IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 18 July 2025, accepted 1 August 2025, date of publication 7 August 2025, date of current version 20 August 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3596640

RESEARCH ARTICLE

Automatic Curriculum Design for Zero-Shot Human-AI Coordination

WON-SANG YOU[1], TAE-GWAN HA[2], SEO-YOUNG LEE[1], AND KYUNG-JOONG KIM[1], (Member, IEEE)

[1]Department of AI Convergence, Gwangju Institute of Science and Technology, Gwangju 61005, South Korea
[2]School of Integrated Technology, Gwangju Institute of Science and Technology, Gwangju 61005, South Korea

Corresponding author: Kyung-Joong Kim (kjkim@gist.ac.kr)

This work was supported in part by Institute of Information and Communications Technology Planning and Evaluation (IITP) Grant funded by the Ministry of Science and Information and Communication Technology (MSIT), South Korea, under Grant RS-2025-25410841; in part by the Culture, Sports and Tourism Reasearch and Development Program through Korea Creative Content Agency Grant funded by the Ministry of Culture, Sports and Tourism, in 2025, under Project RS-2024-00441523; in part by IITP Grant funded by Korean Government (MSIT) under Grant 2019-0-01842; in part by the Artificial Intelligence Graduate School Program (Gwangju Institute of Science and Technology); and in part by the National Research Foundation of Korea (NRF) grant Funded by the Korea Government (MSIT) under Grant RS-2025-16902996.

This work involved human subjects or animals in its research. Approval of all ethical and experimental procedures and protocols was granted by the Institutional Review Board of Gwangju Institute of Science and Technology under Approval No. 20240807-HR-77-02-02.

ABSTRACT Zero-shot human-AI coordination is the training of an ego-agent to coordinate with humans without human data. Most studies on zero-shot human-AI coordination have focused on enhancing the ego-agent's coordination ability in a given environment without considering the issue of generalization to unseen environments. Real-world applications of zero-shot human-AI coordination should consider unpredictable environmental changes and the varying coordination ability of co-players depending on the environment. Previously, the multi-agent UED (Unsupervised Environment Design) approach has investigated these challenges by jointly considering environmental changes and co-player policy in competitive two-player AI-AI scenarios. In this paper, our study extends a multi-agent UED approach to zero-shot human-AI coordination. We propose a utility function and co-player sampling for a zero-shot human-AI coordination setting that helps train the ego-agent to coordinate with humans more effectively than a previous multi-agent UED approach. The zero-shot human-AI coordination performance was evaluated in the Overcooked-AI environment, using human proxy agents and real humans. Our method outperforms other baseline models and achieves high performance in human-AI coordination tasks in unseen environments. The source code is available at https://github.com/Uwonsang/ACD_Human-AI

INDEX TERMS Human-AI coordination, reinforcement learning, multi-agent.

I. INTRODUCTION

Deep reinforcement learning has achieved remarkable success in diverse domains such as gaming [1], [2], [3], autonomous vehicles [4], and robotic controls [5]. Research in deep reinforcement learning has expanded to multi-agent reinforcement learning, where multiple agents cooperate or compete to achieve specific objectives [6], [7]. Most studies in this field have focused on developing powerful AI beyond human performance through AI-AI interactions. Recently, interest has shifted to problem-solving through human-AI interaction.

Some human-AI interaction studies focus on training agents that coordinate with humans in common tasks and problem-solving. These studies typically require a large amount of human data for training, but acquiring sufficient human data is challenging in real-world applications. To overcome this limitation, a zero-shot human-AI coordination study [8] suggests an approach that trains the ego-agent to coordinate with humans without human data. This approach mainly uses population-based training (PBT) [8], [9], [10], [11]. During training, the ego-agent coordinates with various co-player agents from the population pool, rather

The associate editor coordinating the review of this manuscript and approving it for publication was Yang Tang.

than with a specific co-player agent. This prevents overfitting to a specific coordination strategy and helps generalize across diverse human coordination strategies.

Unsupervised environment design (UED) [12] is a method that selectively curates environments for the agent by using a curriculum based on a given utility function. It curates increasingly difficult environments based on the current agent's policy. This approach allows the agent to develop the ability to generalize across all possible environments by experiencing diverse and challenging scenarios. These methods [13], [14], [15] mainly use regret as a utility function. Recently, UED methods have been extended to competitive multi-agent settings [16]. In a multi-agent setting, the co-player policy varies depending on the environment; therefore, the multi-agent UED curates joint environment/co-player pairs for each agent. This approach has successfully trained robust agents across diverse environments and co-players. By extending this method to the zero-shot human-AI coordination setting, we could train a robust agent that can coordinate with real humans across diverse environments.

However, existing approaches in zero-shot human-AI coordination only focus on unseen co-players and do not consider generalization to unseen environments. Real-world applications of human-AI coordination face various unpredictable variables and situations, including diverse partners and changing environmental conditions. Therefore, it is essential to train the ego-agent to be robust against both diverse co-players and environments. While UED methods could potentially address this challenge by extending to zero-shot human-AI coordination settings, the existing multi-agent UED method is not directly applicable to coordination tasks. This method was developed for competitive zero-sum games and does not match the collaborative nature of human-AI coordination tasks. This mismatch motivates the need for a specialized approach that combines the benefits of zero-shot coordination and unsupervised environment design while addressing their respective limitations.

