):81–100, 2023.

Frans A Oliehoek, Matthijs TJ Spaan, and Nikos Vlassis. Optimal and approximate q-value functions for decentralized pomdps. *Journal of Artificial Intelligence Research*, 32:289–353, 2008.

Frans A Oliehoek, Christopher Amato, et al. *A concise introduction to decentralized POMDPs*, volume 1. Springer, 2016.

Afshin Oroojlooy and Davood Hajinezhad. A review of cooperative multi-agent deep reinforcement learning. *Applied Intelligence*, 53(11):13677–13722, 2023.

Rohan R Paleja, Michael Joseph Munje, Kimberlee Chestnut Chang, Reed Jensen, and Matthew Gombolay. Designs for enabling collaboration in human-machine teaming via interactive and explainable systems. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th annual acm symposium on user interface software and technology*, pages 1–22, 2023.

Jack Parker-Holder, Aldo Pacchiano, Krzysztof M Choromanski, and Stephen J Roberts. Effective diversity in population based reinforcement learning. *Advances in Neural Information Processing Systems*, 33:18050–18062, 2020.

Guannan Qu, Yiheng Lin, Adam Wierman, and Na Li. Scalable multi-agent reinforcement learning for networked systems with average reward. *Advances in Neural Information Processing Systems*, 33:2074–2086, 2020.

Nicholas Radcliffe. Using control groups to target on predicted lift: Building and assessing uplift model. *Direct Marketing Analytics Journal*, pages 14–21, 2007.

Roberta Raileanu and Tim Rocktäschel. Ride: Rewarding impact-driven exploration for procedurally-generated environments. In *International Conference on Learning Representations*, 2020.

Tabish Rashid, Mikayel Samvelyan, Christian Schroeder De Witt, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. Monotonic value function factorisation for deep multi-agent reinforcement learning. *Journal of Machine Learning Research*, 21(178):1–51, 2020.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Esmaeil Seraj, Rohan Paleja, Luis Pimentel, Kin Man Lee, Zheyuan Wang, Daniel Martin, Matthew Sklar, John Zhang, Zahi Kakish, and Matthew Gombolay. Heterogeneous policy networks for composite robot team communication and coordination. *IEEE Transactions on Robotics*, 40: 3833–3849, 2024.

Brennan Shacklett, Luc Guy Rosenzweig, Zhiqiang Xie, Bidipta Sarkar, Andrew Szot, Erik Wijmans, Vladlen Koltun, Dhruv Batra, and Kayvon Fatahalian. An extensible, data-oriented architecture for high-performance, many-world simulation. *ACM Transactions on Graphics (TOG)*, 42(4):1–13, 2023.

Ho Chit Siu, Jaime Daniel Pena, Edenna Chen, Yutai Zhou, Victor Lopez, Kyle Palko, Kimberlee Chestnut Chang, and Ross Emerson Allen. Evaluation of human-AI teams for learned and rule-based agents in hanabi. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021.

Kyunghwan Son, Daewoo Kim, Wan Ju Kang, David Earl Hostallero, and Yung Yi. Qtran: Learning to factorize with transformation for cooperative multi-agent reinforcement learning. In *International Conference on Machine Learning*, pages 5887–5896. PMLR, 2019.

Varshith Sreeramdass, Rohan R Paleja, Letian Chen, Sanne van Waveren, and Matthew Gombolay. Generalized behavior learning from diverse demonstrations. In *The Thirteenth International Conference on Learning Representations*, 2025.

DJ Strouse, Kevin McKee, Matt Botvinick, Edward Hughes, and Richard Everett. Collaborating with humans without human data. *Advances in Neural Information Processing Systems*, 34: 14502–14515, 2021.

Peter Sunehag, Guy Lever, Audrunas Gruslys, Wojciech Marian Czarnecki, Vinicius Zambaldi, Max Jaderberg, Marc Lanctot, Nicolas Sonnerat, Joel Z Leibo, Karl Tuyls, et al. Value-decomposition networks for cooperative multi-agent learning. *arXiv preprint arXiv:1706.05296*, 2017.

Haoran Tang, Rein Houthooft, Davis Foote, Adam Stooke, OpenAI Xi Chen, Yan Duan, John Schulman, Filip DeTurck, and Pieter Abbeel. # exploration: A study of count-based exploration for deep reinforcement learning. *Advances in Neural Information Processing Systems*, 30, 2017.

