TRAIT-CONSISTENT POPULATION MODELING FOR HUMAN–AI ZERO-SHOT COOPERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Human–AI and multi-agent zero-shot cooperation (ZSC) requires agents that generalize to unseen human and novel agent behaviors. Population-based training is a central paradigm for ZSC, with the key challenge of designing training populations that capture the structured diversity of human behavior. To cope with this challenge, we introduce **Trait-Embedded Agent Modeling (TEAM)**, a framework for inducing human-aligned behaviors in the training population. Drawing on studies of human gameplay, particularly the widely adopted Bartle's taxonomy, our proposed population exhibits traits inspired by general behavioral patterns observed in social games, without requiring task-specific reward modifications. We instantiate this idea through three lightweight modifications to the learning objectives, yielding a population of socializers, achievers, and explorers. To support our approach of pattern-focused diversity, we formalize ZSC as training on a set of Markov decision processes (MDPs), with a newly sampled MDP at test time. This connects ZSC to existing generalization theory and quantitatively motivates focused population coverage. Our evaluation shows that TEAM outperforms strong baselines on AI-AI and real-human tests on challenging ZSC benchmarks, including Overcooked and StarCraft. We further provide an empirical space-coverage study of the trained population, showing TEAM induces semantically meaningful and interpretable state-action space coverage.

1 INTRODUCTION

A key challenge in generalization of multi-agent systems is test-time robustness to shifts in partner behavior (Leibo et al., 2021), commonly termed zero-shot cooperation (ZSC) (Hu et al., 2020). ZSC is central to human-AI settings where agents must collaborate with unseen human partners without access to human data during training. ZSC is commonly tackled in a two-phase scheme, where a population of agents is first learned and then used to train the ego agent, a generalist policy that will be deployed against unseen partners at test time.

Recent approaches often fit in this paradigm by promoting diversity in the training population. To achieve this, prior ZSC techniques typically strive for high space coverage, for example through entropy-based regularization (Lou et al., 2023; Zhao et al., 2023) or via encouraging incompatibility between agents (Sarkar et al., 2023). Yet, such approaches may not faithfully capture human behavior. A recent human-centric method (Yu et al., 2023) addresses this gap by constructing rewards that encode human-like preferences, mimicking behavioral biases (e.g., some football players prioritize scoring, others prioritize coordination and passing to teammates). While effective, this strategy depends on substantial domain knowledge and carefully engineered, task-specific rewards.

Motivated by the promise of the human-centric approach, we propose **Trait-Embedded Agent Modeling (TEAM)**, a task-agnostic framework for inducing human-inspired trait structure in training populations. Our key idea is to model and promote diversity through task-agnostic behavioral traits that consistently shape agent behavior across episodes, requiring little to no model knowledge or task-specific implementation. TEAM instantiates this idea through lightweight modifications to the learning objectives that induce three canonical traits: socializers, achievers, and explorers. These traits are grounded in established models of human behavior in social games, particularly the foundational Bartle's taxonomy (Bartle, 1996) (Fig.1), which has been widely used to characterize player motivations across various settings (Zackariasson et al., 2010; Ferro et al., 2013).

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

By adopting this perspective, TEAM induces an interpretable diversity that captures human-inspired patterns without task-specific reward engineering.

Yet, a key question remains: is it more effective to constrain a population to specific behavioral patterns than to encourage arbitrary behavioral diversity? To answer this in an algorithm-agnostic manner, we formalize ZSC as training on a set of MDPs, and tap into existing (domain randomization) generalization theory (Chen et al., 2022). This yields an upper bound on ego test-time performance, showing that as the population scales, a sufficient fraction of its members must remain close to the hypothesized test distribution to guarantee ego performance. This result, while intuitive, suggests that ZSC methods should not only strive for wide coverage, as many works advocate (Zhao et al., 2023; Yan et al., 2023), but also emphasize focused behavior patterns within the population.

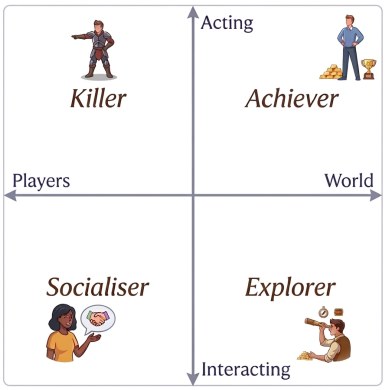


Figure 1: Bartle's player axes (Bartle, 1996), illustrating the two dimensions that induce behavioral archetypes. E.g., explorers are produced by an agent emphasizing interaction (Y axis) with the world (X axis).

Our empirical evaluation assesses two key hypotheses: (i) Enforcing the suggested traits during population training yields an interpretable and trait-focused coverage of the state and action spaces, and (ii) Training the ego agent with partners confined to our proposed traits over entire episodes leads to more effective ZSC with both humans and AI collaborators. To verify the second hypothesis, we adopt three evaluation protocols commonly used in the literature (Jha et al., 2025; Yan et al., 2023; Wang et al., 2025): cross-seed, cross-algorithm, and human studies, across all of which TEAM consistently outperforms leading baselines (Jha et al., 2025; Zhao et al., 2023; Yan et al., 2023; Yu et al., 2023). Our key contributions are summarized as follows:

- We introduce a **conceptual ZSC methodology** in which population diversity is modeled as a set of human-aligned behavioral identities.
- Based on this methodology, we propose TEAM, **a task-agnostic framework for promoting behavioral spectrum** across achievers, socializers, and explorers.
- By **reformulating ZSC**, we demonstrate the benefits of constraining the population to a set of hypothesized test-time behaviors.
- Through **extensive empirical study**, we show that our proposed traits yield structured, interpretable behavioral patterns across the state–action space.

2 Problem Formulation and Background

Problem Formulation. **(Phase 1).** During population training, each member is trained with a designated partner on a two-player episodic Markov game $\langle S, A, P, \mathcal{R}_{tot}, H, \mu \rangle$ with state space S, action space A with the joint action at each timestep given by $\boldsymbol{a}^t = (a_i^t, a_{-i}^t)$, where $-i$ denotes the other player's index. Joint transition kernel P, task reward $\mathcal{R}_{tot}$, horizon H, and initial state distribution $s_1 \sim \mu \in \Delta(S)$. The joint policy is denoted by $\boldsymbol{\pi} = (\pi_i, \pi_{-i})$ with parameters (θ_i, θ_{-i}). **(Phase 2).** Once completed, phase 1 yields a population of K trait-based partners $\mathcal{P}_K = \{\pi_{\theta_1}, \ldots, \pi_{\theta_K}\}$. Since the population is stationary by design and frozen during phase 2, each member can be absorbed into the environment, inducing a single-agent episodic MDP for the ego: $M_{\theta_i} = \langle S,\ A,\ P^{\theta_i},\ R^{\theta_i},\ H,\ \mu \rangle$, where $P^{\theta_i}(s' \mid s, \boldsymbol{a}) = \sum_{a_i \in A} P(s' \mid s, a_{\text{ego}}, a_i)\, \pi_{\theta_i}(a_i \mid s)$. a_{ego} denotes the ego's action, a_i the partner's, and $R^{\theta_i}(s, a) = \sum_{a_i \in A} R(s, a_{\text{ego}}, a_i)\, \pi_{\theta_i}(a_i \mid s)$. We write $\mathcal{U} = \{M_{\theta_i} : \theta_i \in \Theta\}$, where Θ is the space of all possible partners. At test time, the ego, an adaptive history-dependent policy, is paired with a human policy π_H, which induces MDP M^* with state distribution ρ^H. Following (Yu et al., 2023), we model the human as a stationary policy so that it can be absorbed into the MDP. For a history-dependent ego policy $\pi_e \in \Pi$, the value function in the induced MDP M_θ is $V^{\pi_e}_{M_\theta, h}(s) = \mathbb{E}_{M_\theta, \pi_e}\left[\sum_{t=h}^{H} R^\theta(s_t, a_t) \;\middle|\; s_h = s \right]$.

Distance between MDPs. We define the distance between two induced MDPs through the partner

policies that generate them. For a state distribution $\rho \in \Delta(S)$, we define the distance between two MDPs M_{θ_1} and M_{θ_2} as: $d_\rho\big(M_{\theta_1}, M_{\theta_2}\big) = \mathbb{E}_{s\sim\rho}\Big[\, \big\|\pi_{\theta_1}(\cdot \mid s) - \pi_{\theta_2}(\cdot \mid s)\big\|_1 \,\Big]$.

MDP neighborhood. For a human-induced MDP M^* and a radius $\epsilon > 0$, the ϵ-neighborhood of $M^* \in \mathcal{U}$ is: $C_{M^*,\epsilon} = \big\{\, M_\theta \in \mathcal{U} \;:\; d_\rho(M_\theta, M^*) \le \epsilon \,\big\}$.

Empowerment. Empowerment quantifies an agent's influence over future states. From agent i's perspective, it is the conditional mutual information (MI) between the partner's action and a future state s^{t+k} with $k \sim \text{Geom}(p)$ and $p \in (0,1)$, summed over time $\mathcal{E}(\pi_i, \pi_{-i}) = \mathbb{E}_{\pi_i,\pi_{-i}}\big[\sum_{t=0}^{\infty} \gamma^t\, I(a^t_{-i}; s^{t+k} \mid s^t)\big]$, where $I(a^t_{-i}; s^{t+k} \mid s^t) = \mathbb{E}\Big[\log \frac{p(s^{t+k}|s^t, a^t_{-i})}{p(s^{t+k}|s^t)}\Big]$, and $\mathcal{E}_i = \mathcal{E}(\pi_i, \pi_{-i})$. Maximizing $\mathcal{E}_i$ thus lets an assistive agent help a partner without inferring any reward, by shaping the environment to amplify the partner's influence over future outcomes s^+. $\mathcal{E}$ is estimated via contrastive successor representations (Myers et al., 2024) using three encoders $\phi(s, a_i, a_{-i})$, $\phi'(s, a_i)$, $\psi(s^+)$, which embed the partner dynamics and future states and form a reward signal for the agent: $r_i = (\phi(s, a_i, a_{-i}) - \phi'(s, a_i))^\top \psi(s^+) \approx \log \frac{p(s^{t+k}|s^t, a^t_{-i})}{p(s^{t+k}|s^t)} + c$, optimized via RL.

3 Related Work

This section surveys relevant works according to their approach to solving the ZSC challenge. We also include works from outside the ZSC domain that deal with mutual-information-based rewards.