To address these problems, we introduce the automatic curriculum design for zero-shot human-AI coordination. We adopt a return-based utility function and co-player sampling for zero-shot human-AI coordination. The return measures the ego-agent's coordination performance with co-players. It helps the agent learn better coordination strategies even in challenging scenarios and with difficult co-players. The contributions of our study are as follows: (1) We propose a multi-agent UED framework for zero-shot human-AI coordination that uses a return-based utility function and co-player sampling strategy. (2) We show that our method demonstrates higher coordination performance with human proxy agents and real human partners compared to other baselines. Figure 1 shows an overview of our proposed method.

The rest of the paper is organized as follows. Section I provides a review of related work in unsupervised environment design and zero-shot human-AI coordination. Section III presents our proposed automatic curriculum design method, including the return-based utility function and the co-player sampling strategy. Section IV describes the experimental setup and evaluation metrics. Section V presents the experimental results with human proxy agents and real human participants. Finally, Section VI concludes the paper and discusses future research directions.

II. RELATED WORKS

A. UNSUPERVISED ENVIRONMENT DESIGN STRATEGIES

Unsupervised Environment Design (UED) is a method in which the teacher curates a curriculum to enable the student agent to generalize across all possible levels of the environment based on a utility function. The simplest version of the UED teacher uses the utility function as a constant C for uniform random sampling of the environment level $U_t(\pi, \theta) = C$. Here, t is the teacher, π is the student's policy, θ is the environment level parameter [17]. Most UED studies use regret as a teacher's utility function, defined as the difference between the expected return of the current policy and the optimal student policy. The teacher aims to maximize this regret [12], [18], which can be defined as: $U_t(\pi, \theta) = \max_{\pi^* \in \Pi}\{\text{Regret}^{\theta}(\pi, \pi^*)\} = \max_{\pi^* \in \Pi}\{V_{\theta}(\pi^*) - V_{\theta}(\pi)\}$, where π^* is the optimal policy in θ, Π is the set of student policies. This regret-based utility function encourages the teacher to curate challenging environments for the student agent. Through the training process, the student's policy will converge to the minimax regret policy by minimizing regret in teacher-curated environments [12]: $\pi \in \arg\min_{\pi \in \Pi}\{\max_{\theta, \pi^* \in \Theta, \Pi}\{\text{Regret}^{\theta}(\pi, \pi^*)\}\}$ where Θ is the set of environment- level parameters. Since the optimal policy π^* cannot be accessed in practice, the regret must be approximated.

Several UED studies have explored the use of regret. Prioritized Level Replay (PLR) [14] and Robust PLR [13] leverage a regret-based utility function to train the agent by selectively sampling environments with high learning potential. ACCEL [15] continuously edits the environment that pushes the boundaries of the student agent's capabilities by combining an evolutionary approach with Robust PLR. ADD [19] generates diverse and challenging environments using diffusion models with entropy regularization, while No Regrets [20] addresses limitations of regret approximation by introducing Sampling for Learnability, which prioritizes environments where agents succeed approximately 50%.

MAESTRO [16] extends single-agent UED to multi-agent UED in a competitive setting by curating environment/co-player pairs while considering the environment-dependent co-player policy. Recently, the Overcooked Generalisation Challenge (OGC) [21] highlighted generalization challenges in human-AI coordination, showing that existing UED-based methods struggle to achieve zero-shot coordination with novel partners and environments.

However, most existing methods are primarily effective in single-agent and competitive settings. Regret-based utility functions are not suitable for human-AI coordination tasks,

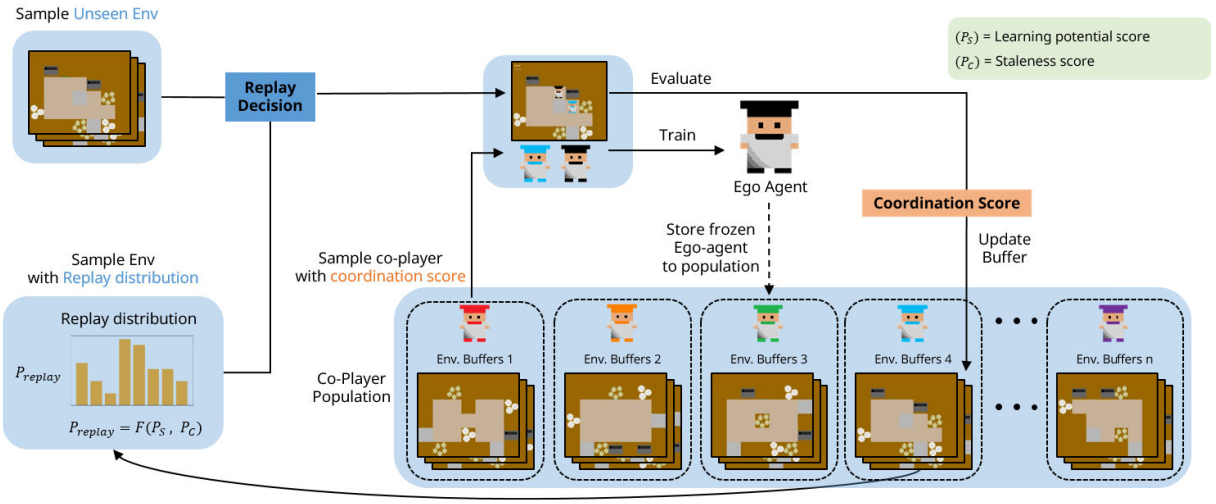


FIGURE 1. Overview of the proposed method. The ego-agent is trained using a population of co-players whose individual environment buffers maintain a fixed size of *K*, containing the environments with the lowest coordination scores. Our approach samples the co-player based on their coordination scores within the population. Depending on the replay decision, the environment is either sampled from the unseen environment buffer or the co-player's buffer, guided by a replay distribution that prioritizes environment sampling based on coordination scores. After sampling, our method calculates the return from the episode trajectory within the sampled environment, which is used to compute the coordination score. This coordination score is then updated within the co-player's buffer.

where coordination and common rewards are essential, as they do not directly reflect collaborative success with diverse partners. In this paper, our method extends competitive multi-agent UED to a zero-shot human-AI coordination setting by using an appropriate utility function.

