SAN: Style-Adaptive Zero-Shot Coordination via Dynamic Partner Inference

***Abstract*—Zero-shot coordination aims to develop agents that can generalize to unseen human partners without relying on human data. To achieve this, Population-Based Training (PBT) has emerged as a prevailing paradigm, which constructs a diverse partner pool via *self-play* to enhance the agent's robustness. However, existing PBT frameworks often fail in real-world interactions where human partners exhibit non-stationary strategies and dynamic style switching. To address this challenge, we propose Style Adaptive zero-shot coordinatioN(SAN), the first style-aware PBT framework designed for dynamic coordination. SAN incorporates a latent style inference module to capture the temporal evolution of partner behaviors. Specifically, we employ a Transformer-based selector to dynamically activate specialized policy heads from a diverse repertoire. To ensure high-quality training, we introduce a style-constrained utility function during the population seeding phase, explicitly incentivizing behavioral diversity via distorted quantile fractions. Extensive evaluations in *Overcooked* environment demonstrate that SAN significantly outperforms state-of-the-art methods in both static and dynamic switching scenarios, manifesting robust zero-shot generalization.**

I. Introduction

Developing artificial intelligence systems that can interact, collaborate, and assist humans is a long-standing objective in AI research [1]–[3]. Classical model-based approaches have achieved remarkable success by leveraging human expert data to construct behavioral models and planning modules [4]–[6]. However, as tasks grow more complex and privacy concerns intensify, collecting sufficient high-quality expert data becomes increasingly time-consuming and costly [7]–[10].

To address these limitations, zero-shot coordination (ZSC) has emerged as a promising paradigm for building agents that generalize to unseen human partners without using human data [11]–[17]. Typically, ZSC methods maintain a policy pool and iteratively improve an agent through *self-play* [18], [19]. Early approaches often treated pool policies uniformly and performed pairwise training to maximize cumulative reward. However, these methods are prone to implicit convention overfitting: agents may converge to equilibria tailored to their co-trainers, leading to poor generalization to unseen partners [20], [21].

Population-Based Training (PBT) [11]–[13], [22] improves robustness by maintaining a diverse partner population and training an ego agent against this pool, often using Proximal Policy Optimization [23]. State-of-the-art PBT techniques, such as fictitious co-play [15], TrajeDiv [12], and maximum entropy population [24], further incorporate explicit diversity objectives to mitigate overfitting. Despite their success, existing PBT methods typically assume that a partner's behavioral style is stationary within an episode. This assumption limits performance when interacting with real-world partners whose strategies are non-stationary and may switch styles online.

To bridge this gap, we propose **S**tyle **A**daptive zero-shot coordinatio**N** (**SAN**), the first style-aware PBT framework. SAN leverages interaction histories to infer a partner's current style and adapts the ego agent's strategy online. The SAN architecture comprises three core components: 1) a feature extractor that encodes the current state into a latent representation; 2) a multi-head action generator that produces diverse cooperative actions tailored to different styles; and 3) a state-enhanced selector with a Transformer backbone that infers the partner's style from historical sequences and activates the corresponding policy head. Furthermore, we introduce utility functions to modulate rewards during self-play, fostering a partner pool with explicitly diverse strategic styles. Our main contributions are listed as follows.

- We devise a novel scheme to enhance style diversity in the partner pool by modeling the action-value distribution via a quantile network and employing distorted quantile fractions to modify the PPO objective. This enables the generation of partners with distinct strategic temperaments, such as aggressive, neutral, and conservative.
- We propose a dynamic action adjustment strategy for the ego agent. During training, each head of the action generator specializes in coordinating with a specific partner style, while the state-enhanced selector learns to distinguish styles from state sequences, ensuring adaptive collaboration with unseen partners.
- To enhance the selector's style recognition ability, we divide states$\{s_t\}$ into two groups based on the deviation between $Q(s_t, \pi(s_t))$ and $V(s_t)$, where π denotes the partner policy, Q and V denote the state-action value function and the state value function, respectively. A larger deviation $|Q(s_t, \pi(s_t)) - V(s_t)|$ implies a bigger influence of the policy's action, indicating the significance of the state s_t in the state sequence. In the basic blocks of the selector transformer, slow/fast paths are used to deal with states in different groups, where the complex multi-layer slow path is used to extract the hidden features from more informative states while a single-layer fast path is used to deal with less significant states.

Empirical studies in the Overcooked environment demonstrate SAN's ability to dynamically infer the partner's strategy style and modify the strategy of the ego agent correspondingly, outperforming existing methods.

II. Related Work

A. Model-based Coordination

Classical approaches in human-AI collaboration are often model-based, relying on constructing behavior models from human data to facilitate planning with a human partner [4], [13], [14], [25]. These approaches typically involve planning and learning with potentially non-optimal models of human behavior, often modeled using probabilistic frameworks that assume human trajectories exist within a continuous space [6], [26]–[28]. Recent work has introduced novel deep learning frameworks that incorporate human operational insights to improve coordination among multiple robots [5], [29]. These studies explore different methods for modeling human behavior. For example, [29] compare the effectiveness of learning a human model using deep learning techniques against a more structured "theory of mind" approach. [5] evaluate the overall utility of incorporating a human model, questioning whether having such a model is beneficial at all.