Population diversity. MEP (Zhao et al., 2023) promotes diversity in the training population through entropy-based exploration, measured by the population mean policy entropy at each step. While MEP focuses on population diversity through an information-theoretic objective, we use reward-based regularization to encourage a broader range of human-inspired behaviors. HSP (Yu et al., 2023) promotes diversity through event-based rewards, assigning each agent a randomized reward function such as prioritizing pot placement versus ingredient chopping. While HSP relies on domain knowledge and hand-crafted rewards, our approach promotes diversity through task-agnostic objectives, avoiding this design overhead. CoMeDi (Sarkar et al., 2023) generates diverse coordination conventions by maximizing self-play reward while minimizing cross-play reward with previously discovered conventions. This encourages the population to cover diverse strategies using reward-based incompatibility, while TEAM instead imposes diversity through human-grounded priors. GAMMA (Liang et al., 2024) collects trajectories from populations trained with methods such as MEP and learns an encoder-decoder to generate new trajectories by combining and interpolating existing strategies. GAMMA agents emerge from behaviors covered by existing baselines, whereas TEAM partners are prescribed by human-inspired semantics.

Environment diversity. CEC (Jha et al., 2025) proposes solving two interconnected problems: ZSC and generalization to novel unseen tasks. Their approach suggests focusing on environmental diversity, rather than population diversity, which allows the authors to use cheaper self-play agents without creating a training population. They propose exposing self-play agents to diverse environmental settings (e.g., wall positions) and showed that it results in higher ZSC with real humans.

Population diversity with inference-based (ego) agent. E3T (Yan et al., 2023) reduces population requirements by training with a single stochastic partner policy while adding a partner modeling module to predict actions. E3T differs from TEAM as it relies solely on randomization techniques, while we utilize additional techniques for non-explorer traits (e.g., achievers via workload adjustment).

Other works (empowered MI-based). (Myers et al., 2024) replace the reward with an MI objective between the human's actions and subsequent global states, encouraging agents to increase their teammate's MI to improve coordination. In contrast to (Myers et al., 2024) and other MI-based methods (e.g., (Li et al., 2022)), which use MI to optimize collaboration, we use MI as a bonus to induce diversity via varying levels of socialization in ZSC, rather than improving learning in self-play.

4 Human Behavioral Taxonomy

Our work draws on the foundational framework for modeling human player behavior through high-level patterns, introduced by (Bartle, 1996). Bartle categorized player motivations and behaviors based on their style of engagement and primary in-game focus. Bartle's axes (depicted in Fig. 1) assign players along two axes: action vs. interaction, and player vs. world-orientation, yielding four archetypes: achievers, explorers, socializers, and killers. Bartle originally proposed these axes based

on observations of Multi-User Dungeons (MUDs), early text-based online role-playing environments. Subsequent research has identified the same player traits in more recent social online games, including EverQuest (Yee, 2006), World of Warcraft (Zackariasson et al., 2010), and Ultima Online (Hamari & Tuunanen, 2014), providing empirical support for Bartle's original framework.
Synthesizing these studies, Hamari and Tuunanen (Hamari & Tuunanen, 2014) conducted a meta-review and proposed a unified taxonomy consisting of seven player types/traits: achievement, exploration, sociability, domination, immersion, along with gaming intensity and in-game demographics, where demographics identify players based on role (e.g., warrior/magician) or proficiency. This taxonomy largely preserves Bartle's original categories, but generalizes the "Killer" into the broader concept of *domination*. Appendix C provides an extended discussion on player types in online games. From this taxonomy, we focus on the types most relevant to our work and social online games:

Achiever: Goal-oriented individuals motivated by long-term success, task completion, and the accumulation of in-game wealth or status (Nacke et al., 2011). Such players may exhibit additional characteristics such as an interest in analyzing the underlying rules in order to optimize performance.
Explorer: Curiosity-driven players who enjoy discovery, mapping the topology of the environment, and experimenting with the underlying "physics" or rules of the game world (Bartle, 1996; Hamari & Tuunanen, 2014). Such players prefer interaction with both the world and other players.
Socializer: Social players are motivated by the relational connections and interactions with other players (Bartle, 1996). Such interaction can be divided into several sub-categories (Zackariasson et al., 2010): pure socializing (e.g., chats), relational (e.g., help and support), and teamwork that leads to group achievements. Our focus is on relational connections that directly affect teamwork.
"Killer". To define the killer trait, we first introduce the notion of a dominant player. Dominant individuals tend to assert themselves or impose their will through assertive and fearless behavior (Moynihan et al., 2023). In gameplay, this tendency is often associated with aggressive gamers (Hamari & Tuunanen, 2014). When such behavior involves seeking to stress a teammate and establish superiority in a manner that may harm others, the actor can be characterized as a killer (Bartle, 1996).

5 Methodology: Trait-Embedded Agent Modeling

This section presents our approach, Trait-Embedded Agent Modeling (TEAM). To achieve ZSC and effective human-AI interaction, we rely on a two-phase scheme (Fig. 2): First, train a diverse population of agents based on human-centered traits, promoting each agent to exhibit its designated trait consistently throughout the episode; second, use this population to train an adaptive ego agent that can collaborate well with all agents in the population, aiming at improving its generalization ability to unknown partners. The training algorithm is presented in Appendix E.

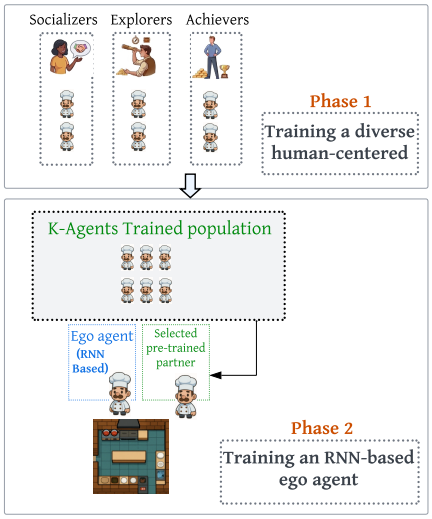


Figure 2: Our proposed workflow, which comprises trait-based population training (phase 1) and ego training (phase 2).

Trait Abstraction and Design Principles. Our framework is inspired by the taxonomy in Sec. 4, which we use as guidelines for structuring our mechanism. Since these traits were identified across multiple social games, we rely on their generality and assume they align with human patterns in general collaborative games (e.g., Overcooked). Note that some traits comprise several characteristics, but we focus on those we found most effective at generating behavioral diversity (e.g., socializer through its assistive aspect). Alternative implementations are also plausible, depending on which of these characteristics are prioritized.

5.1 Phase 1 (Population Training)

In the first phase, we generate a population of total K agents. We construct three agent types: socializers, achievers, and explorers, as detailed below. We focus on three traits, as the fourth Bartle

archetype, killer, in its milder form overlaps with the negative achiever coefficient (presented below). In its more aggressive form, it may conflict with the cooperative objective (discussion in Sec. 8).

Socializer (help and support). Our socializer agent focuses on relational interactions that enable it to assist and support its teammates in accomplishing their goals. To operationalize this idea at the algorithmic level, we impose two conditions: (i) the agent must coordinate effectively with its partner, and (ii) this coordination should lead to an increase in the partner's reward. To this end, we adopt the empowerment objective (Myers et al., 2024). Empowerment encourages an agent to maximize the mutual information (MI) between its teammate's actions and future states, thereby increasing the teammate's control over the environment and facilitating task completion. Our experiments showed that combining this objective with the original environment reward, instead of replacing it, as in (Myers et al., 2024), induces a richer set of behaviors while still constraining the agent to an assistive role as in the original paper. Assuming $\alpha_i \in [0, 1]$, the i-th agent reward is:

$$R_i^{\text{soc}}(s^t, \boldsymbol{a}^t; \alpha_i) = (1 - \alpha_i) R^{\text{tot}}(s^t, \boldsymbol{a}^t) + \alpha_i \mathrm{I}(a_{-i}^t; s^{t+k} \mid s^t), \tag{1}$$

where $k \sim \text{Geom}(p)$ and $p \in (0, 1)$. Notably, unlike prior work that uses empowerment to improve learning efficiency (Myers et al., 2024), we employ it to promote behavioral diversity within a population and induce varying levels of assistiveness.

Achiever. An achiever-type agent is primarily driven by successful task completion, while prioritizing behaviors that maximize its individual role in the joint outcome, thus increasing credit assignment to its own actions. To realize this type and induce varying degrees of achiever behavior, we propose manipulating the reward allocation between the achiever and its partner, which adjusts their respective contributions to the joint objective. An i_{th} achiever learns by maximizing the following reward:

$$R_i^{\text{achiever}}(s^t, \boldsymbol{a}^t) = R_i(s^t, \boldsymbol{a}^t) + \alpha_i R_{-i}(s^t, \boldsymbol{a}^t), \tag{2}$$

where $J_i^{\text{achiever}}(\boldsymbol{\pi}; \alpha_i) = \mathbb{E}_{\pi_i, \pi_{-i}} \left[\sum_{t=0}^{T} \gamma^t R_i^{\text{achiever}}(s^t, \boldsymbol{a}^t) \right]$, and the allocation is controlled via $\alpha_i \in [-0.25, 1]$. As the partner's contribution to task completion decreases, the achiever is encouraged to compensate by assuming a higher workload. The partner $(-i)$ reward is similarly adjusted:

$$R_{-i}^{\text{achiever}}(s^t, \boldsymbol{a}^t) = R_i(s^t, \boldsymbol{a}^t) + \beta_i R_{-i}(s^t, \boldsymbol{a}^t), \tag{3}$$

where $\beta_i = \max(\alpha_i, 0.01)$. Setting $\beta_i \geq 0.01$ prevents a "lazy agent" phenomenon (Sunehag et al., 2017) where the partner isn't motivated to contribute to task completion.
As a complementary approach, for settings where explicit reward allocation is not feasible (e.g., SMAC, where a single reward signal is given for all agents), we propose an additional technique to tune contribution by training the partner (π_{-i}) with a reduced learning rate (LR) compared to the achiever LR. This yields a lower-proficiency partner with decreased contribution to task completion, encouraging higher credit assignment to the achiever.

Explorer. We use the term explorer to refer to an agent that pursues novel solution strategies and prioritizes novelty over reward maximization. This approach may result in lower rewards as long as the agent promotes highly diverse and unconventional solution patterns. To achieve this, we tune the entropy coefficient in the PPO algorithm via α_i but use high temperature values ($\alpha_i \geq 0.1$ compared to the standard 0.01). In our settings, the entropy-regularized PPO loss is given by:

$$\mathcal{L}^{\text{PPO}}(\theta_i) = \mathbb{E}_t \Big[\mathcal{L}^{\text{PPO-clipped}}(\theta) - \alpha_i \, \mathcal{H}[\pi_\theta(\cdot|s_t)] \Big], \tag{4}$$

where: $\mathcal{L}^{\text{PPO-clipped}}(\theta_i) = \min \Big(r_t(\theta_i)\hat{A}_t, \, \text{clip}(r_t(\theta), 1 - \epsilon, 1 + \epsilon)\hat{A}_t \Big)$. $r_t(\theta)$ is the probability ratio, $\hat{A}_t$ is the advantage estimate, ϵ is the clipping parameter, $\mathcal{H}$ denotes the policy entropy, and α_i is the entropy coefficient tuned for the explorer agent. Unlike other entropy-based methods (e.g., MEP) which encourage agents to deviate from the population mean policy, our explorer is agent-intrinsic, encouraging each agent to maintain a diffuse action distribution independent of other agents.