B. ZERO-SHOT HUMAN-AI COORDINATION

In multi-agent cooperation tasks, achieving generalization to partner agents is an important problem. Zero-shot coordination [22] and ad hoc team cooperation [23] aim to address this problem. These approaches commonly rely on self-play methods in which an agent learns through collaboration with its clone. However, self-play methods suffer from limitations. Because they assume partner agents to be optimal or similar to themselves during training, their performance is poorly coordinated with unseen partners such as humans or other agents [24].

To address these issues, [25] increases partner diversity and encourages self-play agents to learn different strategies. One of these approaches is Population-Based Training (PBT), which leverages diversity within a population of partner agents. In each iteration, agents are paired with different partners in the population, allowing them to learn various strategies simultaneously. Fictitious Co-Play (FCP) [8] is a well-known PBT-based approach. It uses the population of past checkpoint self-play agents as partners to train the ego-agent. Trajectory diversity (TrajeDi) [9] improved performance in cooperative tasks by generating diverse policies within the population. It used the Jensen-Shannon divergence to measure the diversity between different policies. Similarly, MACOP [26] focused on expanding partner diversity by continually generating a wide range of incompatible teammates through evolutionary processes. Furthermore, MEP [10] was proposed to mitigate potential issues arising from changes in the distribution of self-play agents during interactions with unseen partners. Recent approaches have further advanced zero-shot coordination. E3T [27] introduces an efficient end-to-end training framework using a mixture partner policy to eliminate the need for pre-trained populations. CoMeDi [28] leverages cross-play to encourage greater diversity than traditional statistical approaches and achieve generalized cooperation.

In addition to these diversity-driven approaches, recent work has explored the use of large language models (LLMs) for human-AI coordination. HAPLAN [29] improves coordination by establishing preparatory conventions using LLM before interaction. HLA [30] employs a hierarchical framework that combines high-level reasoning with low-level control for real-time coordination.

However, most of these approaches focus on improving coordination with unseen partners through partner diversity and do not explicitly address the challenges posed by unseen environments. In this study, we extend the focus to coordination across diverse unseen environments by constructing a population of co-player agents trained in various settings. Through this approach, we seek to address the poor coordination between AI agents and human partners in unseen environments.

III. AUTOMATIC CURRICULUM DESIGN FOR ZERO-SHOT HUMAN-AI COORDINATION

A. OVERVIEW

In this section, we describe our proposed method, Automatic Curriculum Design for Zero-Shot Human-AI Coordination, which is an auto-curriculum that trains the ego-agent using return-based utility functions. Our method focuses on training

Algorithm 1 Automatic Curriculum Design for Zero-Shot Human-AI Coordination

Input: Training environments Λ_{train}
Initialise: Ego policy π, Co-player population $\mathfrak{B}$
Initialise: Co-player Env buffers $\forall \pi' \in \mathfrak{B}$, $\mathbf{\Lambda}(\pi') := \emptyset$

for $i = \{1, 2, \dots\}$ **do**
  for N episodes **do**
    $\pi' \sim \mathfrak{B}$ ▷ Sample co-player via Eq.2
    Sample *Replay-decision*
    if replaying **then**
      $\theta \sim \mathbf{\Lambda}(\pi')$ ▷ Sample a replay env with dist.1
      Collect trajectory τ of π using (θ, π')
      Update π with rewards $\boldsymbol{R}$ (τ)
    else
      $\theta \sim \Lambda_{unseen/train}$ ▷ Sample unseen random env
      Collect trajectory τ of π using (θ, π')
    end if
    Compute Coordination score $S = Return(\theta, \pi')$
    Update $\mathbf{\Lambda}(\pi')$ with θ using score S
  end for
  $\mathfrak{B} \leftarrow \mathfrak{B} \cup \{\pi_i^{\perp}\}$, $\mathbf{\Lambda}(\pi_i^{\perp}) := \emptyset$ ▷ frozen weights
end for

the ego-agent to coordinate well with human partners in diverse environments. It is based on the MAESTRO algorithm [16], which was developed for multi-agent UED. Our method extends MAESTRO to human-AI coordination settings. It curates environment/co-player pairs with learning potential for the ego-agent in human-AI coordination.

The training procedure of our proposed method is shown in Algorithm 1. It begins by sampling the co-player agent π' that coordinates with the ego-agent from the co-player population during training. This sampling is based on coordination performance (return), as described in Section III-B. In the first epoch, the ego-agent coordinates with itself using the self-play method without sampling the co-player.