B. Zero-shot Coordination

Zero-shot coordination (ZSC) [11], [24], [30]–[33] has become a significant area of research due to its unique human-data-free approach to agent collaboration, which relies entirely on self-play between agents. The first ZSC method, introduced by [11], involves independently trained agents working together towards a common objective in a cooperative game setting. A fundamental challenge of this type of method is that these agents can easily develop implicit conventions and overfit to specific partners or cooperation patterns.

Population-based training (PBT) [22] methods have emerged as effective approaches to mitigate this issue. Typically, PBT maintains a diverse population of training partners, training one ego agent with respect to the partner policy pool to collaborate effectively with a broad spectrum of partners. Fictitious co-play (FCP) [15] proposes the first two-stage framework by first creating a pool of self-play policies and then training an adaptive policy against them. [15] show that with diversity induced by different checkpoints and different random seeds, the agent can generalize well in collaborative games. Later, TrajeDiv [12] and Maximum Entropy Population method (MEP) [24] adopt explicit diversity objectives to generate diverse policies as partners and achieve state-of-the-art ZSC performance. Recently,Hidden-Utility Self-Play method (HSP) [17] explicitly models human biases as hidden reward functions in the self-play objective. By approximating the reward space as linear functions, HSP adopts an effective technique to generate an augmented policy pool with biased policies. In order to collaborate well with partners of different abilities, PECAN [34] proposes the policy ensemble method to increase the diversity of partners in the population, and then develop a context-aware method enabling the ego agent to analyze and identify the partner's ability so that it can take different actions accordingly. Note that PECAN heavily relies on domain knowledge to recognize the skill levels of partners, which may become fragile when it is difficult or impossible to obtain such information.

Our proposed SAN framework utilizes a domain-agnostic mechanism to dynamically adjust strategies based on interaction sequences, enabling effective collaboration with diverse, unseen partners.This goal is also related to preserving transferability in learned representations and policies [10], [15], [35].

III. Preliminaries

We model the two-player-environment interaction by a discounted infinite-horizon Markov Decision Process

$\mathcal{M} =< \mathcal{S}, \mathcal{A}^{(i)}, \mathcal{R}, \mathcal{P}, \mu_0, \gamma >$, where $\mathcal{S}$ denotes the state space, $\mathcal{A}^i$ is the action space of player i where $i \in \{1, 2\}$, $\mathcal{R}(S, A^1, A^2)$ denotes the reward function , $\mathcal{P} : S \times A \times S \to [0, 1]$ is the transition kernel, μ_0 is the initial state distribution, and $\gamma \in (0, 1)$ is the discount factor.

Under a classic RL setting, an action-value function Q is used to represent the expected return as $Q^\pi(\boldsymbol{s}, \boldsymbol{a}) = \mathbb{E}\left[Z^\pi(\boldsymbol{s}, \boldsymbol{a})\right]$, where the expectation takes over all sources of intrinsic randomness. While under the distributional setup, it is the random return $Z^\pi(\boldsymbol{s}, \boldsymbol{a})$ itself rather than its expectation that is being directly,which has been shown effective in a range of reinforcement learning algorithms [36]–[38]. Similar to the classical Bellman equation, $Z^\pi(\boldsymbol{s}, \boldsymbol{a})$ follows the distributional Bellman equation:

$$Z^\pi(\boldsymbol{s}, \boldsymbol{a}) \stackrel{D}{=} R(\boldsymbol{s}, \boldsymbol{a}) + \gamma Z^\pi\left(\boldsymbol{s}', \boldsymbol{a}'\right). \tag{1}$$

where $\stackrel{D}{=}$ denotes "equal in distribution" and $\boldsymbol{a}' \sim \pi\left(\cdot \mid \boldsymbol{s}'\right), \boldsymbol{s}' \sim P(\cdot \mid \boldsymbol{s}, \boldsymbol{a})$.

The return distribution can be modeled by a network $Z_\tau(s, a; \theta)$. Let F_Z^{-1} denote the quantile function for random variable Z, which represents the inverse function of the cumulative distribution function $F_Z(z) = \Pr(Z < z)$, i.e. $F_Z^{-1}(\tau) := \inf\{z \in \mathbb{R} : \tau \le F_Z(z)\}$, where τ denotes the quantile fraction. In the following, we denote $Z_\tau := F_Z^{-1}(\tau)$. Given state s and action a, the action-value distribution is approximated by a number of quantile values at quantile fractions. Let $\{\tau_i\}_{i=0,\ldots,N}$ denote an ensemble of quantile fractions, which satisfy $\tau_0 = 0, \tau_N = 1, \tau_i < \tau_j, \forall i < j$, and $\tau_i \in [0, 1], i = 0, \ldots, N$. We further denote $\hat{\tau}_i = \left(\tau_i + \tau_{i+1}\right)/2$. Based on eq.1, the pairwise temporal difference (TD) error between two quantile fractions $\hat{\tau}_i$ and $\hat{\tau}_j$ is defined as

$$\Delta_{ij}^t = r_t + \gamma Z_{\hat{\tau}_i}(s_{t+1}, a_{t+1}; \bar{\theta}) - Z_{\hat{\tau}_j}\left(s_t, a_t; \theta\right), \tag{2}$$