5.2 Training procedure

In Phase 1, achievers and explorers are trained from scratch, each with its own designated partner. For socializers, we follow the original empowerment protocol: a partner is pre-trained via self-play

Table 1: Results on AI-AI evaluation. Values report mean ± standard error over 20 seeds, combining cross-play and cross-seed evaluation.

Environment	TEAM (ours)	FCP	MEP	HSP	E3T
Coin Game	$\mathbf{30.7 \pm 2.2}$	18.97 ± 3.1	14.1 ± 2.2	26.1 ± 2.0	11.3 ± 2.3
SMAC (2s vs 3z), win rate	$\mathbf{0.59 \pm 0.11}$	0.509 ± 0.11	0.13 ± 0.1	0.505 ± 0.11	0.13 ± 0.09
SMAC (2m vs 2z), win rate	$\mathbf{0.626 \pm 0.11}$	0.504 ± 0.11	0.143 ± 0.07	0.39 ± 0.1	0.15 ± 0.11
Overcooked v2 (demo wide)	$\mathbf{108.7 \pm 3.81}$	99.99 ± 4	102.8 ± 2.1	101.9 ± 3.1	90.7 ± 1.51
Overcooked v2 (AA, recipe center)	$\mathbf{237.1 \pm 6.6}$	152.4 ± 18.3	221.0 ± 8.9	212.6 ± 11.2	197.2 ± 4.9

and then paired with a newly initialized socializer, while keeping the partner fixed. This design leads to more meaningful empowerment behavior, likely due to more stable reward signals, as the partner policy remains stationary. For all types, we follow the procedure suggested by previous works (Zhao et al., 2023; Strouse et al., 2021) and take 3 checkpoints during training (init, mid, fully trained) that will later be used to train the ego agent. Four agents in the population train via mixed traits (e.g., socializes with explorer).

In phase 2, the ego agent is paired with the phase 1 population. We additionally include a self-play round in which the ego is trained against itself. The self play round get $\frac{1}{\text{population size}}$ of the total training. Similar to previous works, our ego is trained by pairing it with a different member from the (phase 1) trained population in each episode. We use an RNN, shared between the actor and critic and enables better adaptability via a memory mechanism. Full details are given in Appendix D.

6 Generalization Bound on Ego Performance

This section motivates confinement of the population around the hypothesized test-time distribution. Building on the formalization we introduced in Sec. 2, we tap into generalization theory from single-agent domain randomization (Chen et al., 2022), which provides an upper bound on ego performance in a general ZSC setting (Theorem 1). We first make the following assumptions, derived from (Chen et al., 2022). The assumptions introduce optimality/boundedness, which we use only for analysis purposes.

Assumption 1. *Given a phase* 1 *population, an ego is optimal with respect to the population-averaged return, that is:* $\pi_e^* \;=\; \arg\max_{\pi \in \Pi} \; \mathbb{E}_\theta \, \mathbb{E}_{s_1 \sim \mu}\big[\, V^{\pi}_{M_\theta,1}(s_1)\,\big]$.

Here, Π is the space of all possible history-dependent policies. Assumption 1 is reasonable in our setting, since our ego is (i) LSTM-based, which is essential for the generalization analysis, and (ii) trains simultaneously with the entire population, yielding a best response policy.

Assumption 2. *There exist* $\epsilon_0 > 0$ *and a Lipschitz constant* $L < \infty$ *such that for the trained ego* π_e*, the value function for any two MDPs in* C_{M^*,ϵ_0} *is* L*-Lipschitz w.r.t the distance function* d*:*

$$\Big|V^{\pi_e}_{M_{\theta_1},1}(s_1) - V^{\pi_e}_{M_{\theta_2},1}(s_1)\Big| \;\le\; L \cdot d_\rho\big(M_{\theta_1}, M_{\theta_2}\big) \qquad \forall\, M_{\theta_1}, M_{\theta_2} \in C_{M^*,\epsilon_0}, \tag{5}$$

where C_{M^*,ϵ_0} and d as defined in Sec.2. A worst-case derivation of L is in Appendix G. Assumption 2 means the ego performance with a test partner in the ϵ_0 ball satisfies smoothness. In practice, this assumption ensures that if a training partner is close to the human partner, the ego gets a similar value against both.

Theorem 1 (Upper bound on ego performance gap)**.** *Let Assumptions 1 and 2 hold and suppose* $M^* \in \mathcal{U}$*, where* M^* *is the MDP induced by the human partner* π_H *and* V^* *denotes the value of the optimal policy. Let* $N_\epsilon = \big|\{\, i \in [K] \;:\; d_\rho(M_{\theta_i}, M^*) \le \epsilon \,\}\big|$*, where* K *and* N_ϵ *are the total number of population members and the number of members whose induced MDP lies within* ϵ *of* M^**, respectively. Assuming that once trained in phase 1, the partners are sampled uniformly, then for every* $0 \le \epsilon < \epsilon_0$ *and a starting state* s_1*, the ego* π_e^* *satisfies:*

$$V^*_{M^*,1}(s_1) - V^{\pi_e^*}_{M^*,1}(s_1) \;=\; \frac{K}{\mathbb{E}_\theta[N_\epsilon]} \cdot O\!\left(D\sqrt{\hat{d}_e\, H\, \log\big(H \cdot \mathcal{N}_H\big)}\right) \;+\; 2L\,\epsilon. \tag{6}$$

$\mathcal{N}_H$ is the number of population members (with induced MDPs) whose behavior differs by more than $1/H$ per step. $\hat{d}_e$ is the number of state-action pairs the ego needs to visit before it can tell its partner

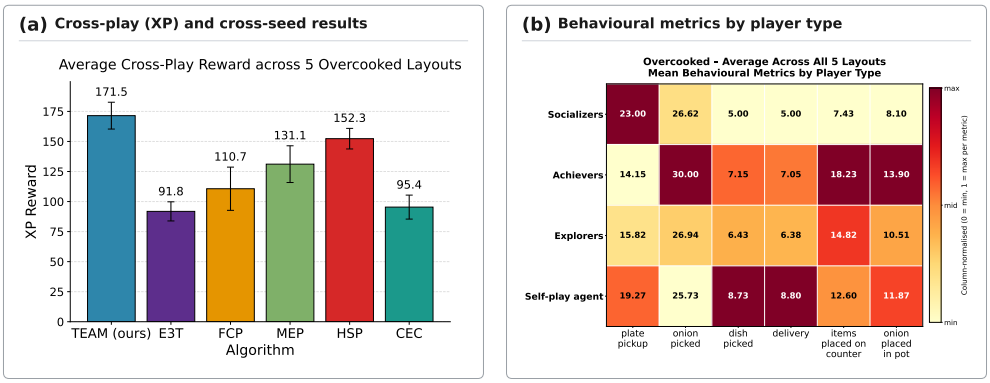


Figure 3: **(a)** Average results over the five Overcooked layouts under cross-play and cross-seed evaluation. **(b)** Heatmap of behavioural differences between traits, measured as the number of times each action category was selected per episode.

apart from every other member of the class of training MDPs. D is the maximum expected time required to reach any state from any other state under a stationary policy that minimizes this time. A detailed mathematical formulation of $\mathcal{N}_H, d_e, D$ with Theorem 1 specifications is in Appendix G.

Interpretation and motivation for TEAM. For a fixed maximum distance ϵ between the human-induced MDP and training MDPs, the gap decreases with the fraction $\frac{K}{\mathbb{E}_\theta[N\epsilon]}$ of training partners within the ϵ-neighborhood, formalizing the benefit of human-centered populations for generalization. Conversely, spreading K agents over the space to achieve coverage increases the required ϵ and thus the $L\epsilon$ term, weakening the guarantee. Note that the bound applies to a general ego agent rather than the TEAM ego, providing an algorithm-agnostic motivation for maintaining proximity to human and test partners, even at the expense of population spread. Empirical assessment of proximity between TEAM and baseline populations to human players with $\frac{K}{N_\epsilon}$ measurements is in the evaluation.

Dimensionality perspective. The "coverage VS. proximity" tradeoff can also be viewed through the lens of dimensionality. Assume each player is represented by some d traits, where each trait is sampled as $\alpha_i \in [0, L]$. An ϵ-coverage of all traits require $(\frac{L}{\epsilon})^d$ agents, which in the worst case can be exponential in the state cardinality. The TEAM approach reduced dimensionality to a 3-dimensional space by setting $d = 3$, that we show produces good results despite trading off coverage.

7 Evaluation

This section evaluates two core hypotheses: (i) TEAM induces diverse and interpretable coverage of the state space, and (ii) A trait-consistent population improves AI-AI and human–AI generalization. We compare our approach to MEP (Zhao et al., 2023), CEC (Jha et al., 2025), FCP (Strouse et al., 2021), E3T (Yan et al., 2023), and HSP (Yan et al., 2023). We reimplemented MEP and HSP in JAX using the hyperparameters reported in their paper and accompanying code. The implementations of E3T, CEC, FCP and the human interface are based on (Zhao et al., 2023). For fair comparison, we used the same hyperparameters (e.g., learning rate, population training time, network architecture, etc), for both the MLP-based population and the ego across all methods. Full details are given in Appendix D. All source code will be made available on publication under an open-source license. Our experiments follow three evaluation protocols commonly used in the literature (Jha et al., 2025; Yan et al., 2023; Zhao et al., 2023): cross-seed, cross-play, and human studies. In cross-seed, an ego is evaluated against egos from the same methods that were trained with different seeds. In cross-play, the ego agent of each method is evaluated versus all the ego agents of the other methods, across all seeds. Cross play/seed experiments use 20 random seeds, while human experiments use 5. Human experiments are conducted on the Overcooked V1 benchmark (Strouse et al., 2021). The AI-AI evaluation is conducted on Overcooked V1, Overcooked V2 (Gessler et al., 2025), StarCraft (SMAX), and the coin game (Raileanu et al., 2018), based on the JAX implementation (Rutherford et al., 2024).