After co-player sampling, our method employs a *Replay-decision* based on a Bernoulli distribution. This decision process determines whether to sample an environment θ from the co-player's environment buffer or randomly sample an unseen environment θ from the finite set of training environments not yet stored in the co-player's environment buffer. When an environment is sampled from the co-player's environment buffer, our method follows the replay distribution given by Equation (1), which is based on the return. It then collects trajectories from the ego-agent in the sampled environment/co-player pairs. The ego-agent policy is updated whenever an environment is sampled from the co-player's environment buffer. Coordination scores are calculated using the collected trajectories and updated in the individual environment buffer of the sampled co-player accordingly. After N episodes of training, the ego agent's frozen weight policy is added to the co-player population $\mathfrak{B}$ with the environment buffer.

$$P_{\text{replay}} = (1 - \rho) \cdot P_S + \rho \cdot P_C \tag{1}$$

The replay distribution is a combination of two distributions: (P_S), based on the learning potential score, which is measured using the return. (P_C), based on how long each environment was sampled from the environment buffer. This equation follows the previous replay mechanism PLR [14]. The staleness coefficient $\rho \in [0, 1]$ is a weighting hyperparameter that balances (P_S) and (P_C). For more details on P_S, see Section III-C.

B. PRIORITIZED CO-PLAYER SAMPLING

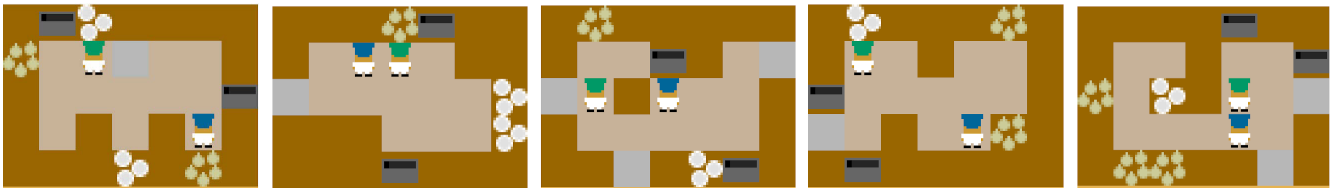
The multi-agent UED study for two-player zero-sum games [16] prioritizes sampling the co-player with the highest regret environment in its buffer and focuses on minimizing regret to obtain a minimax regret policy.

However, this approach is hard to apply directly to the human-AI coordination setting (common pay-off game). The ego-agent needs to learn a policy that not only minimizes regret but also considers the reward from coordination interaction, since the ego and co-player agent receive the common reward together. We need to set the co-player sampling metric by considering coordination for human-AI coordination. Most human-AI coordination studies [10], [11] use return as a co-player sampling metric. They sample co-players who achieve low returns when coordinating with the ego-agent, which indicates challenging coordination scenarios. This approach optimizes the lower bound of coordination performance with any co-player in the population, ensuring that the ego-agent can coordinate well with various co-players and human players.

Unlike previous human-AI coordination approaches that sample co-players based on low returns in a static environment, our method jointly considers both the co-player and the environments they have encountered during training. Equation (2) formalizes this strategy by sampling the co-player whose buffer contains the lowest-return environment in the co-player population. Each co-player environment buffer has a fixed size of k and stores environments based on the lowest return. It combines two existing ideas: (1) sampling co-players with low returns to improve worst-case coordination [10], [11], and (2) prioritizing high-potential learning experiences from environment buffers [14].

$$\text{Co-Player} \in \arg\min_{\pi' \in \mathfrak{B}} \left\{ \min_{\theta \in \mathbf{\Lambda}(\pi')} Return(\theta, \pi') \right\} \tag{2}$$

Here, $\mathfrak{B}$ is the co-player population, consisting of ego-agent policies that are periodically frozen during training, $\mathbf{\Lambda}(\pi')$ is the environment buffer of the co-player agent π', and $Return(\theta, \pi')$ is the return of the environment/co-player pairs (θ, π').



(a) $test_0$ (b) $test_1$ (c) $test_2$ (d) $test_3$ (e) $test_4$

FIGURE 2. Evaluation Layouts. The evaluation layouts were selected based on a previous study [31]. These layouts are selected to cover a range of difficulties, from easy ($test_0$) to difficult ($test_4$). Easier layouts allow agents to work independently within separate areas, whereas harder layouts require coordination with other agents without collisions.

C. LEARNING POTENTIAL SCORE BASED ON RETURN

Our method follows the replay distribution (Equation 1) when selecting an environment for the ego-agent from an individual co-player buffer. In this replay distribution, P_S is the first priority component that assigns higher priority to environments with learning potential for training the ego-agent.

As discussed in Section III-B, return is the appropriate scoring metric to build a prioritization distribution in human-AI coordination. It represents the coordination performance during the episode. Previous works [10], [11] have demonstrated that return captures coordination performance and also indicates which environments require more training. If the replayed environment has a low return, it indicates that the ego-agent is hard to coordinate in that environment. The details of S are as follows:

$$S_i = \frac{1}{T}\sum_{t=0}^{T} \text{Return}_t(\theta_i, \pi') \tag{3}$$

$$P_S(\theta_i|S) = \frac{\text{rank}\,(S_i)^{\frac{1}{\beta}}}{\sum_j \text{rank}\,(S_j)^{\frac{1}{\beta}}} \tag{4}$$

Equation (3) is the expected return of the replayed environment with the sampled co-player during the episode. Here, T represents the total number of timesteps in the episode, $t \in \{0, \ldots, T\}$ denotes each timestep within the episode, and i denotes the index of the replayed environment from the co-player's buffer. This environment score is normalized using Equation (4). This prioritization approach adapts the temperature-controlled ranking of previous work [14] to our coordination setting. We normalize the scores using a rank-based prioritization, which sorts the scores in the co-player buffer in ascending order, prioritizing low-return environments. j denotes the index of environments in the co-player's buffer. $\beta \in [0, 1]$ is the temperature parameter that adjusts the influence of the distribution.