Tonghan Wang, Jianhao Wang, Yi Wu, and Chongjie Zhang. Influence-based multi-agent exploration. In *International Conference on Learning Representations*, 2020.

Xihuai Wang, Shao Zhang, Wenhao Zhang, Wentao Dong, Jingxiao Chen, Ying Wen, and Weinan Zhang. Zsc-eval: An evaluation toolkit and benchmark for multi-agent zero-shot coordination. *Advances in Neural Information Processing Systems*, 37:47344–47377, 2024.

Baicen Xiao, Bhaskar Ramasubramanian, and Radha Poovendran. Agent-temporal attention for reward redistribution in episodic multi-agent reinforcement learning. In *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems*, page 1391–1399. International Foundation for Autonomous Agents and Multiagent Systems, 2022.

Yaodong Yang, Rui Luo, Minne Li, Ming Zhou, Weinan Zhang, and Jun Wang. Mean field multi-agent reinforcement learning. In *International conference on machine learning*, pages 5571–5580. PMLR, 2018.

Chao Yu, Akash Velu, Eugene Vinitsky, Jiaxuan Gao, Yu Wang, Alexandre Bayen, and Yi Wu. The surprising effectiveness of ppo in cooperative multi-agent games. *Advances in Neural Information Processing Systems*, 35:24611–24624, 2022.

Rui Zhao, Jinming Song, Yufeng Yuan, Haifeng Hu, Yang Gao, Yi Wu, Zhongqian Sun, and Wei Yang. Maximum entropy population-based training for zero-shot human-ai coordination. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 6145–6153, 2023.

Lulu Zheng, Jiarui Chen, Jianhao Wang, Jiamin He, Yujing Hu, Yingfeng Chen, Changjie Fan, Yang Gao, and Chongjie Zhang. Episodic multi-agent reinforcement learning with curiosity-driven exploration. *Advances in Neural Information Processing Systems*, 34:3757–3769, 2021.

A Experimental Details and Environment Configurations

A.1 Layouts and Reward Shaping

Layout suite. We evaluate coordination in Overcooked layouts with 2, 3, and 4 agents (Fig. 2).

- **2-agent setting (Overcooked-AI).** Pipeline, Asymmetric Advantages, Forced Coordination, Cramped Room, and Coordination Ring. Counter Circuit is slightly revised and treated as a Pipeline variant to align with the multi-agent Pipeline settings.
- **3-agent setting.** Extends the above and adds Open Room, which increases interaction density and makes coordination less visually apparent. Asymmetric Advantages induces delayed teammate response because some agents cannot access key resources directly and must rely on others to stage items on shared counters. With three agents and two workstations, role allocation is underdetermined, increasing ambiguity in who should service which station and when.
- **4-agent setting.** Scales the passing-beneficial layouts to stress large-team coordination under frequent congestion.

Passing regimes. Layouts in the top row of Fig. 2 benefit from passing and staged handoffs, while those in the bottom row often make passing redundant and instead emphasize congestion and collision avoidance.

Reward and shaping. In all layouts, agents must coordinate to prepare and serve onion soup under standard Overcooked dynamics.

- **Sparse task reward.** Delivery of a completed three-onion soup yields 20.
- **Intermediate shaping.** Placing an onion into a pot yields $+3$; picking up a cooked soup yields $+5$; dish pickup shaping is 0.

This shaping eases exploration while preserving long-horizon coordination requirements, since high return still requires multi-step coordinated sequences to finish and deliver soups.

A.2 Hyperparameter Rationale

Table 2 summarizes the primary settings used across experiments.