Overcooked V1. Overcooked is a cooperative cooking benchmark widely used for ZSC human–AI

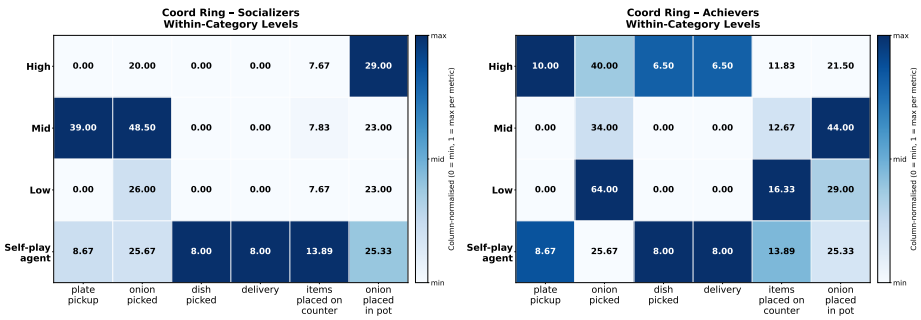


Figure 4: Studying intra-trait diversity on the Coord Ring map for achievers and socializers. For each trait, α_i values are grouped into low, mid, and high levels to analyze emergent behavior. Numbers in each cell denote the number of times an action was performed during an episode

coordination, in which two agents share a kitchen and must coordinate to prepare and deliver onion soups under a time limit. It includes several layouts that impose different coordination challenges.
Overcooked V2. A structurally different version of the benchmark with partial observability, multiple ingredients recipe, and asymmetric information, where recipes are observable by only one agent.
Coin game. A two-player social dilemma in which collecting one's colored coin rewards only the acting agent, while collecting the opponent's coin rewards the collector and penalizes the opponent.
Starcraft. A popular benchmark in which a team of allies must cooperatively defeat an enemy team.

Q_1: Does enforcing our suggested traits during population training yield a trait-focused and interpretable coverage of the state and action spaces? Overcooked V1 results. Fig. 3(b) reports a behavioral heatmap over complete episodes, averaged across all five layouts and all different levels within a trait. A self-play agent trained only on the environment reward is included as a reference. The observed patterns match each trait's intended behavior: socializers perform more plate and onion pickups while making few deliveries, reflecting an assistive role; achievers lead in onion pickups, onion-in-pot placements, and deliveries, indicating responsibility for most of the cooking pipeline; and explorers place the most items on counters, reflecting a preference for less conventional actions. All traits differ from the self-play baseline, demonstrating that trait-based regularization induces semantically meaningful behavioral diversity. Fig. 4 further shows that increasing α_i produces behavioral changes within each trait. Higher socializer levels shift behavior from plate/onion pickups to almost exclusively onion handling, while higher achiever levels increasingly take over onion-related tasks. This shows that TEAM controls behavior both across traits and continuously within each trait. The explorer heatmap is provided in Appendix F.5. All heatmaps use column-wise normalization.
Non-overcooked environments. Appendix F.4 provides behavior analysis on the SMAC and coin game, showing attacks/coin collection differ between traits based on the trait characteristics.

Q_2: Does training the ego with trait-consistent partners improve ZSC with humans and AI? Overcooked V1 results. Figure 3(a) reports cross-play and cross-seed performance averaged over the five Overcooked layouts. TEAM achieves the highest reward (171.48), followed by HSP (152.32), MEP (131.12), and FCP (110.66). HSP performs particularly well on Coordination Ring and Counter Circuit, where its event-based rewards favor onion-oriented behaviors that align with the counter-passing structure of these layouts. Per-layout results are provided in Appendix F.7. For the human study, we recruited 32 participants (ages 21–60; 69% male, 31% female) for approximately 30-minute sessions. The study complied with institutional ethical guidelines: participants provided informed consent and received compensation of at least the local minimum wage. Fig. 5 shows that the human evaluation mirrors the AI–AI results. TEAM achieves the highest mean reward on three of the five layouts and matches HSP on Coordination Ring and Counter Circuit, while showing the largest gains on Asymmetric Advantage and Forced Coord, demonstrating human-AI effectiveness.
Other benchmarks. Table 1 summarizes the results on Coin Game, Overcooked V2 and SMAC. TEAM consistently achieves the highest performance across all five settings, with the largest relative margins on Coin Game and both SMAC maps, where the next-best baseline (HSP or FCP) trails by roughly 15–20% points. In SMAC we used the LR variant of the achiever. The HSP biases we used are in Appendix D. Note that CEC requires modifying the environment structure for terrain diversity; this is not possible in non-Overcooked settings. For this reason, we focus on the other baselines.

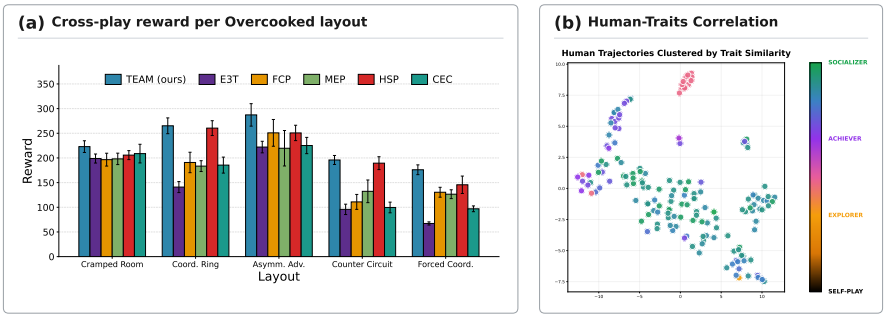


Figure 5: **(a)** Human-experiments ($\pm$ SEM) on 5 Overcooked layouts.**(b)** Correlation map – each dot is a human trajectory; colors indicates corelation level to our traits and self-play.

7.1 Similarity to Human Players, Ablation, and Analysis

Similarity to human players. Table 2 reports the L_1 distance between all populations and human trajectories on five Overcooked layouts, with TEAM achieving the smallest distance and K/N_ϵ values. In Fig. 5b, we show correlations of human trajectories to our proposed traits and self-play-based populations, indicating much higher correlation with the traits than with self-play (construction of Fig. 5b, Table 2 in Appendix D.3).

Table 2: L_1 distance from human trajectories by population.

Population	L_1 distance	$\frac{K}{N_\epsilon}$ $(\epsilon = 0.54)$
TEAM	0.49	90%
HSP	0.55	36%
FCP	0.56	9%
MEP	0.59	5%

Self-play and per-trait ablations. In Table 4 (Appendix F.1) we provide an ablation for the self-play round in phase 2. The Table shows a modest impact on cross-play, where most performance improvement stems from the cross-seed evaluation. In Appendix F.2 we include per-trait ablation where the ego is trained each time with only one-trait population (e.g., achievers), indicating the improvement lies in the combination of traits and not in a single one.

Cross seed vs cross-play analysis. Table 6 (Appendix F.3) shows separate results for cross-seed and cross-play across all settings. Team outperforms baselines separately on each category.

8 Discussion

Comparison with HSP. HSP exhibits competitive performance but relies on hand-crafted task events that can be difficult or infeasible to design. For example, replicating TEAM's assistive socializer requires rewarding onion pickups and identifying which actions constitute "assistance." While such events are available in Overcooked, comparable signals are hard to obtain in other domains.

Killer trait. We omit the killer, since its more disruptive form could produce adversarial policies and prevent task completion. In its milder form, it overlaps with the negative α_i values of the achiever that we use, which incentivize the achiever to complete the task at the expense of its partner.

Limitations/future work. Humans may exhibit mixed traits, yet covering the full space is combinatorially expensive. We use a small combinations set, leaving scalable coverage to future work.

9 Conclusion

This paper introduces a novel methodology for ZSC in which population diversity is modeled as a set of human-aligned behavioral identities. Building on this, we present TEAM, a task-agnostic framework for inducing socializers, achievers, and explorers through lightweight modifications to the learning objective, yielding structured behavioral diversity without task-specific reward engineering. By reformulating ZSC, we tap into MDP generalization theory, showing that broad population coverage can hinder generalization if not properly structured, motivating focused diversity around human-relevant behaviors. Experiments across Overcooked, Overcooked V2, SMAC, and Coin Game demonstrate distinct, controllable behaviors and gains over existing baselines.

REFERENCES

Richard Bartle. Hearts, clubs, diamonds, spades: Players who suit muds. *Journal of MUD research*, 1(1):19, 1996.

Xiaoyu Chen, Jiachen Hu, Chi Jin, Lihong Li, and Liwei Wang. Understanding domain randomization for sim-to-real transfer. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=T8vZHIRTrY.

Lauren S Ferro, Steffen P Walz, and Stefan Greuter. Towards personalised, gamified systems: an investigation into game design, personality and player typologies. In *Proceedings of the 9th Australasian conference on interactive entertainment: Matters of life and death*, pp. 1–6, 2013.

Tobias Gessler, Tin Dizdarevic, Ani Calinescu, Benjamin Ellis, Andrei Lupu, and Jakob Nicolaus Foerster. Overcookedv2: Rethinking overcooked for zero-shot coordination. *arXiv preprint arXiv:2503.17821*, 2025.

Juho Hamari and Vili Lehdonvirta. Game design as marketing: How game mechanics create demand for virtual goods. *International Journal of Business Science & Applied Management (IJBSAM)*, 5 (1):14–29, 2010.

Juho Hamari and Janne Tuunanen. Player types: A meta-synthesis. *Transactions of the Digital Games Research Association*, 1(2), 2014.

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.

Barry Ip and Gabriel Jacobs. Segmentation of the games market using multivariate analysis. *Journal of Targeting, Measurement and Analysis for Marketing*, 13(3):275–287, 2005.

Kunal Jha, Wilka Carvalho, Yancheng Liang, Simon Shaolei Du, Max Kleiman-Weiner, and Natasha Jaques. Cross-environment cooperation enables zero-shot multi-agent coordination. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=zBBYsVGKuB.

Joel Z Leibo, Edgar A Dueñez-Guzman, Alexander Vezhnevets, John P Agapiou, Peter Sunehag, Raphael Koster, Jayd Matyas, Charlie Beattie, Igor Mordatch, and Thore Graepel. Scalable evaluation of multi-agent reinforcement learning with melting pot. In *International conference on machine learning*, pp. 6187–6199. PMLR, 2021.

Pengyi Li, Hongyao Tang, Tianpei Yang, Xiaotian Hao, Tong Sang, Yan Zheng, Jianye Hao, Matthew E Taylor, Wenyuan Tao, Zhen Wang, et al. Pmic: Improving multi-agent reinforcement learning with progressive mutual information collaboration. *arXiv preprint arXiv:2203.08553*, 2022.