IV. EXPERIMENTS SETUP

Overcooked AI [24] is a cooperative game played in a grid-world environment inspired by the kitchen. The goal is to cook as much food as possible in a limited time. Agents can get rewards by completing tasks sequentially, such as preparing ingredients, cooking, and serving dishes. Cooperation is essential in this environment, where success depends on effective coordination between players and their partners.

A. BASELINE MODEL

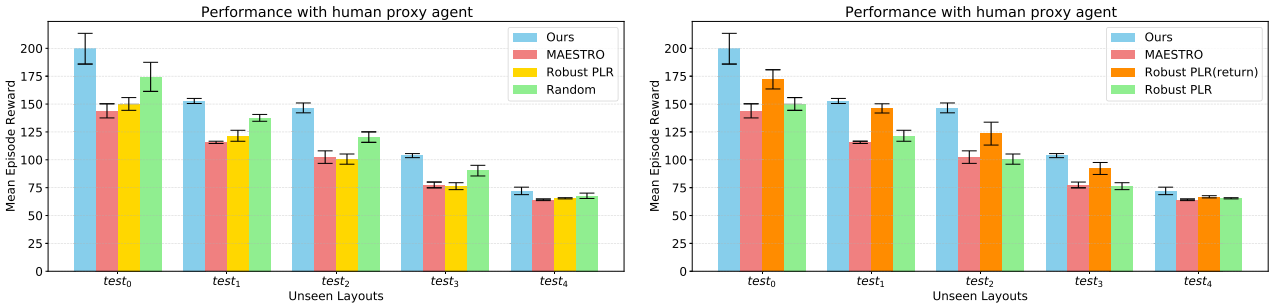
We compare our method with three key baseline models related to UED: MAESTRO [16], Robust PLR [14], and Domain Randomization [17]. Robust PLR and domain randomization are methods originally used in a single-agent setting, so we modified them for multi-agent settings using the self-play method. We used the PPO method [32] to train the ego-agent. PPO is the standard algorithm for training ego-agents through population-based learning with self-play in zero-shot human-AI coordination research. Previous studies [8] have demonstrated that this approach enables agents to coordinate with humans without relying on human data. The model architecture and hyperparameter choices are provided in Appendix A.

B. LAYOUT GENERATION

In our experiments, we automatically generated a total of 6,000 different layouts for Overcooked-AI, each layout was 7×5. These layouts were generated using a heuristic-driven layout generator capable of adjusting three layout parameters: the number of interactive blocks (such as pots, plates, outlets, and onions), the number of spaces, and the size of the layouts, which is fixed in this study. The generator was designed to ensure that each layout included at least 12 spaces and 5 interactive blocks. To avoid redundancy, the layout generator compared a layout array during generation. In addition, we check the solvability of the layout using the A* path-finding algorithm [33], which can find a path for the agent to access any interactive block. Further details on the algorithm are provided in the Appendix B.

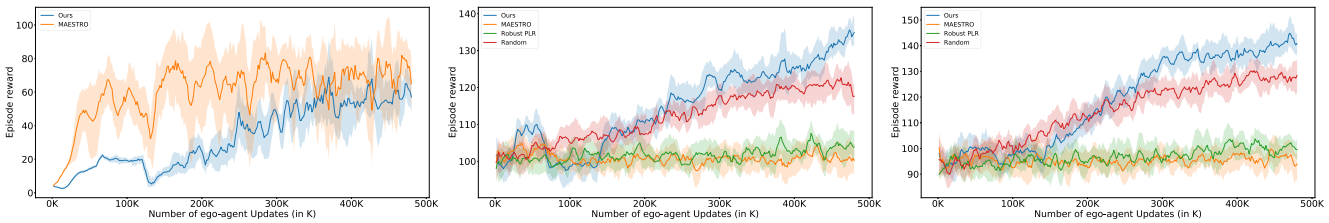
C. HUMAN PROXY

To evaluate the coordination performance of the trained agent in our experiments, we used a human proxy agent from a previous study [31] as a co-player agent. This human proxy agent performs as human-like as possible by adjusting its parameters based on human data using cross-entropy methods.



(a) **Overall Results** (b) **Utility Function Analysis**

FIGURE 3. (a) Overall results with a human proxy model. Our method achieves higher coordination performance with the human proxy agent than other baselines on all five evaluation layouts (Mean and standard error were computed over 100 runs with three random seeds). (b) Utility function analysis (*Regret* vs *Return*). Using a return-based utility function has an advantage over a regret-based utility function in training an ego-agent for the regret-based utility function for multi-agent coordination settings (Mean and standard error were computed over 100 runs with three random seeds).



(a) **Training** (b) **Training-evaluation** (c) **Evaluation**

FIGURE 4. Learning curves for the training, training-evaluation, and evaluation phases. The solid line shows the average episode reward across the three different random seeds, while the shaded areas show the maximum and minimum rewards. The results show that our model achieves higher performance on both the training and evaluation layouts.