- **Environment and rollout** uses $n_{\text{envs}} = 64$ parallel environments to reduce gradient variance while maintaining practical throughput, fixes the episode horizon at $H = 400$ to retain delayed coordination effects relevant to credit assignment, and collects $n_{\text{steps}} = 1024$ steps per environment per PPO update to cover diverse interaction contexts within each update.
- **PPO optimization** sets $\gamma = 0.99$ and $\lambda = 0.95$ to preserve long horizon returns while controlling variance, uses clip range 0.15 and target KL 0.025 to discourage destabilizing policy updates, and keeps entropy coefficient 0.05 to encourage early exploration in layouts that admit multiple coordination modes.
- **Learning rate choices and comparability** uses 3×10^{-4} for SP and FCP since a single shared policy is optimized and typically tolerates larger steps, uses 1×10^{-4} for CTDE backbones and ISCS augmented CTDE to improve stability under multi policy optimization, and keeps all remaining PPO settings identical between an ISCS run and its corresponding backbone so comparisons isolate shaping rather than optimizer changes.
- **ISCS posterior and event settings** trains the posterior online with one epoch per update and batch size 2048 to reduce variance and avoid overfitting transient rollouts, applies max grad norm 0.5 consistent with PPO for stability, uses zero label smoothing, and fixes event horizon at $K = 4$ as a short window that captures coordination relevant follow ups without conflating unrelated long delay effects.

Table 2: Key hyperparameters used across experiments.

PARAMETER	VALUE
ENVIRONMENT / ROLLOUT	
PARALLEL ENVS (n_{ENVS})	64
EPISODE HORIZON (H)	400
PPO ROLLOUT STEPS (n_{STEPS})	1024
TOTAL ENVIRONMENT STEPS	3×10^7
RANDOM SEEDS	12
PPO	
LEARNING RATE	3×10^{-4} (SP/FCP), 1×10^{-4} (MAPPO, HAPPO, MAPPO+ISCS, HAPPO+ISCS)
PPO EPOCHS (n_{EPOCHS})	8
BATCH SIZE	1024
CLIP RANGE	0.15
DISCOUNT (γ)	0.99
GAE λ	0.95
ENTROPY COEF.	0.05
MAX GRAD NORM	0.5
TARGET KL	0.025
ISCS POSTERIOR TRAINING AND EVENT SHAPING	
POSTERIOR LEARNING RATE	1×10^{-4}
POSTERIOR BATCH SIZE	2048
POSTERIOR EPOCHS / UPDATE	1
POSTERIOR MAX GRAD NORM	0.5
LABEL SMOOTHING	0.0
EVENT HORIZON (K)	4
ISCS BONUS COEFFICIENT (β)	5

A.3 CHOICE OF INTRINSIC SCALING COEFFICIENT β

Directed uplift is integrated as an intrinsic reward using the standard intrinsic motivation formulation $\hat{r}_{i,t} = r_t^{\text{env}} + \beta\, r_{i,t}^{\Delta p,(K)}$, where $\beta \geq 0$ controls the relative strength of shaping. Throughout this work we use the event-level intrinsic $r_{i,t}^{\Delta p,(K)}$ from Eq. (11); we write $r^{\Delta p,(K)}$ explicitly here to avoid confusion with the one-step variant. In this implementation, the uplift signal is a baseline adjusted probability lift, averaged over target teammates and computed over a short horizon coordination event. Because it is formed as a difference of predicted event probabilities and then averaged across targets, $r_{i,t}^{\Delta p,(K)}$ is naturally small and bounded. The coefficient is selected to keep shaping informative early in training while remaining secondary once task reward becomes dense.

Our choice $\beta = 5$ is primarily a reward-scale calibration. Across layouts, the *late-stage* per-step extrinsic reward (averaged over environments and steps) is typically on the order of 10^{-1} to 1 (roughly 0.15 to 1, depending on layout and team size), reflecting frequent deliveries and shaped intermediate progress. By contrast, the per-step intrinsic uplift term is typically on the order of 10^{-3} (often a few $\times 10^{-4}$ to 10^{-3}) after averaging across targets $j \neq i$. Multiplying by $\beta = 5$ keeps the intrinsic contribution on the order of a few $\times 10^{-3}$, which is small relative to the extrinsic signal once learning has reached the regime where task reward is informative.

This scale choice is especially important early in training. At initialization, deliveries are rare and the per-step extrinsic reward can be near 10^{-4} (or smaller) on average, so a modest intrinsic term can materially improve credit assignment and exploration by reinforcing influence-salient actions that increase the probability of short-horizon teammate follow-ups. As learning begins, we typically observe a rapid improvement phase where the per-step extrinsic reward rises into the 10^{-2} to 10^{-1} range. In this regime, the intrinsic uplift remains on the order of 5×10^{-3}, which corresponds to roughly a 15–20% contribution relative to extrinsic reward for most of this transition period. This ratio is large enough to provide a meaningful coordination bias when the extrinsic signal is still sparse and noisy, but small enough to avoid overwhelming the task objective. As training further stabilizes and the extrinsic reward reaches the 10^{-1} to 1 range, the same intrinsic scale becomes comparatively

minor, preserving the intended role of uplift as a shaping term rather than an alternate optimization target.