Yancheng Liang, Daphne Chen, Abhishek Gupta, Simon Shaolei Du, and Natasha Jaques. Learning to cooperate with humans using generative agents. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=v4dXL3LsGX.

Xingzhou Lou, Jiaxian Guo, Junge Zhang, Jun Wang, Kaiqi Huang, and Yali Du. Pecan: Leveraging policy ensemble for context-aware zero-shot human-ai coordination. In *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '23, pp. 679–688, Richland, SC, 2023. International Foundation for Autonomous Agents and Multiagent Systems. ISBN 9781450394321.

Andrew B Moynihan, Ana Guinote, and Eric R Igou. Relational dynamics and meaning in life: Dominance predicts perceived social support, belongingness, and meaning in life. *Personality and Individual Differences*, 211:112249, 2023.

Vivek Myers, Evan Ellis, Sergey Levine, Benjamin Eysenbach, and Anca Dragan. Learning to assist humans without inferring rewards. *Advances in Neural Information Processing Systems*, 37: 71540–71567, 2024.

Lennart E Nacke, Chris Bateman, and Regan L Mandryk. Brainhex: preliminary results from a neurobiological gamer typology survey. In *International conference on entertainment computing*, pp. 288–293. Springer, 2011.

Roberta Raileanu, Emily Denton, Arthur Szlam, and Rob Fergus. Modeling others using oneself in multi-agent reinforcement learning. In *International conference on machine learning*, pp. 4257–4266. PMLR, 2018.

Alexander Rutherford, Benjamin Ellis, Matteo Gallici, Jonathan Cook, Andrei Lupu, Garar Ingvarsson, Timon Willi, Ravi Hammond, Akbir Khan, Christian Schroeder de Witt, Alexandra Souly, Saptarashmi Bandyopadhyay, Mikayel Samvelyan, Minqi Jiang, Robert Tjarko Lange, Shimon Whiteson, Bruno Lacerda, Nick Hawes, Tim Rocktäschel, Chris Lu, and Jakob Nicolaus Foerster. Jaxmarl: Multi-agent rl environments and algorithms in jax. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024.

Bidipta Sarkar, Andy Shih, and Dorsa Sadigh. Diverse conventions for human-ai collaboration. *Advances in neural information processing systems*, 36:23115–23139, 2023.

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.

Caroline Wang, Arrasy Rahman, Jiaxun Cui, Yoonchang Sung, and Peter Stone. Rotate: Regret-driven open-ended training for ad hoc teamwork. *arXiv preprint arXiv:2505.23686*, 2025.

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.

Leo Sang-Min Whang and Geunyoung Chang. Lifestyles of virtual world residents: Living in the on-line game" lineage". *CyberPsychology & behavior*, 7(5):592–600, 2004.

Xue Yan, Jiaxian Guo, Xingzhou Lou, Jun Wang, Haifeng Zhang, and Yali Du. An efficient end-to-end training approach for zero-shot human-ai coordination. *Advances in neural information processing systems*, 36:2636–2658, 2023.

Nick Yee. The demographics, motivations, and derived experiences of users of massively multi-user online graphical environments. *Presence: teleoperators and virtual environments*, 15(3):309–329, 2006.

Chao Yu, Jiaxuan Gao, Weilin Liu, Botian Xu, Hao Tang, Jiaqi Yang, Yu Wang, and Yi Wu. Learning zero-shot cooperation with humans, assuming humans are biased. *arXiv preprint arXiv:2302.01605*, 2023.

Peter Zackariasson, Nils Wåhlin, and Timothy L Wilson. Virtual identities and market segmentation in marketing in and through massively multiplayer online games (mmogs). *Services Marketing Quarterly*, 31(3):275–295, 2010.

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, pp. 6145–6153, 2023.

A Large Language Model (LLM) Usage Statement

We used LLMs as a writing assistant during the preparation of this manuscript. Specifically, the LLM was used to help polish grammar, improve sentence clarity, and suggest phrasing during the drafting and editing process.

B Broader Impact

Our work targets cooperative human–AI coordination, which has potential positive impact in assistive and collaborative settings; we do not foresee specific negative societal impacts beyond those broadly associated with multi-agent reinforcement learning.

C Behavioral Taxonomy

Our work draws on the foundational framework for modeling human player behavior, introduced by Bartle (Bartle, 1996). It was originally derived from observing the Multi-User Dungeons (MUDs), early text-based online role-playing environments. Bartle categorized player motivations and behaviors based on their style of engagement and primary in-game focus. The Bartle's axes assign players along two axes: action vs interaction, and player vs. world-orientation, yielding four archetypes: achievers, explorers, socializers, and killers. These types capture behavior in persistent, open-ended multiplayer worlds, where players freely choose between progressing in the game, e.g., fighting monsters (achiever), exploring the game's mechanics and terrain (explorer), social interaction that can be expressed in several ways, e.g., chat and guild forming (socializer), or acting on other players by dominating and disrupting them (killers).

Building on Bartle's original framework, a substantial body of work has extended and empirically examined player typologies across different game settings (Zackariasson et al., 2010; Ferro et al., 2013; Hamari & Lehdonvirta, 2010; Ip & Jacobs, 2005). Synthesizing these studies, Hamari and Tuunanen (Hamari & Tuunanen, 2014) conducted a meta-review and proposed a unified taxonomy consisting of seven player types: *achievement, exploration, sociability, domination, immersion*, along with *gaming intensity* and *in-game demographics*. This taxonomy largely preserves Bartle's original categories, retaining *achievement*, *exploration*, and *sociability*, while generalizing the "Killer" archetype into the broader concept of *domination*. The empirical works surveyed span a diverse set of game genres, including massively multiplayer online games (e.g., *World of Warcraft*), first-person shooters (e.g., *Counter-Strike*), real-time strategy games (e.g., *StarCraft*), and casual or social games (e.g., *FarmVille*). Crucially, while these games differ significantly in structure, they share a common property: player behavior is expressed through interaction with both the environment and other players under some form of objective or incentive structure. For example, in *World of Warcraft*, sociability appears as long-term collaboration in guilds, whereas in *Counter-Strike*, it manifests as short-term team coordination. Similarly, achievement may correspond to completing quests in *World of Warcraft* or winning matches in *StarCraft*, and exploration may involve discovering new areas in open-world settings or experimenting with strategies in competitive games

Overcooked differs from the games surveyed in the meta-analysis in being short-horizon, and lacking combat or adversarial elements, so "Killer" behaviors do not appear, though aspects of domination and influence over partners can still emerge. Despite these differences, it retains the core structure of multi-agent interaction under shared or individual objectives, with success being highly interdependent and requiring continuous coordination between players.

C.1 Definition of Player Types

(Hamari & Tuunanen, 2014) described a taxonomy of seven categories of player types that have been identified in prior studies on collaborative multi-player games:

- **Achiever:** Goal-oriented individuals motivated by long-term success, task completion, and the accumulation of in-game wealth or status (Nacke et al., 2011). Such players may exhibit additional characteristics such as an interest in analyzing the underlying rules in order to optimize performance.

- **Explorer:** Curiosity-driven players who enjoy discovery, mapping the topology of the environment, and experimenting with the underlying "physics" or rules of the game world (Bartle, 1996; Hamari & Tuunanen, 2014). Such players prefer interaction with both the world and other players.
- **Socializer:** Social players are motivated by the relational connections and interactions with other players (Bartle, 1996). Such interaction can be divided into several sub-categories (Zackariasson et al., 2010): pure socializing (e.g., chats), relational (e.g., help and support), and teamwork that leads to group achievements. Our focus is on relational connections that directly affect teamwork.
- **Domination/killer.** Domination generalizes the killer trait that originated from Bartle's axes into a larger group of human behaviors. Such individuals can be characterized by the tendency to assert oneself or impose one's will by exhibiting assertive and fearless behavior (Moynihan et al., 2023). In the context of gameplay, it is often associated with aggressive gamers (Hamari & Tuunanen, 2014) and players who seek visibility in the game, either by assertiveness or aggressive interaction (Ferro et al., 2013). When such interaction includes an agent that attempts to cause stress to its teammate and establish superiority, in a manner that may hurt others, the actor can be regarded as a killer (Bartle, 1996).
- **Immersion:** Players who prioritize role-playing, becoming part of a fantasy story, and escaping the realities of the physical world (Hamari & Tuunanen, 2014). This involves adopting game-specific morals and identities that may differ significantly from the player's real-world persona (Whang & Chang, 2004).
- **Gaming Intensity and Skills:** A behavioral continuum that categorizes players by their willingness to invest time and effort into the game. It segments players between "Hardcore" (highly dedicated, competitive, and time-intensive) and "Casual" (shorter sessions, focused on relaxation and killing time) (Hamari & Tuunanen, 2014; Ip & Jacobs, 2005).
- **In-game Demographics:** Segmentation based on the functional role associated with a player's in-game identity, such as their chosen character class (e.g., warrior vs. priest) or profession (e.g., blacksmith vs. tailor) (Hamari & Tuunanen, 2014; Hamari & Lehdonvirta, 2010). It can also be used to identify players based on their level of proficiency.

Traits currently used in the human-AI literature. Recent works on human-AI (Wang et al., 2024; Strouse et al., 2021) are currently employing the proficiency trait by setting checkpoints during training, which yields a lower level of proficiency compared to fully trained agents. In the human behavior literature, it is addressed via the notion of in-game demographics (Hamari & Tuunanen, 2014), which categorizes players as amateurs, semi-professionals, etc.

D Hyperparameter, Architecture and Experiments Specifications

This section specifies all architecture and configuration details used in the main experiments. We divide it into three subsections. Sec. D.1 details the TEAM parameters, while Sec. D.2 details the baselines parameters. In Sec. D.3 we explain the construction of Fig 5 (human proximity).

Compute resources. All experiments were conducted on a shared internal SLURM cluster node equipped with 8× NVIDIA A100 GPUs, 256 CPU cores (2 sockets, 64 cores per socket, x86 64 architecture). Each training run used a single A100 GPU.

D.1 Team Specifications

During Phase 1, the population members were trained according to the parameters detailed in Table 2. Note that, similarly to previous works (Yu et al., 2023), the Phase 1 partners are constructed only from fully connected layers, and no memory component (e.g., GRU) is used, with init/mid/final checkpoints to obtain more training partners. The ego training regime (phase 2) is detailed in Table 3. Note that our structure is derived from all previous works that use an RNN-based ego agent.