D. EVALUATION

We evaluated the human-AI zero-shot coordination performance on five unseen layouts (Figure 2), selected based on coordination difficulty. Specifically, the five layouts correspond to the 100th, 75th, 50th, 25th, and 0th percentiles of the maximum achievable self-play rewards, as in a previous study [31]. This helps to measure the adaptability of the ego-agent based on the difficulty of the layout. In addition, we conducted a utility function analysis.

V. EXPERIMENTS RESULT

A. EXPERIMENTS WITH HUMAN PROXY

1) OVERALL RESULT

Figure 3(a) shows the comparison result of our method and the baseline models (MAESTRO, Robust PLR, and Domain Randomization) in terms of coordination performance with a human proxy agent on unseen layouts. The bar graph shows the evaluation results, which are the average episode rewards (three random seeds) with the human proxy agent on the evaluation layouts. As a result, our method showed better performance in all five evaluation layouts compared to other baselines. This performance gap is statistically significant across all layouts except $test_4$, as shown by the results of the paired t-test in Appendix E (see Table 5). We observe that our method achieves a higher generalization coordination performance with unseen partners on unseen evaluation layouts. We also performed cross-play experiments, where our method outperformed other baselines in unseen evaluation layouts. The results of these experiments are provided in the Appendix C.

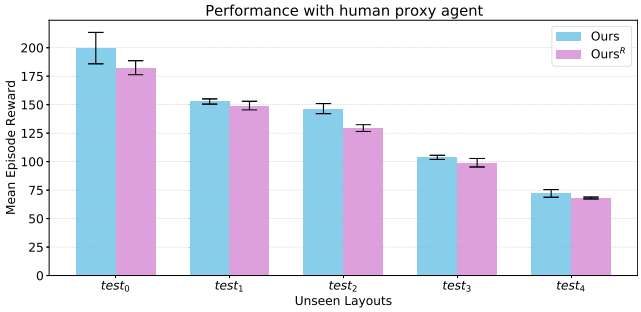
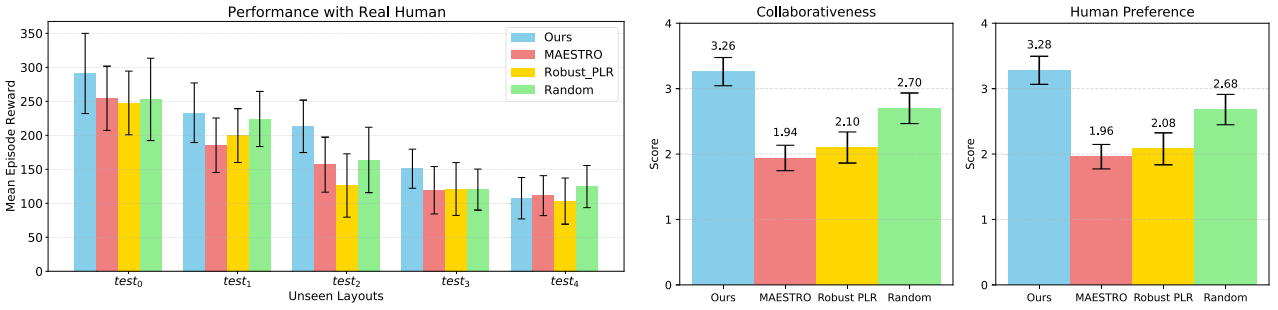


FIGURE 5. Results of the prioritized co-player sampling Prioritized co-player sampling leads to higher coordination performance with the human proxy agent compared to random co-player sampling across all five evaluation layouts. (Mean and standard error were computed over 100 runs with three random seeds.) Ours^R represents our proposed method with random co-player sampling.

B. STUDY OF THE UTILITY FUNCTION

1) EVALUATION RESULT

We conducted an analysis to try to figure out which utility functions are appropriate for a human-AI coordination setting. Figure 3(b) shows how the utility function affects the training of the ego-agent. Our method and Robust PLR (return) outperformed MAESTRO (regret-based baseline)



(a) **Human Study Result.** (b) **Human Survey Result.**

FIGURE 6. **This plot shows the results of an experiment with 20 real humans: (a) The average reward obtained on the five unseen evaluation layouts, (b) The average evaluation scores for the two items "Collaborativeness" and "Human Preference". Each subject was scored on a scale of 1 to 4 based on their voting ranking. A higher score indicates a more positive evaluation.**

and Robust PLR in all five evaluation layouts with the human proxy agent, demonstrating the effectiveness of return over regret as a utility function.

In this experiment, return is a more appropriate utility function than regret in human-AI coordination settings. The regret utility function is not suitable for curating challenging environments for coordination tasks. Regret quantifies the opportunity cost of not making the optimal decision without considering coordination. Therefore, it is difficult to assess the effectiveness of coordination performance using regret. However, return is appropriate for evaluating coordination performance because it is derived from the common rewards that agents receive through their collaborative efforts to achieve a common goal.