A.4 Behavioral Diversity Metric

Behavioral diversity is quantified using a determinant based population diversity formulation (Parker-Holder et al., 2020; Wang et al., 2024) that measures the volume induced by a similarity matrix over behavior features. The metric is used as a behavioral diagnostic of within team role differentiation in Pipeline.

- **Per agent behavior features** are computed separately for each agent and concatenated from two blocks that capture complementary aspects of coordination behavior.
- **Event rate block** counts coordination relevant interactions and converts them into per step rates over an episode. The feature vector includes item specific counts for drops, picks, and counter mediated handoffs received, as well as the number of serves. Item specific counts are tracked for onion, dish, and soup. Handoff reception is registered when one agent places an item on a counter and a different agent retrieves an item from the same counter within a short temporal window $\delta = 5$ steps.
- **Visitation block** captures spatial role specialization using an agent heatmap over grid cells. The heatmap is converted into a visitation distribution by normalizing by the total visitation mass and then flattened into a vector.
- **Block normalization** applies L2 normalization independently to the event rate block and the visitation block to prevent either interaction frequency or spatial coverage from dominating similarity.
- **Similarity matrices** are computed using cosine similarity within each block. Let θ_p^{event} and θ_p^{heat} denote the normalized event rate and visitation vectors for agent p. The blockwise similarity matrices are

  $$K_{pq}^{\text{event}} = \langle \theta_p^{\text{event}}, \theta_q^{\text{event}} \rangle \qquad K_{pq}^{\text{heat}} = \langle \theta_p^{\text{heat}}, \theta_q^{\text{heat}} \rangle.$$

  A mixed similarity matrix is then formed as

  $$K = \rho\, K^{\text{event}} + (1 - \rho)\, K^{\text{heat}},$$

  where $\rho \in [0, 1]$ controls the relative contribution of event based and visitation based similarity and is fixed to $\rho = 0.8$.
- **Within seed role diversity score** follows a stabilized determinant formulation. For a team with P agents, the behavioral diversity score is computed as

  $$\text{BD} = \log\det\big(I + \alpha K + \lambda I\big),$$

  where $I \in \mathbb{R}^{P \times P}$ is the identity matrix, $\alpha = 1$ scales the similarity matrix, and $\lambda = 10^{-6}$ is a small diagonal ridge for numerical stability. This stabilized log determinant is monotone in the determinant of the shifted matrix and provides a robust proxy for determinant based diversity when features are nearly collinear.
- **Interpretation** treats larger values of BD as stronger within team role differentiation under the chosen feature representation. The score increases when agents exhibit less similar event profiles and less similar visitation distributions, corresponding to more specialized roles and complementary coordination behavior in Pipeline.