α_i **selection.** For all traits, we sample α_i on fixed intervals. For achiever, we use $\alpha_i \in \{-0.25 + 0.15k \mid k = 0, \ldots, 6\}$.. For socializers, we use $\alpha_i \in \{0 + 0.15k \mid k = 0, \ldots, 6\}$., for explorers, we use $\alpha_i \in \{0.25 + 0.1k \mid k = 0, \ldots, 6\}$. For the mixed traits, we use agents that are a mix of socializers and explorers (0.3,0.25),(0.45,0.25), and explorers with achievers (0.25,0.15), (0.25,0.3).

Table 2: Population agent: architecture and key hyperparameters.

Component / Parameter	**Specification**
Architecture	
Algorithm	IPPO (independent PPO, two agents)
Policy class	Categorical (discrete actions)
Hidden layers	2 fully-connected layers, 64 units each
Activation	$\tanh$
Actor / Critic heads	Separate; linear output (logits / scalar value)
Weight init	Orthogonal ($\sqrt{2}$ hidden, 0.01 actor, 1.0 critic)
Optimizer	Adam ($\varepsilon = 10^{-5}$) with global-norm clipping
LR schedule	Linear anneal to 0
Hyperparameters	
Learning rate	3×10^{-5}
Total environment steps	3.5×10^{7}
Parallel environments / rollout length	100 / 128
PPO epochs / minibatches	40 / 4
Discount γ / GAE-λ	0.99 / 0.95
PPO clip / entropy / VF coef.	0.2 / 0.01 / 0.5
Intrinsic reward weight	0.2

Table 3: Ego agent: architecture and key hyperparameters.

Component / Parameter	**Specification**
Architecture	
Algorithm	PPO with recurrent policy (self-play + population)
Observation encoder	2 Conv layers (2×2 kernels, 64 then 32 channels, ReLU)
Post-conv MLP	2 dense layers ($2h$, h units), $\tanh$
Recurrent core	LSTM (`OptimizedLSTMCell`), hidden size h=120
Actor head	3 dense layers (h, $\frac{3h}{4}$, $\frac{h}{2}$), $\tanh$
Critic head	4 dense layers ($2h$, h, $\frac{3h}{4}$, $\frac{h}{2}$) $\rightarrow$ scalar
Weight init	Orthogonal ($\sqrt{2}$ hidden, 0.01 actor, 1.0 critic)
Optimizer	Adam ($\varepsilon = 10^{-5}$) with global-norm clipping
LR schedule	Linear anneal to 0
Training scheme	Alternating self-play and partner rounds (population of $3\times$pop$+3$)
Hyperparameters	
Learning rate	3×10^{-4}
Total environment steps	1.45×10^{8}
Parallel environments / rollout length	100 / 128
PPO epochs / minibatches	4 / 4
Discount γ / GAE-λ	0.99 / 0.95
PPO clip / entropy / VF coef.	0.2 / 0.01 / 0.5
Updates per partner / population size	15 / 22

D.2 Baselines Specifications

FCP. During phase 1, we train FCP with the same population size and 3 checkpoint as TEAM. All the population parameters are identical to the ones TEAM uses in Table 2. In phase 2, the ego is also identical to TEAM, with the same parameters as in Table 3.

MEP. During phase 1, we train MEP with the same population size as TEAM with the same checkpoint regime. All the population parameters are identical to the ones TEAM uses in Table 2. For the coefficient of population entropy, we use similar values as the original paper, $\alpha = 0.01$ in

general, except for Forced Coordination in Overcooked V1, which uses $\alpha = 0.04$. In phase 2, the ego is also identical to TEAM, with the same parameters and compute budget as in Table 3.

E3T and CEC. Both CEC and E3T train via self-play. We use the implementation provided for JAX (Jha et al., 2025). Note that the RNN architecture is similar to Table 3.

HSP biases and parameters. We follow the original HSP protocol, where half the phase 1 population is trained via MEP (with standard checkpoints), and the other half via behavior biases. Note that the HSP algorithm trains a population double the size (the non-MEP part) and then filter half that are most diverse. The RNN is identical to what is described for TEAM in Table 3. For the biases in Overcooked V1/V2, we use the same biases described in the original paper (the second, non-onion ingredient gets bias similar to tomatoes). For the coin game, we use the following biases: picking one's own coin, picking the partner's coin, and receiving a reward when the partner picks one's own coin. We ablated HSP by removing the "the partner picks one's own coin" bias but didn't observe performance improvement. In SMAC, we used 5 different biases, assigning different rewards in the range [-5,5] for the first three events, and a binary reward for (4) and (5):

1. Attack action.
2. Damage dealt/how much the enemy is damaged- this bias assigns varying values for hitting the enemy, which results in either damage-seeking vs avoiding engagements.
3. Damage taken- this bias assigns varying values for the event of "taking damage" and affects how much damage the agent is willing to absorb, yielding either a risk-averse or risk-seeking strategy.
4. Move to enemy - reward how close or not the ally agent is to the enemy. In this bias, we reward the agent if its distance to the enemy shrinks.
5. Task reward - a binary selection of using the task reward or not.

D.3 Human Proximity Measurements

Construction of corelation map (Fig. 5b). We analyze 2019 human-human Overcooked gameplay episodes (172 games meeting a minimum-length threshold of 380 steps, 165 after excluding 7 episodes with all-zero behavior counts). For each episode, we replay the recorded sequence of game states and joint actions through the original Overcooked engine to recover per-step event flags, which we aggregate into six behavioral categories per player per game: plate pickups, onions picked up, dishes (soup) picked up, deliveries, items placed on counters, and onions placed in pots. Each episode's resulting six-dimensional count vector is standardized and embedded into two dimensions using t-SNE — a nonlinear dimensionality-reduction technique that positions points so that episodes with similar behavioral profiles in the original six-dimensional space end up near each other in the 2D layout, making clusters of similar play styles visually apparent. Position in this 2D space thus reflects behavioral similarity to other human episodes, not similarity to any predefined archetype. Independently of position, we color each point by computing the Spearman rank correlation between its six-category vector and each of four reference behavior profiles from Fig. 3(b) of the paper (Socializer, Achiever, Explorer, and Self-play): the four correlations are combined into a single ordinal score reflecting which reference profile the episode's behavior most closely resembles, and this score determines the point's color on a continuous gradient. This produces a visualization where spatial clustering reveals natural groupings in raw human behavior, while color reveals how each cluster relates to the paper's predefined behavioral archetypes.

Construction of Table 2. To evaluate how closely a pretrained population partner agent matches real human behavior, we measure, for each state in a recorded human trajectory, the probability the agent's frozen policy assigns to the human's actual action, and define the distance as one minus this probability. Averaging over every state in every human episode for a given layout, and across all agents in the population, yields an overall distance score, where zero indicates perfect prediction.

E Algorithm

We summarize TEAM in Algorithms 1 and 2, corresponding to the two training phases described in Section 5. In Phase 1, we train a population of trait-induced partners by branching on the agent's

Algorithm 1: TEAM Phase 1: Trait-Induced Population Training

Input: Traits $\mathcal{T} = \{\text{SOC}, \text{ACH}, \text{EXP}, \text{MIX}\}$, agents per trait M, coefficients $\{\alpha_i\}$
Output: Population $\mathcal{P}$ of trait-induced partners
$\mathcal{P} \leftarrow \emptyset$;
foreach *trait* $\tau \in \mathcal{T}$*, agent* $i = 1, \ldots, M$ **do**
  Initialize agent π_i and partner π_{-i};
  if $\tau = \text{SOC}$ **then**
    Use pretrained partner; train with reward in Eq. 1;
  else if $\tau = \text{ACH}$ **then**
    Train with reward in Eq. 2 (or reduced-LR partner);
  else if $\tau = \text{EXP}$ **then**
    Train with PPO loss in Eq. 4 using high entropy coefficient α_i;
  else // $\tau = \texttt{MIX}$
    Choose a pairing $\text{EXP} + \text{SOC}$ or $\text{EXP} + \text{ACH}$;
    Train π_i with the PPO loss in Eq. 4 (high entropy coefficient α_i), combined with the reward in Eq. 1 or Eq. 2 according to the chosen pairing;
  Train π_i with PPO and save checkpoints at {init, mid, final};
  $\mathcal{P} \leftarrow \mathcal{P} \cup \{\text{checkpoints of } \pi_i\}$;
return $\mathcal{P}$;

Algorithm 2: TEAM Phase 2: Ego Agent Training

Input: Frozen population $\mathcal{P}$ from Algorithm 1, number of rounds N
Output: Trained ego agent π_e
Initialize recurrent ego agent π_e;
for *round* $= 1, \ldots, N$ **do**
  Update π_e via PPO under self-play;
  foreach *partner* $\pi \in \mathcal{P}$ **do**
    Update π_e via PPO while π remains frozen;
return π_e;

designated trait: socializers are trained with the empowerment-augmented reward (Eq. 1) against a fixed pretrained partner, achievers are trained with the workload-adjusted reward (Eq. 2) jointly with their partner, and explorers are trained with a high entropy coefficient in the PPO loss (Eq. 4). For each agent, we save three checkpoints (init, mid, final) following standard practice (Strouse et al., 2021; Zhao et al., 2023), yielding a diverse population $\mathcal{P}$. In Phase 2, the ego agent is trained against this frozen population via PPO, alternating between self-play and rounds in which it is paired with each partner in $\mathcal{P}$.

F TRAIT ANALYSIS, ABLATIONS, AND RESULTS SUMMERY

This section provides the following ablations and results:

- **(ablation) Self-play ablation.** Sec. F.1 ablates the Phase 2 self-play round.
- **(ablation) Per-trait ablation.** Sec. F.2 evaluates each trait separately by training the ego on a single-trait population.
- **(results summery) Cross-seed vs. cross-play analysis.** Sec. F.3 reports separate cross-seed and cross-play results across all settings.
- **(results summary) Tabular summary and per-layout results of main paper overcooked experiments.** Sec. F.7 presents a tabular summary of the main results from Fig. 3 along with per-layout separation.
- **(behavior analysis) Behavior analysis on non-overcooked maps.** Sec. F.4 analyzes traits on coin game and SMAC.
- **(behavior analysis) General Trait analysis on forced cord and asymmetric advantage, and explorer on coord ring.** Sec. F.5 presents complementary analysis to the heat maps in the main paper for explorer on the coord ring map, along with all traits on forced cord and asymmetric advantage.
- **(behavior analysis) Achievers variants comparison.** Sec. F.6 compares behavior analysis of the LR variance of the reward allocation variant.

F.1 SELF-PLAY ABLATION

As stated in the main text, in our scheme we use a self-play round during phase 2 of the ego training. We present ablation results in Table 4, measuring performance without self-play. The results are separated into cross-seed, cross-play, and general results. The rightmost column shows the original results with self-play for ease of comparison.

Table 4: TEAM self-play ablation separated into cross-seed, cross-play, and combined results (mean $\pm$ SEM).