The Huber regression loss is commonly adopted to train the quantile network, which is defined as,

$$\begin{aligned} \rho_\tau^\kappa\left(\Delta_{ij}\right) &= \left|\tau - \mathbb{I}\left\{\Delta_{ij} < 0\right\}\right| \frac{\mathcal{L}_\kappa\left(\Delta_{ij}\right)}{\kappa}, \\ \text{with} \quad \mathcal{L}_\kappa\left(\Delta_{ij}\right) &= \begin{cases} \frac{1}{2}\Delta_{ij}^2, & \text{if } \left|\Delta_{ij}\right| \le \kappa, \\ \kappa\left(\left|\Delta_{ij}\right| - \frac{1}{2}\kappa\right), & \text{otherwise.} \end{cases} \end{aligned} \tag{3}$$

Thus the objective of quantile value network $Z_\tau(s, a; \theta)$ is given by

$$J_Z(\theta) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \left(\tau_{i+1} - \tau_i\right) \rho_{\hat{\tau}_j}^\kappa\left(\Delta_{ij}^t\right) \tag{4}$$

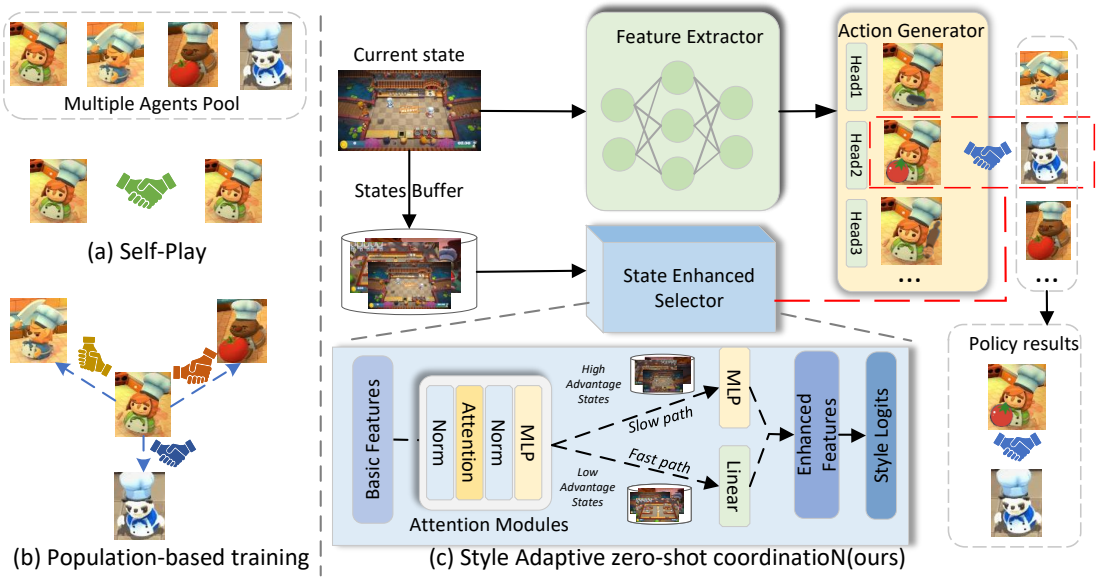


Fig. 1: Overall framework of SAN. (a) Self-play training (SP). The partner policy pool consists of SP policies. (b) Population-based training (PBT). A partner is sampled from population at each iteration to cooperate with the ego agent. (c) The main structure of our proposed SAN method consists of three components, i.e., a neural network-based feature extractor, a multi-head action generator, and a state-enhanced selector.

where κ is a constant threshold, and each $\rho_{\hat{\tau}_j}$ is weighted by the target distribution fractions $\tau_{i+1} - \tau_i$.

IV. Methodology

In this section, we present our Style Adaptive zero-Shot coordinatioN (SAN) Algorithm, where a partner policy pool is first constructed by running self-play repeatedly with PPO and then an ego agent is trained against this pool. Fig.1(c) illustrates the overall structure of our SAN. Our framework is composed of the following key components:

- **The Feature Extractor**, where a convolutional neural network is used to extract the basic features of the current state.
- **The Multi-Head Action Generator**, which generates actions cooperating well with partners of different styles with different heads based on the latent state features.
- **The State-Enhanced Selector**, which is based on a Transformer architecture, is designed to recognize the partner's style from historical state information and select the corresponding head in the action generator.

In the training of SAN, each head in the action generator is trained to collaborate effectively with partners of a specific style in the policy pool, and the state-enhanced selector is trained to distinguish the partner's style based on state sequences. As a result, the agent is able to recognize the partner's style and choose the appropriate action head to collaborate with the partner, when collaborating with a new partner. To further improve the performance of SAN, we also propose a Stylized Policy Pool construction scheme and fast-slow path strategy in the state-enhanced selector, which will be described in the subsequent subsections.