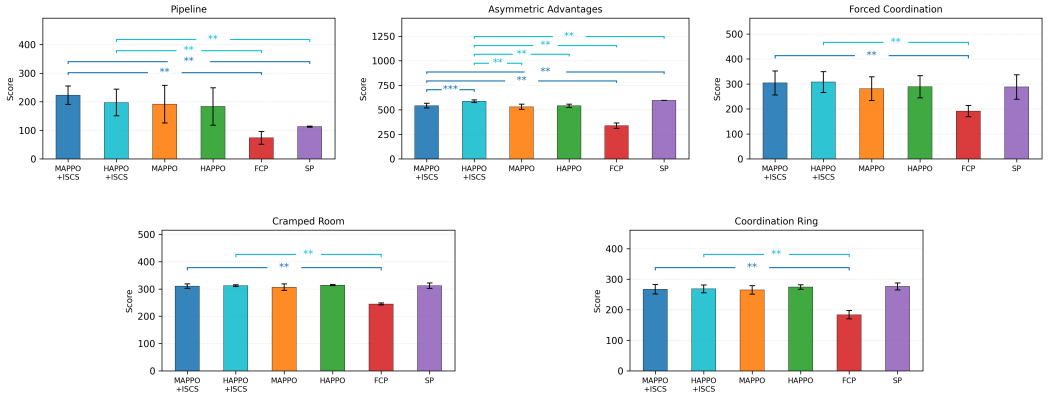


Figure 5: **Main results (2 agents) across layouts.** Top row: interaction-forward layouts where passing can be beneficial (Pipeline, Asymmetric Advantages, Forced Coordination). Bottom row: layouts where passing is less central and coordination is dominated by congestion/collision patterns (Cramped Room, Coordination Ring). Bars show mean final episode return across 12 seeds; error bars indicate ± 1 standard deviation. Brackets report paired Wilcoxon signed-rank tests between each ISCS-augmented approach (MAPPO+ISCS or HAPPO+ISCS) and each baseline, with Holm correction applied within each approach (per layout); when shown, MAPPO+ISCS vs. HAPPO+ISCS is uncorrected.

B Additional Results and Analysis

Table 3: Mean $\pm$ std over 12 seeds with maximum in parentheses for all layouts and team sizes where best mean is bolded.

Layout	MAPPO+ISCS	HAPPO+ISCS	MAPPO	HAPPO	FCP	SP	Heuristic
2-agent settings							
Pipeline	$\mathbf{222.7 \pm 32.1}$ (**290**)	197.2 ± 46.6 (247)	191.2 ± 65.9 (284)	183.3 ± 65.6 (290)	73.7 ± 22.3 (106)	112.8 ± 1.9 (118)	–
Asymmetric Advantages	542.4 ± 24.9 (590)	586.4 ± 14.0 (596)	530.2 ± 26.2 (584)	540.8 ± 17.2 (567)	337.2 ± 27.2 (363)	596.0 ± 0.0 (596)	–
Forced Coordination	304.2 ± 47.9 (358)	$\mathbf{307.7 \pm 41.9}$ (**349**)	280.8 ± 47.7 (349)	288.8 ± 44.7 (352)	191.2 ± 22.7 (216)	287.8 ± 49.0 (355)	–
Cramped Room	310.1 ± 8.2 (315)	312.0 ± 3.1 (315)	306.5 ± 12.1 (315)	314.0 ± 2.3 (315)	245.0 ± 3.5 (247)	312.2 ± 9.8 (315)	–
Coordination Ring	266.8 ± 15.5 (287)	268.2 ± 12.6 (281)	265.0 ± 13.9 (284)	274.1 ± 7.4 (281)	183.7 ± 13.5 (207)	276.2 ± 11.5 (284)	–
3-agent settings							
Pipeline	$\mathbf{155.5 \pm 48.4}$ (**272**)	136.6 ± 29.7 (207)	105.1 ± 16.8 (134)	121.3 ± 15.3 (150)	55.5 ± 7.8 (67)	119.1 ± 35.6 (204)	324
Asymmetric Advantages	195.6 ± 23.5 (252)	$\mathbf{200.8 \pm 15.5}$ (**218**)	176.7 ± 26.9 (213)	191.8 ± 17.5 (213)	117.7 ± 20.4 (144)	188.3 ± 34.3 (221)	290
Forced Coordination	330.7 ± 38.3 (391)	$\mathbf{337.6 \pm 44.6}$ (**417**)	283.9 ± 31.7 (380)	304.8 ± 44.2 (383)	197.0 ± 16.0 (213)	322.1 ± 36.5 (360)	426
Open Room	344.6 ± 16.4 (374)	360.4 ± 11.6 (380)	318.7 ± 50.5 (377)	353.4 ± 17.8 (386)	202.3 ± 22.4 (221)	379.8 ± 52.7 (423)	414
Coordination Ring	213.5 ± 10.3 (244)	209.5 ± 7.7 (218)	206.6 ± 10.9 (218)	203.8 ± 18.1 (244)	166.6 ± 15.0 (187)	262.5 ± 9.4 (278)	250
Cramped Room	432.0 ± 30.9 (485)	452.9 ± 28.4 (493)	388.1 ± 52.3 (451)	424.8 ± 59.9 (491)	271.8 ± 26.9 (345)	504.3 ± 28.6 (553)	346
4-agent settings							
Pipeline	112.0 ± 22.0 (139)	$\mathbf{128.3 \pm 23.7}$ (**155**)	97.8 ± 20.6 (136)	120.6 ± 21.5 (147)	66.9 ± 16.4 (95)	88.1 ± 25.2 (112)	–
Asymmetric Advantages	229.8 ± 18.8 (247)	$\mathbf{253.4 \pm 26.2}$ (**286**)	197.8 ± 35.3 (244)	225.5 ± 36.0 (281)	171.9 ± 16.1 (204)	173.8 ± 50.2 (218)	–
Forced Coordination	314.4 ± 24.8 (352)	$\mathbf{387.5 \pm 48.6}$ (**423**)	282.0 ± 22.5 (349)	373.3 ± 56.6 (420)	176.6 ± 17.9 (204)	348.3 ± 37.8 (402)	–