2) LEARNING CURVE

Figure 4(a) shows the learning curves of our method and MAESTRO during training. By comparing these learning curves, we evaluate which utility function is more effective for training the ego-agent in human-AI coordination. MAESTRO shows high episode reward in the early training stage due to overfitting easy difficulty environments, but shows high fluctuation when it encounters relatively less seen environments. On the other hand, our method steadily increases episode rewards by sampling more difficult environments according to the agent's coordination ability. Appendix D shows how the buffer composition of each method changes as the training progresses. Figure 4(b) illustrates the coordination performance of the ego-agent with the human proxy agent on randomly selected layouts from the training set at each training step. Meanwhile, Figure 4(c) shows the evaluation graph of the coordination performance of the ego agent with the human proxy agent in the evaluation layouts. The use of return as a utility function leads to an increase in episode rewards, whereas regret does not. This suggests that the ego-agent trained with a return-based approach achieves more effective coordination with the human proxy agent.

C. STUDY OF THE CO-PLAYER SAMPLING

1) EVALUATION RESULT

To evaluate the effect of prioritized co-player sampling, we compared our proposed method with random co-player sampling. Figure 5 shows that prioritized co-player sampling improves the training performance of the ego-agent. Our sampling strategy prioritizes co-players with low returns and jointly considers the challenging environments they have encountered. As a result, the ego-agent optimizes the lower bound of coordination performance. In contrast, random co-player sampling does not account for the varying difficulty levels of co-players or environments. This comparison highlights the importance of curating challenging coordination scenarios for the ego-agent.

D. HUMAN-AI COORDINATION

1) HUMAN STUDY RESULT

Additionally, we conducted a human study to evaluate coordination with real humans. We recruited 20 participants from the local student community and evaluated the proposed method and the baseline models with our participants. All participants provided their informed consent before the experiment, which was conducted following ethical guidelines approved by the Institutional Review Board (IRB).

Figure 6 shows the average coordination performance of five evaluation layouts in the human study. Our method outperformed the baseline models on all layouts ($test_0$, $test_1$, $test_2$, $test_3$), but showed a similar performance to other baselines in $test_4$, with no statistically significant differences ($p > 0.05$) according to a t-test. This performance is a limitation of our approach. Our method is designed to improve the lower bound of coordination performance by prioritizing training with difficult environment/co-player pairs, which enhances robustness in most scenarios. However, this strategy may bias the model towards specific coordination patterns, potentially limiting generalization in extremely complex environments like $test_4$. The sophisticated coordination requirements of $test_4$ may fall outside the

range of patterns for which our method has been optimized. However, consistent performance improvements in $test_0$ through $test_3$ support the effectiveness of our method. The detailed results of the paired t-tests for all layouts are provided in Appendix E (see Table 6).

2) HUMAN SURVEY RESULT

For each session of the experiment, the participants ranked the models in terms of collaborativeness (the most collaborative partner) and human preference (their top preference). The human evaluation scores in Figure 6 (b) were converted to a scale of 1 to 4 based on each participant's subjective ranking of the models. A higher score indicates a better evaluation. The plot shows that our method has higher collaborativeness and human preference compared to the baseline models. Participants rated the agent trained with our method as more cooperative and preferred it for coordination over baseline models. Collaborativeness and human preference scores for each layout are provided in the appendix D.

VI. CONCLUSION

A. SUMMARY

In this paper, we propose Automatic Curriculum Design for Zero-Shot Human-AI coordination. Our method extends the multi-agent Unsupervised Environment Design (UED) approach to zero-shot human-AI coordination, which trains the ego-agent to coordinate with human partners in unseen environments. We use return as a utility function, unlike previous multi-agent UED methods that use regret. Our method outperformed other baselines when evaluated with a human proxy in Overcooked-AI. Ablation studies demonstrate the effectiveness of the proposed components. Finally, we had a real human experiment to evaluate the human-AI coordination performance of our method. The result of the human-AI coordination experiment shows that our method outperforms other baselines.

B. LIMITATION AND FUTURE WORK

When using the return as a measure of the learning potential of the environment/co-player pair, the evaluation performance improved compared to a previous method. However, prioritizing environments with low returns leads to the issue of sampling environment/co-player pairs that may be too challenging for the agent to learn effectively. In future work, we plan to adopt an alternative metric based on success probability [20] to promote the sampling of pairs of learnable environment-co-players for human-AI coordination. The success of coordination can be evaluated through indicators such as collision-free movements and cooperative completion of cooking tasks. We will also use mutation to adjust environments that are sampled repeatedly to prevent overfitting to specific environments [15], [34].

In the current multi-agent UED, the co-player pool is constructed from the past frozen weights of trained ego-agents. This approach mitigates the non-stationarity problem compared to self-play approaches, in which the co-player policy is changed during training. However, since it essentially plays with past versions of itself, it struggles to train diverse strategies. We plan to investigate methods for increasing the diversity of the co-player pool in UED.

APPENDIX A
IMPLEMENTATION DETAILS AND HYPERPARAMETERS

Our implementation is based on PLR [14] and Overcooked-AI [24]. In all experiments, we train the ego-agent using the PPO method [32]. The details of the network architecture and hyperparameter choices are provided in Tables 1 and Table 2, respectively. Additionally, Table 3 presents the final hyperparameter choices for all methods.

TABLE 1. Network architecture details.

Layers	Channels	Kernel Size
Conv2D + LeakyReLU	25	5×5
Conv2D + LeakyReLU	25	3×3
Conv2D + LeakyReLU	25	3×3
Flatten	/	/
(FC + LeakyReLU)×3	64	/
FC	6	/

TABLE 2. PPO agent hyperparameters.