A. Stylized Policy Pool in SAN

In the partner policy pool construction stage, we model the distribution of the action value function with a quantile network, and utilize the distorted quantile fractions of the value function to modify the PPO objective. Here, we parameterize the value distribution Z of a policy π with $Z_\tau(s,a;\theta)$. Given a set of quantile fractions $\{\tau_i\}_{i=0,\ldots,N}$, which satisfy $\tau_0 = 0, \tau_N = 1$,and $\tau_i < \tau_j, \forall 0 \le i < j \le N$, the value distribution can be written as

$$Z_\pi(s,a) := \sum_{i=0}^{N-1} (\tau_{i+1} - \tau_i)\delta_{z_{\hat{\tau}_i}(s,a;\theta)} \tag{5}$$

where δ_z is a Dirac at $z \in \mathcal{R}$ and $\hat{\tau}_i = (\tau_i + \tau_{i+1})/2$. Here, the parameter θ can be optimized w.r.t. the objective (4). The distorted expectation is computed as

$$Q_\beta(s,a) = \sum_{k=1}^{N} \left(\psi_\beta(\tau_k) - \psi_\beta(\tau_{k-1})\right) z_{\hat{\tau}_i}(s,a;\theta) \tag{6}$$

ψ is the utility function to compute the distorted expectation. In this work, we adopt Wang metric [39] to model different policies, and compute the distorted quantile fractions as

$$\psi_\beta^{Wang}(\tau) = \Phi(\Phi^{-1}(\tau) + \beta) \tag{7}$$

where Φ is the standard normal distribution and β is the scalar parameter. Typically, we can choose $\beta = 0$ for neutral policies

, $\beta > 0$ for aggressive policies , and $\beta < 0$ for conservative policies.

This allows us to construct partner policies with different styles (aggressive, neutral and conservative) without leveraging domain knowledge. Specifically, we first sample $\beta \in [-1.5, 1.5]$ and then train different policies by self-play using the following β-distorted PPO clipped objective.

$$\mathcal{L} = \min\left(\frac{\pi_\phi(s|a;\beta)}{\pi_{\phi_{old}}(s|a;\beta)} A^{\pi_{\phi_{old}}}(s,a;\beta), \psi(\epsilon, A^{\pi_{\phi_{old}}}(s,a;\beta)),\right)$$

where

$$\psi(\epsilon, A) = \begin{cases}(1+\epsilon)A, & if A \geq 0,\\ (1-\epsilon)A, & if A < 0.\end{cases}$$

and $A_\pi(s_t, a_t; \beta)$ denotes the distorted advantages

$$A^\pi(s_t, a_t; \beta) = \sum_{l=0} (\lambda\gamma)^l \delta^r_{t+l} \tag{8}$$

with $\delta^r_t = r_t + \gamma Q_\beta(s_{t+1}, a_{t+1}) - Q_\beta(s_t, a_t)$. Note that we include three checkpoints in each self-play training procedure in the stylized policy pool: the first checkpoint (i.e. a randomly initialized agent), the last checkpoint (i.e. a fully trained agent), and the remaining checkpoint that achieves closest to half of the reward of the last checkpoint (i.e. a half-trained agent).

B. State-Enhanced Selector with Dual-Path Inference

Fig.1(c) illustrates the structure of the state-enhanced selector module, where we utilize a fast-slow path technique to prioritize "informative" interactions for style inference. This module processes all states stored in the state buffer as input. Initially, basic features are extracted from these states. Attention modules are then employed to capture the stylistic information within the sequence. The features of the different states s are subsequently categorized into *informative* and *regular* groups based on the deviation between $Q(s_t, a_t^{(p)})$ and $V(s_t)$, where $a_t^{(p)} \sim \pi(\cdot \mid s_t)$ denotes the partner's action at s_t. Here, π represents the partner policy, while Q and V denote the state-action value function and the state value function, respectively. Furthermore, we quantify this action informativeness as $|Q(s_t, a_t^{(p)}) - V(s_t)|$. A larger deviation indicates that the partner's choice at s_t has a stronger influence on the task outcome, making it more representative of their underlying style.

Within the attention modules of the selector, states in different groups are processed using distinct paths. For important states, a complex, multi-layer slow path is employed to extract hidden features. In contrast, a single-layer fast path is used to handle the less significant regular states. Finally, all features are reassembled into their original positions within the sequence, guided by the state mask. This process can be formally expressed as follows:

$$S^3 = \text{Reassembled}(\text{Slow}(S^1), \text{Fast}(S^2)) \tag{9}$$

where S^1 and S^2 represent important and regular state features. Reassembled$(\cdot)$ represents the reassembling operation, we directly use concatenation for reassembly here. Slow$(\cdot)$ and Fast$(\cdot)$ represent the slow path and fast path, respectively. The essential difference between the two paths is that one passes through multiple layers of linear transformations, while the other passes through only one layer. This allows the state-enhanced selector module to pay more attention to the critical states in the trajectory and infer the partner's strategy style based on these states.