B.1 Additional Main Results Across All Layouts

We report full results for all evaluated layouts and team sizes in Fig. 5, Fig. 6, and Fig. 7, with corresponding numeric summaries in Table 3. Layouts are grouped by whether passing and staged handoffs are beneficial or largely redundant, which clarifies when directed short horizon coordination shaping has the most headroom.

- **Interaction forward layouts** Pipeline, Asymmetric Advantages, and Forced Coordination emphasize structured handoffs and delayed teammate responses. In these settings, adding ISCS to CTDE backbones more often changes the ordering among learned methods and yields the largest gains relative to the corresponding backbone.
- **Passing redundant and congestion dominated layouts** Cramped Room, Coordination Ring, and Open Room reduce the importance of explicit passing, with performance driven more by

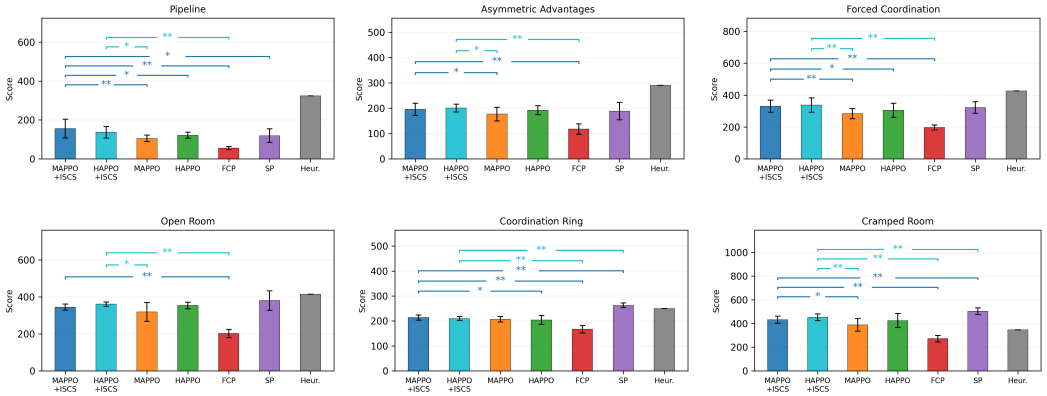


Figure 6: **Main results (3 agents) across layouts.** Top row: layouts where passing can be beneficial (Pipeline, Asymmetric Advantages, Forced Coordination). Bottom row: layouts where passing is redundant (Open Room, Coordination Ring, Cramped Room). Bars show mean final episode return across 12 seeds; error bars indicate ± 1 standard deviation. Brackets report paired Wilcoxon signed-rank tests between each ISCS-augmented approach (MAPPO+ISCS or HAPPO+ISCS) and each baseline, with Holm correction applied within each approach (per layout).

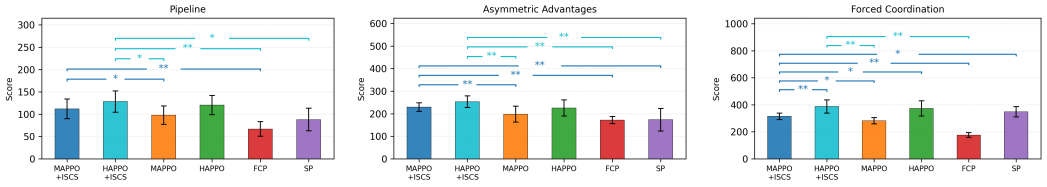


Figure 7: **Main results (4 agents) across layouts.** Layouts are interaction-forward and stress large-team coordination under congestion (Pipeline, Asymmetric Advantages, Forced Coordination). Bars show mean final episode return across 12 seeds; error bars indicate ± 1 standard deviation. Brackets report paired Wilcoxon signed-rank tests between each ISCS-augmented approach (MAPPO+ISCS or HAPPO+ISCS) and each baseline, with Holm correction applied within each approach (per layout); when shown, MAPPO+ISCS vs. HAPPO+ISCS is uncorrected.

collision avoidance and local throughput. In these settings, ISCS remains competitive while the strongest baseline can depend on how effectively the learned policy resolves congestion and approach tile contention.