	TEAM cross-seed (no self-play)	TEAM cross-play (no self-play)	TEAM general score (no self-play)	TEAM original score (with self-play)
SMAC (2s vs 3z)	0.98 ± 0.03	0.5 ± 0.15	0.604 ± 0.14	0.59 ± 0.11
SMAC (2m vs 2z)	0.9 ± 0.09	0.49 ± 0.15	0.57 ± 0.15	0.629 ± 0.11
Coin game	49.2 ± 7.9	32.25 ± 7.7	35.7 ± 3.25	30.7 ± 2.2
Overcooked v1 (coord_ring)	214 ± 10.5	193 ± 6	194 ± 11.2	197.9 ± 9.9
Overcooked v1 (counter)	143 ± 10	120 ± 12	122 ± 7.1	126.1 ± 7.8
Overcooked v1 (cramped room)	202 ± 8.9	189 ± 10.9	191 ± 10.9	195.1 ± 7.1
Overcooked v1 (forced coord)	46.2 ± 8	64 ± 7	49.1 ± 6.9	55.1 ± 6.9
Overcooked v1 (asym advantage)	320 ± 28	245 ± 29	259 ± 25	283.2 ± 24.1
Overcooked v2 (demo wide)	90.9 ± 27.5	110.6 ± 27.3	106.3 ± 4.4	108.7 ± 3.81
Overcooked v2 (AA with recipe center)	260.1 ± 32.5	224.1 ± 35.6	231.1 ± 6.1	237.1 ± 6.6

F.2 PER-TRAIT ABLATION

Table 5 presents per-trait ablation, where we train the ego with the same size of population, but only with explorer/socializer/achiever. Ablation is conducted on the five original Overcooked V1 layouts.

F.3 CROSS SEED VS CROSS-PLAY ANALYSIS

Table 6 presents the cross-seed and cross-play results, separated, for TEAM and the second-best performer, in each setting.

F.4 BEHAVIOR ANALYSIS ON NON-OVERCOOKED MAPS

We recorded population behavior compared to the self-play agent (similarly to Fig. 3b): in the coin game, we categorized agent types by how they treat the partner resources: the socializer agent

Table 5: Ablation of TEAM's trait components on Overcooked v1 layouts (mean ± SEM). "Original" is the full model.

	Only achiever	Only explorer	Only socializer	Original
Cramped room	184 ± 8.9	180 ± 9.8	176 ± 5.8	195 ± 7.1
Asymm advantage	244.5 ± 29.1	211.5 ± 20.3	221.5 ± 26.4	283.2 ± 24.1
Forced coord	41 ± 7.6	53.3 ± 5.2	39.3 ± 8	55.1 ± 6.9
Coord ring	183.5 ± 7.7	179.6 ± 10.3	180.5 ± 8.4	197.9 ± 9.9
Counter circuit	117 ± 6.6	113.4 ± 7.2	110.1 ± 8	126.1 ± 7.8

Table 6: TEAM compared with the second-best performing baseline in each environment (mean ± SEM). The baseline is given in parentheses.

	TEAM cross-seed	TEAM cross-play	Second-performer cross-seed	Second-performer cross-play
Coin game	41.3 ± 6.8	27.8 ± 10.4	37.7 ± 8.6 (HSP)	23.1 ± 11.1 (HSP)
SMAC (2s vs 3z)	0.97 ± 0.03	0.49 ± 0.11	0.83 ± 0.09 (HSP)	0.41 ± 0.11 (HSP)
SMAC (2m vs 2z)	0.939 ± 0.04	0.54 ± 0.11	0.93 ± 0.06 (FCP)	0.38 ± 0.11 (FCP)
Demo wide (Overcooked v2)	98.9 ± 3.5	111.5 ± 3.8	105.2 ± 2.3 (MEP)	102.2 ± 2.1 (MEP)
Asymm_advantages_recipes_center (Overcooked v2)	276.3 ± 8.5	$226.7 \pm 9, 2$	262.5 ± 10.5 (MEP)	210.8 ± 9.3 (MEP)
Coord ring (Overcooked v1)	215.2 ± 9.3	193.1 ± 8.7	208.9 ± 6.3 (HSP)	186.1 ± 5.7 (HSP)
Counter (Overcooked v1)	144.2 ± 6.9	121.2 ± 7.0	131.5 ± 9.4 (HSP)	114.2 ± 9.7 (HSP)
Cramped room (Overcooked v1)	208.2 ± 6.9	192 ± 7.3	207.9 ± 5.4 (HSP)	185.8 ± 4.9 (HSP)
Forced coord (Overcooked v1)	75.3 ± 6.1	49.6 ± 6.7	65.1 ± 3.7 (MEP)	41.9 ± 6.9 (MEP)
Asymm advantage (Overcooked v1)	400.3 ± 20.5	250.2 ± 24.1	264.2 ± 15.1 (HSP)	206.5 ± 15.2 (HSP)

collected partner coins 30% less than self-play, while the achiever had 40% more partner coin collection compared to self-play, yielding strategies that differ by how the games are solved, via greedy coin harvest or forming collaboration. Explorers picked partner coin 15% more than self-play, likely due to higher stochasticity. In SMAC (2m vs 2z map), we categorize agents based on the amount of fire action and distance between the ally units: the socializer fires 50% less than self-play throughout the episode, but on average remains closer to the partner than all others, indicating tight collaboration. Achievers used the fire action 25% more than self-play, with the second-highest distance between the allies of all types, indicating higher independence. Explorers exhibit the largest distance as they conducted higher terrain exploration.

F.5 All Traits on Asymmetric Maps and Explorers on the Coordination Ring.

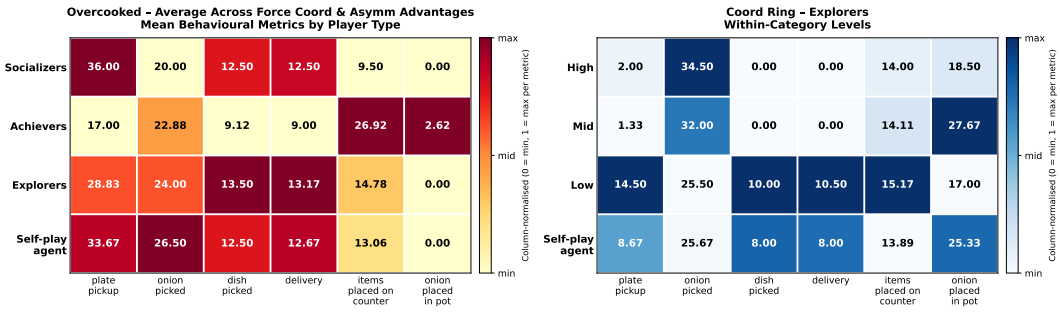


Figure 6: Studying intra-trait diversity within each trait on the coordination ring for achievers and socializers.

Fig. 6 complements the main behavioral analysis along two axes. The left panel reports cross-trait behavior averaged over Force Coord and Asymm Advantages, two layouts structurally different from those dominating Fig. 3(b). The same qualitative pattern persists: socializers lead on plate pickups while remaining low on onion-in-pot, consistent with their assistive role; achievers stand out by being the only trait that, when playing as the left player in Asymm Advantages, chooses to pick onions rather than approach the closer plates, reflecting their tendency to assume a larger share of the workload; and explorers occupy an intermediate profile. The right panel completes the intra-trait

analysis from Fig. 4 by showing how explorer behavior varies with the entropy temperature on Coord Ring: as temperature increases from Low to High, the agent shifts from a self-play-like profile touching every step of the pipeline to an unconventional regime concentrated almost exclusively on onion handling and pot placement, mirroring the controlled within-trait spectra observed for socializers and achievers.

F.6 ACHIEVERS VARIANTS COMPARISON

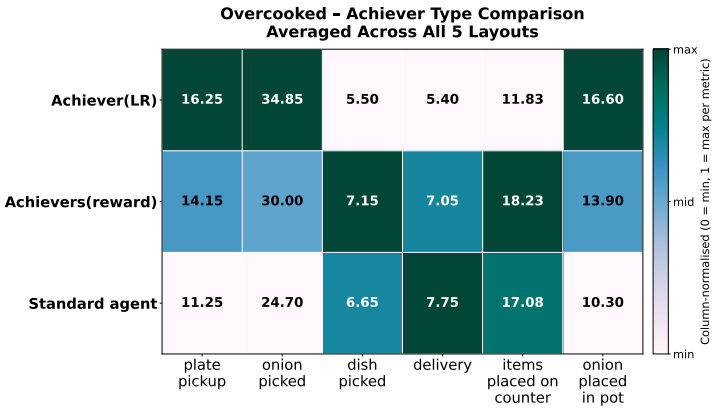


Figure 7: Comparison of achiever variants

Fig. 7 compares the two achiever variants against a standard self-play agent, averaged across all five Overcooked layouts. Both variants successfully induce achiever behavior, taking on a larger share of the workload relative to the baseline. However, Achiever-reward yields a better-spread behavioral profile, increasing engagement across the full pipeline (plate pickups, onion picks, dish picks, deliveries, and onion-in-pot placements), whereas Achiever-LR concentrates the shift more narrowly on plate pickups and onion handling. This indicates that explicit reward reallocation produces a richer and more balanced achiever behavior than learning-rate manipulation.

F.7 TABULAR SUMMARY OF MAIN PAPER RESULTS WITH PER-LAYOUT GRAPHS

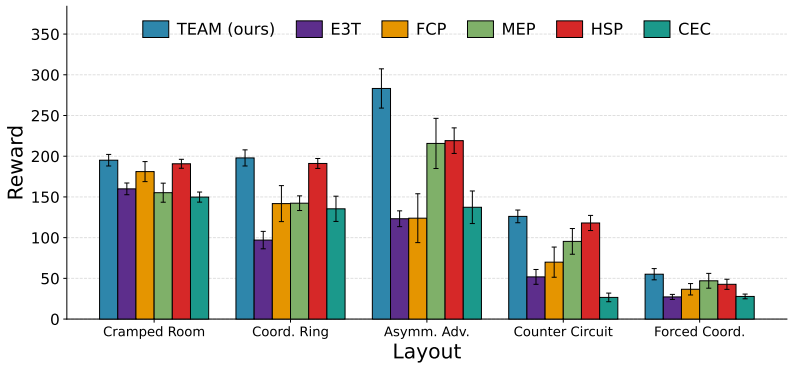


Figure 8: Per layout results on the 5 Overcooked maps in cross play and cross seeds.

Table 7: Cross-play (XP) reward per Overcooked layout, mean ± SEM across 20 training seeds. The best result in each row is in **bold**.