V. EXPERIMENTS

To evaluate the effectiveness of our proposed SAN algorithm, we conducted a series of experiments in Overcooked [5]. It is a cooperative multiplayer game that emphasizes teamwork and coordination in a fast-paced kitchen setting. In this environment, players assume the roles of chefs working together to prepare and serve a variety of dishes under time constraints. We compare our algorithm with the SOTA PBT algorithms including FCP [15], MEP [24], TrajeDiv [12], and HSP [17], all implemented based on the HSP code repository [1]. Besides, we also include the ability-aware PECAN method [34] as our baseline. First, we assess the zero-shot coordination capability of these algorithms by having the trained agents collaborate with human proxy models obtained through behavior cloning with human data. To validate the style-adaptive property of our proposed algorithm, we also compare the performance of different methods when cooperating with partners of random/shifting styles. Then, we conduct ablation studies to demonstrate the effectiveness and contribution of each module in SAN. Besides, we also visualize action distribution of different agents in the partner policy pool and the style recognition accuracy of the selector in SAN to further validate the effectiveness of the stylized policy pool and the selector, respectively.

A. Collaborative Evaluation with Learned Human Proxy Models

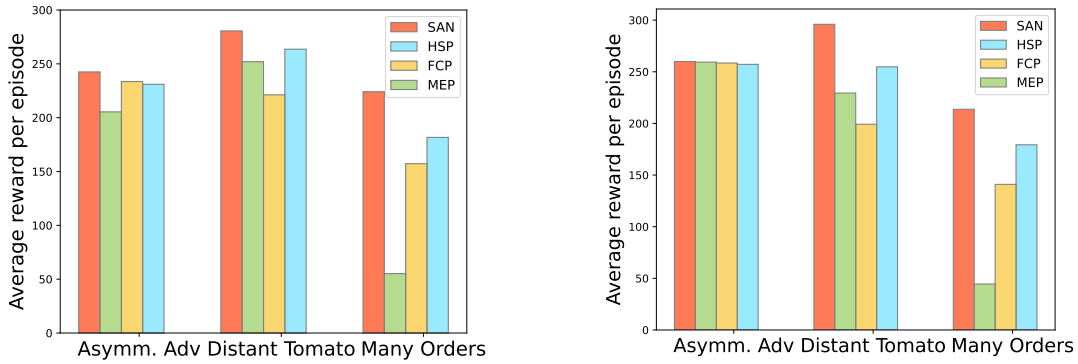
Table. I presents a comprehensive comparison of the coordination performance of various agent models, including SAN, HSP, FCP, MEP, TrajeDiv, and PECAN, when cooperating with a human proxy model. Here, we choose three different one-recipe layouts in the Overcooked environment, i.e., Asymmetric Advantages, Coordination Ring, and Counter Circuit, with increasing difficulty levels. In the construction of the human proxy, we use behavior-cloned model that released in the code of [5]. To ensure a fair comparison across all models, we maintained an identical population size for each method. Furthermore, to account for potential variability, each algorithm was executed multiple times with different random seeds, and the reported results reflect the average performance across these runs.

The results clearly indicate that SAN significantly outperforms the other baseline models, particularly when evaluated within the Counter Circuit/Coordination Ring layouts, which are known for their high demand for effective collaboration.

[1]https://github.com/samjia2000/HSP

TABLE I: Comparison of different methods cooperating with proxy human on the Overcooked task with different layouts. The results of HSP,FCP,MEP and TrajeDiv are taken from [17],and PECAN is taken from [34]. Our proposed approach achieves state-of-the-art performance, demonstrating its effectiveness when collaborating with a proxy human. The best results are highlighted in **bold**.

Method	HSP	FCP	MEP	TrajDiv	PECAN	SAN (Ours)
Asymmetric Advantages	300.3	282.8	291.7	289.3	150.0	**313.3**
Coordination Ring	160.0	161.3	161.8	150.8	130.5	**173.3**
Counter Circuit	107.4	95.9	108.8	60.1	60.0	**118.6**



(a) Average reward with Random-style partner

(b) Average reward with Shifting-styles policy

Fig. 2: Average episode rewards of different methods when cooperating with unseen partners of random/shifting styles.

Even in the less collaboration-intensive Asymmetric Advantage layout, SAN continues to demonstrate strong performance, highlighting its versatility across different levels of cooperative requirements. One of the critical insights from this evaluation is SAN's ability to enhance cooperative behavior, even when working with fixed learned human proxy that has a relatively static strategy style. This is mainly attributes to the dynamic strategy adjustment capability of SAN, which is able to locally choose the best action (head) from strategies trained against the partner pool to match unseen partners. This superior cooperative capability makes SAN a more robust and adaptable agent in diverse interaction scenarios, offering significant advantages over existing methods for unseen partners even with fixed strategy styles.

B. Collaborative Evaluation with partners of random/shifting styles

To validate SAN's ability to cooperate with partners with different styles, we develop two novel partners termed Random-style partner and Shifting-style partner, designed to introduce greater variability and unpredictability into the interaction dynamics. In this evaluation, we select the Distant Tomato and the Many Orders layouts, two particularly challenging environments which contain two distinct recipes. The complexity introduced by multiple recipes provides a more stringent test of our method's ability to adapt to varying conditions. Note that we exclude PECAN and TrajDiv in this comparison as they perform worst even when cooperating with the human proxy as indicated in the first experiment.