B.2 Aggregate Summary of Best Performance

Across all layouts and team sizes, ISCS improves performance when added to CTDE backbones. HAPPO+ISCS achieves the best mean performance in all 4 agent settings and is strongest on Asymmetric Advantages and Forced Coordination in the 2 agent and 3 agent settings. MAPPO+ISCS is strongest on Pipeline in the 2 agent and 3 agent settings. These trends provide a compact summary of the full grid results, while the complete mean and variance values appear in Table 3.

B.3 Statistical Testing Protocol

For each layout and team size, methods are evaluated over 12 random seeds. Normality and homoscedasticity checks show violations in some comparisons at the 0.05 level. Given the small sample size and occasional assumption failures, paired Wilcoxon signed-rank tests with a two-sided alternative are used for all pairwise comparisons, pairing runs by seed. Within each layout, Holm correction is applied within each ISCS augmented approach across its four baseline comparisons. Asterisks in figures indicate Holm corrected significance using the thresholds below.

- $*$ with $p < 0.05$

- $**$ with $p < 0.01$
- $***$ with $p < 0.001$
- *ns* otherwise

B.4 Notes on Baselines and Evaluation Protocol

Our main results evaluate SP performance at test time, where agents are paired with themselves. We include fictitious co-play (FCP) (Strouse et al., 2021) as a representative partner-diversity baseline, but our protocol differs from the standard FCP evaluation objective in an important way. In Strouse et al. (2021), FCP is evaluated under *partner shift*, using a fixed population of unseen partners at test time, which directly measures robustness to heterogeneous teammates. In contrast, our primary figures measure SP test performance, which favors methods that specialize to cooperating with an identical copy of the learned policy.

- **FCP training protocol used in this work** For each seed i, a partner pool is constructed by sampling 12 self play training seeds and selecting 8 checkpoints from the self play population associated with seed i. A single FCP agent is then trained to perform well when paired with any of these 8 partner policies. For reporting, the trained FCP agent is evaluated in self play style by pairing it with the top performing agent among the same 8 checkpoints, and the resulting scalar return is reported in the main comparisons.
- **Behavioral consequence under the constructed pool** Because the partner pool includes early and mid training checkpoints that can be unreliable, the FCP agent is incentivized to adopt safer lower variance behaviors that are less dependent on timely teammate follow through. Empirically this often manifests as unilateral progress of the cooking pipeline, for example starting cooking early rather than waiting for partners to complete pot filling, and prioritizing short horizon shaped gains that are less likely to be blocked by partner interference. This strategy can reduce coordination failures against weak partners, but it can also suppress higher return interaction driven routines that require synchronized staging, handoffs, and delayed follow ups.

Overall, FCP underperforms self-play in our evaluation protocol. This does not contradict fictitious co-play results (Strouse et al., 2021), but reflects a mismatch between FCP's intended evaluation setting (robustness to unseen partners) and our primary objective (self-play test performance). We therefore treat FCP as a partner-diversity baseline representative, and interpret its self-play performance accordingly.

C Three-Agent Heuristic Controllers

All heuristics in this appendix are defined for the **three-agent** setting (PIDs 1–3) and are used as *plausible human-like baselines* rather than optimal solutions. Concretely, they encode a reasonable first-pass strategy a person might adopt after a brief inspection of the layout (e.g., fixed role split, simple staging/handoff rules, and conservative deconfliction), but they are not guaranteed to maximize reward.

C.1 Shared Controller Template (All Layouts, 3-Agent)

- **Action space** Each heuristic outputs one of six discrete actions $\{0 = \texttt{NORTH}, 1 = \texttt{SOUTH}, 2 = \texttt{EAST}, 3 = \texttt{WEST}, 4 = \texttt{STAY}, 5 = \texttt{INTERACT}\}$. Roles, when used, are fixed for the episode either by PID or by a simple spawn or lane rule to keep behavior deterministic.
- **Navigation and interaction** All layouts use an approach tile rule. Agents navigate only on walkable floor tiles and never path onto object tiles such as pots, sources, serving stations, or counters. To interact with an object at an adjacent cell, an agent stands on its designated approach tile, rotates to face the object, then issues `INTERACT`. If not facing correctly, the agent turns first. Some implementations optionally insert a single `STAY` before `INTERACT` to stabilize alignment.