Layout	TEAM (ours)	E3T	FCP	MEP	HSP	CEC
Cramped Room	$\mathbf{195.1 \pm 7.1}$	159.9 ± 7.2	181.1 ± 12.3	155.2 ± 11.7	190.7 ± 5.5	149.8 ± 6.2
Coord. Ring	$\mathbf{197.9 \pm 9.9}$	97.0 ± 10.7	141.8 ± 22.1	142.3 ± 9.0	191.1 ± 6.1	135.4 ± 15.5
Asymm. Adv.	$\mathbf{283.2 \pm 24.1}$	123.2 ± 9.7	123.9 ± 30.0	215.7 ± 30.8	219.1 ± 15.7	137.3 ± 20.0
Counter Circuit	$\mathbf{126.1 \pm 7.8}$	51.8 ± 9.1	69.9 ± 18.54	95.4 ± 15.8	118.0 ± 9.3	26.6 ± 5.3
Forced Coord.	$\mathbf{55.1 \pm 6.9}$	27.2 ± 3.1	36.6 ± 7.0	47.0 ± 9.1	42.7 ± 6.2	27.8 ± 2.9
Average	**171.48**	91.82	110.66	131.12	152.32	95.8

We evaluate each method by averaging cross-play (XP) reward across all unordered pairs of training seeds (each method trained with 20 seeds, giving $\binom{20}{2} = 190$ cross-pairings per layout), for cross-play, 100*20 = 2000 pairings for each method. Table 7 and Fig 8 report the mean reward per layout for each method, with standard errors ($\mathrm{SEM} = \sigma/\sqrt{20}$) over the 20 seeds. Our method, TEAM, achieves the highest mean reward on every one of the five Overcooked layouts. The advantage is largest on the more coordination-heavy layouts: on *Asymmetric Advantages* TEAM reaches 283.2 ± 24.1, outperforming the next-best baseline (HSP, 219.1 ± 15.7) by roughly 29%, and on *Counter Circuit* it reaches 126.1 ± 7.8 versus the next-best HSP at 118.0 ± 9.7. On the simpler *Cramped Room* layout, the margin narrows (195.1 vs. HSP 190.7), as expected when the coordination problem is mild and several methods saturate. Averaged across layouts, TEAM attains 171.48 XP reward, compared to 152.32 for HSP, 131.12 for MEP, 110.66 for FCP, 95.8 for CEC, and 91.82 for E3T. Overall the per-layout breakdown shows that TEAM's gains are not driven by a single layout but are consistent across the full benchmark.

G GENERALIZATION BOUND ON EGO PERFORMANCE

G.1 PRELIMINARIES.

We follow the definition used in (Chen et al., 2022)

Defining d_e and $\mathcal{N}_H$. Fix $\theta \in \Theta$, and let $\pi^*_{M_\theta}$ be the optimal stationary policy of M_θ and $\rho^*_{M_\theta}$ its optimal average reward:

$$\rho^*_{M_\theta} = \max_{\pi: S \to A} \lim_{T \to \infty} \frac{1}{T} \mathbb{E}_{M_\theta, \pi} \left[\sum_{t=1}^{T} R^\theta(s_t, a_t) \right]. \tag{7}$$

The *bias function* of M_θ is

$$\lambda^*_{M_\theta}(s) = \lim_{T \to \infty} \mathbb{E}_{M_\theta, \pi^*_{M_\theta}} \left[\sum_{t=1}^{T} \left(R^\theta(s_t, a_t) - \rho^*_{M_\theta} \right) \middle| s_1 = s \right], \tag{8}$$

which is the total advantage of starting in s over the long-run average. under Assumption 3 it satisfies $0 \le \lambda^*_{M_\theta}(s) \le D$. We write $\Lambda = \{\lambda^*_{M_\theta} : \theta \in \Theta\}$. For $\lambda \in \Lambda$ let $P^\theta\lambda(s,a) = \sum_{s' \in S} P^\theta(s' \mid s,a)\,\lambda(s')$ denote the expected value of λ one step after taking a in s under partner θ, and set

$$\mathcal{F} \;=\; \big\{\, f_{M_\theta}(s,a,\lambda) = P^\theta\lambda(s,a) \;:\; \theta \in \Theta,\; \lambda \in \Lambda \,\big\}. \tag{9}$$

Each element of $\mathcal{F}$ is one partner's one-step prediction, so two partners are hard to distinguish exactly when their predictions agree.

We use two standard measures of $\mathcal{F}$. The *covering number* $\mathcal{N}_H(\mathcal{F}, 1/H)$ is the smallest m for which there exist $f_1, \dots, f_m \in \mathcal{F}$ with $\min_i \|f - f_i\|_\infty \le 1/H$ for every $f \in \mathcal{F}$. The *eluder dimension* $\hat{d}_e = \dim_E(\mathcal{F}, 1/H)$ is the number of state-action pairs the ego may need to visit before it can tell its partner apart from every other member of the class of training MDPs. Formally, it is the largest n for which there exist query points $x_1, \dots, x_n \in S \times A \times \Lambda$ with the following property: for each $k \in [n]$ there is a pair of partners $\theta, \theta' \in \Theta$ that agree at every earlier point,

$$\big|f_{M_\theta}(x_m) - f_{M_{\theta'}}(x_m)\big| \le 1/H \qquad \text{for all } m < k, \tag{10}$$

yet differ at x_k itself,

$$\big|f_{M_\theta}(x_k) - f_{M_{\theta'}}(x_k)\big| > 1/H. \tag{11}$$

Thus d_e is the length of the longest such sequence, and so bounds how many observations the ego may need before its partner is identified.

Defining D. To define D, we make the following assumption:

Assumption 3. *There exists $D < \infty$ such that every induced MDP $M_\theta \in \mathcal{U}$ has diameter at most D. That is, for any $\theta \in \Theta$ and any two states $s \neq s'$,*

$$\min_{\pi:\, S \to A} \mathbb{E}_{M_\theta,\,\pi}\big[\,T(s' \mid M_\theta, \pi, s)\,\big] \;\le\; D, \tag{12}$$

where $T(s' \mid M_\theta, \pi, s)$ is a random variable that denotes the first time step at which s' is reached when the ego follows the policy π from s in M_θ. This guarantees that from any state, some policy reaches any other state in at most D steps on average.

Derivation of maximum L in assumption 2. To show the existence of L, we derive its maximum possible value, though in practice L can be much smaller. Writing $\delta = d(M_{\theta_1}, M_{\theta_2})$, the two induced MDPs disagree at any given step with probability at most δ, and a disagreement at step t costs at most the remaining return $(H-t)R_{\max}$. Summing over the H steps gives $\sum_{t=1}^{H} \delta\,(H-t)\,R_{\max} \le H^2 R_{\max}\,\delta$, so the assumption holds in our setting with $L = H^2 R_{\max}$ and any ϵ_0, uniformly over starting states and hence in expectation over μ.

G.2 Deriving Theorem 1 from Domain Generalization Theory

Once reformulating ZSC as the problem of learning on a set of MDPs, we can apply Theorem 4 from (Chen et al., 2022). In their paper, they define ν as the probability distribution over the simulator class U, used to sample dynamics during domain randomization; $\nu\,(C_{M^*,\epsilon})$ is the probability, under ν, that a randomly sampled MDP lies within distance ϵ of the real MDP M^*. Note that in practice, U in our setting is the set of the trait-based agents, narrowing the group of possible partners from all policies to the trait-based. This significantly reduces the covered space, and under the premise that trait based agent produce human-relevant patterns.

To convert $\nu\,(C_{M^*,\epsilon})$ to our setting we first examine epsilon, the probability of a policy to be randomized within an epsilon gap from

$$\mathbb{E}_{\theta\sim\nu}[N_\epsilon] = \mathbb{E}_{M_{\theta_1},\dots,M_{\theta_K}\sim\nu}\left[\sum_{i=1}^{K} \mathbb{1}[M_{\theta_i} \in C_{M^*,\epsilon}]\right] = \sum_{i=1}^{K} \Pr_{M_{\theta_i}\sim\nu}[M_{\theta_i} \in C_{M^*,\epsilon}] = \sum_{i=1}^{K} \nu(C_{M^*,\epsilon}) \tag{13}$$

$$= K \cdot \nu(C_{M^*,\epsilon}).$$

This gives us:

$$\mathbb{E}_{\theta\sim\nu}[N_\epsilon] = K \cdot \nu(C_{M^*,\epsilon}). \tag{14}$$

We set $\nu(C_{M^*,\epsilon}) = \frac{\mathbb{E}_\theta[N_\epsilon]}{K}$, and apply theorem 4 (Chen et al., 2022), yielding:

$$V^*_{M^*,1}(s_1) - V^{\pi^*_e}_{M^*,1}(s_1) \;=\; \frac{K}{\mathbb{E}_\theta[N_\epsilon]} \cdot O\Big(D\sqrt{\hat{d}_e\, H\, \log\big(H \cdot \mathcal{N}_H\big)}\Big) \;+\; 2L\,\epsilon. \tag{15}$$

G.2.1 Dimensionality of sampling space

The distance from M^* is measured in terms of policy deviation:

$$d_\rho\big(M_{\theta_i}, M^*\big) \;=\; \mathbb{E}_{s\sim\rho}\Big[\,\big\|\pi_{\theta_i}(\cdot \mid s) - \pi_H(\cdot \mid s)\big\|_1\,\Big]. \tag{16}$$

$d_\rho(M_{\theta_i}, M^*) \in [0,2]$, since $\|\pi_{\theta_i}(\cdot \mid s) - \pi_H(\cdot \mid s)\|_1 = 2 \cdot \mathrm{TV}\big(\pi_{\theta_i}(\cdot \mid s), \pi_H(\cdot \mid s)\big) \in [0,2]$ for every state s. There are infinitely many policies over the interval $[0,2]$. In our setting, policies are induced by a mapping g from a d-dimensional trait space to $[0,2]$. Specifically, suppose that d traits are sampled uniformly, with $\mathbf{i} = (i_1, \ldots, i_d) \sim \mathrm{Unif}([0,1]^d)$, and each trait level ranges from 0 to 1. This trait vector induces a policy $\pi_{\theta_\mathbf{i}}$. The d traits can represent any set of behavioral or preference dimensions, such as socialization tendencies or positional preferences. The resulting distance to the human-induced MDP M^* is therefore characterized by a mapping:

$$g : [0,1]^d \to [0,2], \qquad g(\mathbf{i}) = d_\rho(M_{\theta_\mathbf{i}}, M^*).$$

Thus, the trait space contains infinitely many possible policies, with the number of combinations growing exponentially with d. To limit this dimensionality, we set $d = 3$ and focus on three behavioral traits: achievers, explorers, and socializers.