Random-style partner:The strategy of the Random-style partner is composed of four distinct preference policies, each reflecting different patterns of behavior. Specifically, these policies are constructed by integrating the two categories of primary actions: 1) consistently picking up onions and placing them into a pot v.s. randomly placing them, 2) consistently carrying soup and delivering it continuously v.s. placing it randomly. By introducing these variations, the strategy not only diversifies the agent's possible actions but also simulates a wider range of behaviors that are not aligned with the existing policies in the partner policy pool. Each of these four policies is executed with equal probability, ensuring that the behavior is both unpredictable and challenging for the agents. This randomness in the Random-style partner requires the agent to adapt to highly dynamic scenarios that they might encounter when interacting with human partners. In Figure 2a, we report the average reward achieved by all methods when paired with the Random-style partner. The results reveal that SAN consistently outperforms all baseline models. This superior performance underscores the effectiveness of SAN in adapting to varying and unpredictable strategies, a capability that is crucial for achieving robust and reliable cooperation in complex, real-world scenarios.

Shifting-style partner: The strategy of Shifting-style partner is obtained by randomly selecting one policy from a predefined unseen policy set every 50 steps. This partner is intentionally constructed to simulate the real-world variability and test the robustness of our strategy under conditions of frequent change. As illustrated in Figure 2b, our approach consistently outperforms other methods, with the performance gap becoming

TABLE II: Ablation results to show the effect of each component in SAN

Method	Performance
SAN *w.o fast-slow path*	111.0
SAN *w.o. stylized policy pool and fast-slow path*	131.6
SAN *w.o stylized policy pool*	137.6
SAN *w.o selector and fast-slow path*	151.8
SAN	173.3

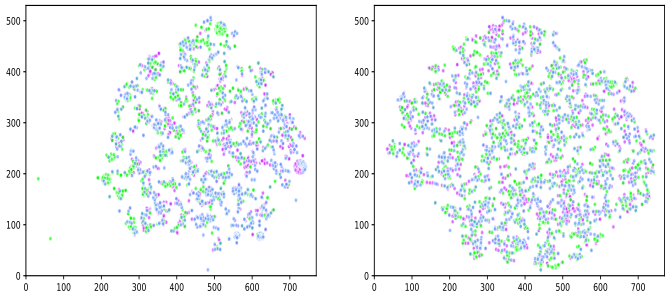


Fig. 3: The t-SNE results of partners' policy $\pi_p(\cdot|s)$ in HSP and SAN.The results show our proposed stylized policy pool (right) exhibits more diversity in strategies compared to the HSP policy pool (left).

increasingly evident as the complexity of the environment escalates. It is important to note that the partner strategies utilized in this evaluation were exclusively employed for testing purposes and were not part of the training process. This result highlights the ability of SAN to dynamically respond to changes in the partner's strategy, thereby demonstrating its superior adaptability in complex, unpredictable environments.

C. Ablation Study

To study the effect of each component in SAN, we ablate the stylized policy pool and the state-enhanced selector of SAN, respectively. Note that to show the importance of the fast/slow path in our selector, we also include it as an ablation module and use "selector" to merely represent a vanilla transformer-based selector module without the fast/slow path, where "selector" + "fast/slow path" equals the whole state-enhanced selector component in SAN.

In Table.II, we report the performance of SAN without different components. The results validate the effectiveness of each component in SAN. It can be seen that the stylized policy pool contributes the most to the overall performance, and both the transformer structure and the fast-slow path are significant in the state-enhanced selector of SAN. Note that it is interesting to observe that merely removing the fast-slow path in SAN would greatly reduce performance, resulting in an overall score even much lower than SAN without the whole state-enhanced selector component. The main reason may lie in that the single gate network in the vanilla transformer-based selector is insufficient for classifying a more diverse policy pool in SAN.

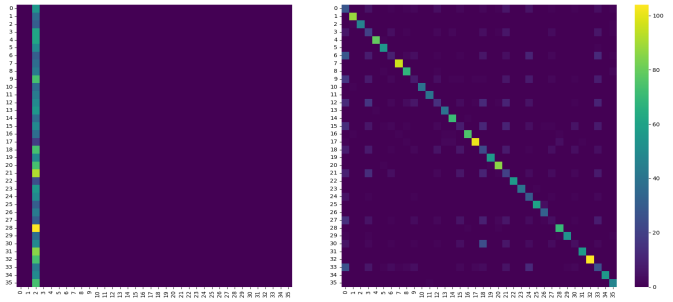


Fig. 4: Performance of Selector: The block in the i-th row j-th column represents the percentage of the ego agent's selector in choosing the j-th head when cooperating with the i-th partner. The left heat map represents the prediction results at the beginning of whole episode , the right one is the later stage of the same episode. It shows that selector tends to recognize states into random same style at the beginning and can finally demonstrate good performance.