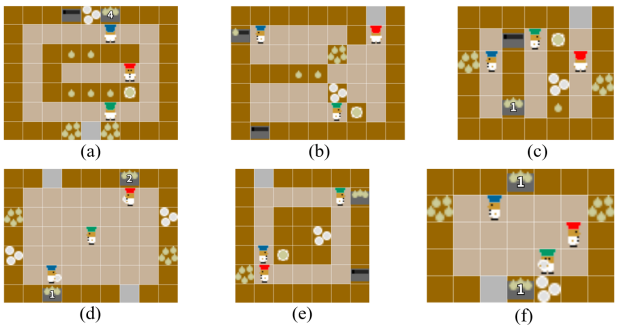


Figure 8: Overview of the six three-agent layouts discussed in Appendix C: (a) Pipeline, (b) Asymmetric Advantages, (c) Forced Coordination, (d) Open Room, (e) Coordination Ring, (f) Cramped Room.

- **Collision handling** Collision handling is layout specific. Some layouts rely on caller side arbitration using a central priority rule on same target or swap conflicts. Other layouts apply conservative local yielding, for example avoiding occupied cells or avoiding narrow corridor tiles when a finisher agent is active.

C.2 Heuristic Implementation (3-Agent Layouts)

- **Pipeline** enforces a streamlined passing invariant (Fig. 8a). Plated soups move downstream through fixed buffers, while onions move upstream. Agents commit to persistent lane roles so traffic is mostly one directional within each corridor. In steady state, the controller maintains buffer slack to keep handoffs feasible and prevent deadlocks caused by blocked drop and pick actions.
- **Asymmetric Advantages** is driven by delayed response dependencies and explicit staging counters (Fig. 8b). Fixed PID roles are assigned and single slot mailboxes are used for onions, dishes, and soups, with a strict no overwrite convention. A key ordering rule is onion first when a pot is idle and underfilled. Agents drop or defer non onion items that would otherwise block ingredient acquisition, then resume staging once cooking is underway.
- **Forced Coordination** uses a strict division of labor and a small set of single slot buffers (Fig. 8c), making cross agent dependence explicit. One agent prioritizes filling pots and starting cooking whenever a pot becomes full and idle. Another routes items, prefetches dishes during cooking, and plates ready pots. The third replenishes a shared onion buffer and prioritizes delivery. Minimal bookkeeping is used only when needed for deterministic routing, for example tracking which pot produced a soup so it can be staged to the correct pickup location.
- **Open Room** is organized as two symmetric pot owners plus one support runner (Fig. 8d). The owners each execute the same pot driven cycle. Fill to three onions, start cooking, prefetch a dish during cooking, plate when ready, then deliver. The runner supplies onions under a conservative insertion constraint, only when a target pot is idle and sufficiently underfilled, and yields on conflicts to avoid blocking owner interactions at pot approaches.
- **Coordination Ring** relies on two explicit handoffs with fixed role specific approach tiles to avoid contention (Fig. 8e). PID1 maintains an onion handoff counter and serves soup staged by PID3. PID2 runs the top pot end to end including serving its own soup. PID3 runs the bottom pot and stages finished soup to a dedicated counter with a no overwrite rule. To prevent deadlocks without blocking legal handoff interactions, the policy avoids only occupied cells locally, while a caller side central arbitration rule resolves same target or swap conflicts with priority PID3 $>$ PID1 $>$ PID2.
- **Cramped Room** is collision heavy, so the heuristic uses strict labor splitting with conservative local yielding (Fig. 8f). PID1 and PID3 are pot owners. They fetch onions, fill their assigned pot to three, and start cooking, preferably empty handed, then park while a finisher completes plating and delivery. PID2 is the dominant finisher. It acquires a dish as soon as any pot is cooking, plates whichever pot becomes ready, preferring the closer approach,

and delivers immediately. Owners avoid a small corridor set when PID2 is in a finishing phase, and PID3 additionally yields on a deterministic same target or swap rule versus PID2 to prevent repeated stalls.