D. Diversity of the Stylized Policy Pool

In this subsection, we empirically validate that the stylized policy pool does contain partners with diverse action styles. Here, we first randomly sample several states s from five interaction trajectories, and then use t-SNE to project the action distribution $\pi(\cdot|s)$ of the entire partner policy pool to a 2-dimensional feature space. Each point in the feature space represents the action distribution for a specific strategy in the policy pool for certain state s. Fig. 3 shows the visualization results. The left sub-figure shows the result of the HSP policy pool, which models human biases as domain-knowledge-based hidden reward functions in the construction of the policy pool. Compared to the result of SAN in the right sub-figure, the points in the left subfigure are more concentrated, which indicates that the policy pool in SAN has a wider diversity. The diversity in the stylized policy pool of SAN would enable the ego agent to encounter a wider variety of strategies, thereby enhancing its cooperative capabilities.

E. Performance of Selector

To show the capability of the state-enhanced selector in recognizing the style of different partners, we investigate the style prediction result of our trained ego agent when collaborating with agents from the policy pool in different stages (early and final). In Figure 4, we report the prediction results of our ego agent at the beginning of cooperation in the left heat map, and the results in the later stage in the right one. In these heat maps, the block in the i-th row j-th column represents the percentage of the ego agent's selector in choosing the j-th head when cooperating with the i-th partner. It shows that state-enhanced selector in SAN tends to recognizes its partner into same style at the beginning and can finally achieve good recognition performance as most of the percentages concentrate on the diagonal.

VI. Conclusion

In this paper, we introduced SAN, a style-adaptive population-based training framework for zero-shot coordination with non-stationary partners. SAN infers partner style online and dynamically selects actions to adapt to changing behaviors. By constructing a stylized partner pool with a risk-sensitive distributional utility, SAN encourages behavioral diversity in self-play. Experiments on Overcooked demonstrate that SAN outperforms prior ZSC methods in both stationary and dynamic settings.

References

[1] G. Klien, D. D. Woods, J. M. Bradshaw, R. R. Hoffman, and P. J. Feltovich, "Ten challenges for making automation a" team player" in joint human-agent activity," *IEEE Intelligent Systems*, vol. 19, no. 6, pp. 91–95, 2004.

[2] A. Ajoudani, A. M. Zanchettin, S. Ivaldi, A. Albu-Schäffer, K. Kosuge, and O. Khatib, "Progress and prospects of the human–robot collaboration," *Autonomous robots*, vol. 42, pp. 957–975, 2018.

[3] A. Dafoe, Y. Bachrach, G. Hadfield, E. Horvitz, K. Larson, and T. Graepel, "Cooperative ai: machines must learn to find common ground," 2021.

[4] T. B. Sheridan, "Human–robot interaction: status and challenges," *Human factors*, vol. 58, no. 4, pp. 525–532, 2016.

[5] M. Carroll, R. Shah, M. K. Ho, T. Griffiths, S. Seshia, P. Abbeel, and A. Dragan, "On the utility of learning about humans for human-ai coordination," *Advances in neural information processing systems*, vol. 32, 2019.

[6] A. Bobu, D. R. Scobee, J. F. Fisac, S. S. Sastry, and A. D. Dragan, "Less is more: Rethinking probabilistic models of human behavior," in *Proceedings of the 2020 acm/ieee international conference on human-robot interaction*, 2020, pp. 429–437.

[7] C. D. Kidd and C. Breazeal, "Robots at home: Understanding long-term human-robot interaction," in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2008, pp. 3230–3235.

[8] F. Biondi, I. Alvarez, and K.-A. Jeong, "Human–vehicle cooperation in automated driving: A multidisciplinary review and appraisal," *International Journal of Human–Computer Interaction*, vol. 35, no. 11, pp. 932–946, 2019.

[9] X. Pan, W. Wang, X. Zhang, B. Li, J. Yi, and D. Song, "How you act tells a lot: Privacy-leaking attack on deep reinforcement learning." in *AAMAS*, vol. 19, no. 2019, 2019, pp. 368–376.

[10] S. Gao, Y. Fu, K. Liu, and Y. Han, "Contrastive knowledge amalgamation for unsupervised image classification," in *International Conference on Artificial Neural Networks*. Springer, 2023, pp. 192–204.

[11] H. Hu, A. Lerer, A. Peysakhovich, and J. Foerster, ""other-play" for zero-shot coordination," in *International Conference on Machine Learning*. PMLR, 2020, pp. 4399–4410.

[12] A. Lupu, B. Cui, H. Hu, and J. Foerster, "Trajectory diversity for zero-shot coordination," in *International conference on machine learning*. PMLR, 2021, pp. 7204–7213.

[13] H. Hu, A. Lerer, B. Cui, L. Pineda, N. Brown, and J. Foerster, "Off-belief learning," in *International Conference on Machine Learning*. PMLR, 2021, pp. 4369–4379.

[14] B. Cui, H. Hu, L. Pineda, and J. Foerster, "K-level reasoning for zero-shot coordination in hanabi," *Advances in Neural Information Processing Systems*, vol. 34, pp. 8215–8228, 2021.

[15] D. Strouse, K. McKee, M. Botvinick, E. Hughes, and R. Everett, "Collaborating with humans without human data," *Advances in Neural Information Processing Systems*, vol. 34, pp. 14 502–14 515, 2021.

[16] K. Lucas and R. E. Allen, "Any-play: An intrinsic augmentation for zero-shot coordination," *arXiv preprint arXiv:2201.12436*, 2022.

[17] C. Yu, J. Gao, W. Liu, B. Xu, H. Tang, J. Yang, Y. Wang, and Y. Wu, "Learning zero-shot cooperation with humans, assuming humans are biased," in *The Eleventh International Conference on Learning Representations*, 2023.

[18] N. Brown and T. Sandholm, "Superhuman ai for multiplayer poker," *Science*, vol. 365, no. 6456, pp. 885–890, 2019.

[19] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel *et al.*, "A general reinforcement learning algorithm that masters chess, shogi, and go through self-play," *Science*, vol. 362, no. 6419, pp. 1140–1144, 2018.

[20] W. Fu, W. Du, J. Li, S. Chen, J. Zhang, and Y. Wu, "Iteratively learn diverse strategies with state distance information," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[21] R. Charakorn, P. Manoonpong, and N. Dilokthanakul, "Generating diverse cooperative agents by learning incompatible policies," in *The Eleventh International Conference on Learning Representations*, 2023.

[22] M. Jaderberg, V. Dalibard, S. Osindero, W. M. Czarnecki, J. Donahue, A. Razavi, O. Vinyals, T. Green, I. Dunning, K. Simonyan *et al.*, "Population based training of neural networks," *arXiv preprint arXiv:1711.09846*, 2017.

[23] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[24] R. Zhao, J. Song, Y. Yuan, H. Hu, Y. Gao, Y. Wu, Z. Sun, and W. Yang, "Maximum entropy population-based training for zero-shot human-ai coordination," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 5, 2023, pp. 6145–6153.

[25] J. M. Bradshaw, P. J. Feltovich, and M. Johnson, "Human–agent interaction," in *The handbook of human-machine interaction*. CRC Press, 2017, pp. 283–300.

[26] S. Nikolaidis and J. Shah, "Human-robot cross-training: Computational formulation, modeling and evaluation of a human team training strategy," in *2013 8th ACM/IEEE international conference on human-robot interaction (HRI)*. IEEE, 2013, pp. 33–40.

[27] S. Javdani, S. S. Srinivasa, and J. A. Bagnell, "Shared autonomy via hindsight optimization," *Robotics science and systems: online proceedings*, vol. 2015, 2015.

[28] D. Sadigh, S. Sastry, S. A. Seshia, and A. D. Dragan, "Planning for autonomous cars that leverage effects on human actions." in *Robotics: Science and systems*, vol. 2. Ann Arbor, MI, USA, 2016, pp. 1–9.

[29] R. Choudhury, G. Swamy, D. Hadfield-Menell, and A. D. Dragan, "On the utility of model learning in hri," in *2019 14th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. IEEE, 2019, pp. 317–325.

[30] Y. Li, S. Zhang, J. Sun, Y. Du, Y. Wen, X. Wang, and W. Pan, "Cooperative open-ended learning framework for zero-shot coordination," in *International Conference on Machine Learning*. PMLR, 2023, pp. 20 470–20 484.

[31] C. Guan, L. Zhang, C. Fan, Y. Li, F. Chen, L. Li, Y. Tian, L. Yuan, and Y. Yu, "Efficient human-ai coordination via preparatory language-based convention," *arXiv preprint arXiv:2311.00416*, 2023.

[32] T. Gessler, T. Dizdarevic, A. Calinescu, B. Ellis, A. Lupu, and J. N. Foerster, "Overcookedv2: Rethinking overcooked for zero-shot coordination," in *The Thirteenth International Conference on Learning Representations*.

[33] X. Wang, S. Zhang, W. Zhang, W. Dong, J. Chen, Y. Wen, and W. Zhang, "Zsc-eval: An evaluation toolkit and benchmark for multi-agent zero-shot coordination," *Advances in Neural Information Processing Systems*, vol. 37, pp. 47 344–47 377, 2024.

[34] X. Lou, J. Guo, J. Zhang, J. Wang, K. Huang, and Y. Du, "Pecan: Leveraging policy ensemble for context-aware zero-shot human-ai coordination," in *22nd International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2023*, 2023, pp. 679–688.

[35] S. Gao, Y. Fu, K. Liu, W. Gao, H. Xu, J. Wu, and Y. Han, "Collaborative knowledge amalgamation: Preserving discriminability and transferability in unsupervised learning," *Information Sciences*, vol. 669, p. 120564, 2024.

[36] M. G. Bellemare, W. Dabney, and R. Munos, "A distributional perspective on reinforcement learning," in *International conference on machine learning*. Pmlr, 2017, pp. 449–458.

[37] W. Dabney, G. Ostrovski, D. Silver, and R. Munos, "Implicit quantile networks for distributional reinforcement learning," in *International conference on machine learning*. PMLR, 2018, pp. 1096–1105.

[38] W. Bai, C. Zhang, Y. Fu, P. Zhao, and H. Qian, "Pacer: A fully push-forward-based distributional reinforcement learning algorithm," *Neurocomputing*, p. 131301, 2025.

[39] S. S. Wang, "A class of distortion operators for pricing financial and insurance risks," *Journal of risk and insurance*, pp. 15–36, 2000.