Parameter	Value
PPO	
λ_{GAE}	0.98
γ	0.99
Number of epochs	8
Rollout Length	400
PPO clip range	0.05
RMSprop optimizer (ϵ)	0.00001
Learning rate	0.001
Value loss coefficient	0.1
Entropy Coefficient	0.1
Number of mini-batches	20
Minibatch size	5000

Overcooked-AI: To evaluate the improvement in zero-shot human AI coordination performance in unseen environments, we experiment in the Overcooked AI environment, where players can cook and serve food. The environment has interactive blocks such as onions, plates, pots, and outlets, wall blocks that agents cannot move through, and space blocks that agents can move through. To get a reward from the environment, each player picks up 3 onions (3 reward) and puts them in the pot (3 reward). After 20 seconds, the onion soup is ready. Take a plate to serve the onion soup (5 reward). They then submit the dish to the outlets. Finally, all players get a common reward of 20. This reward shaping is based on a previous Overcooked-AI benchmark paper [24].

APPENDIX B
LAYOUT GENERATOR

The algorithm 2 shows the overall layout generation process. A layout generator is a heuristic-driven tool whose

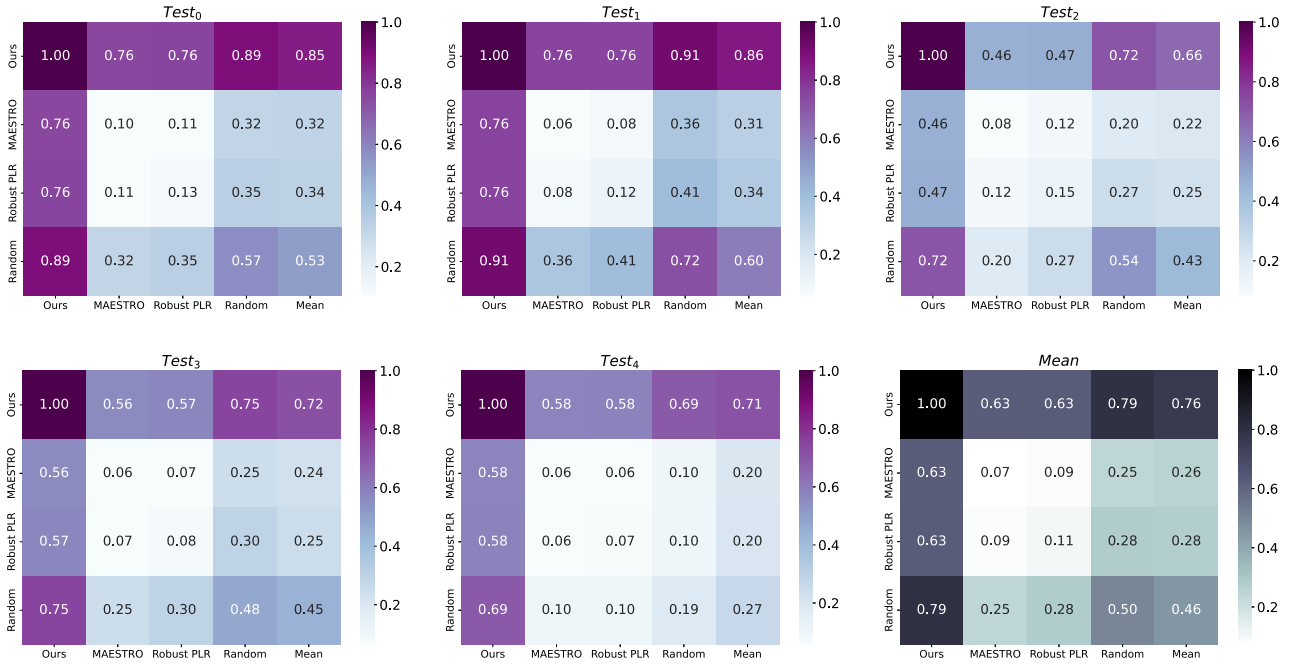


FIGURE 7. AI-AI Cross-play result This result is the normalized value of the reward obtained by cross-playing the trained model on five evaluation layouts and their average.

TABLE 3. Training hyperparameters for each method.

Parameter	Value
Robust PLR	
Replay rate, p	0.5
Buffer size, K	4,000
Scoring function	Positive value loss
Prioritization	rank
Temperature, β	0.3
Staleness coefficient, ρ	0.3
MAESTRO	
λ coef	0.2
Buffer size of co-player, K	1,000
Scoring function	Positive value loss
Prioritization	rank
Population size, i	8
Episodes N	375
Our	
λ coef	0.2
Buffer size of co-player, K	1,000
Scoring function	return
Prioritization	reverse-rank
Population size, i	8
Episodes N	375

input consists of three parameters: The maximum size of the layout buffer M, the number of interactive blocks N, and the number of empty spaces E. The generator ensures the presence of a minimum of E spaces and N interactive blocks within each map. The locations of the blocks in the layout are randomly selected. In this experiment, we set the parameters as $M = 6,000$, $N = 6 \sim 9$, and $E = 14$. During the layout generation process, the generator checks the

Algorithm 2 Layout Generation Process

Input: Maximum size of layout buffer M, Number of interactive blocks N, Number of empty space E
Initialise: Layout buffer L

```
while Size of layout buffer < M do
    Make random 7 × 5 array as a Layout
    Randomly place N blocks
    if Duplicate(Layout, L) then
        Continue
    end if
    Place two players in a random position
    Remove non-reachable blocks
    if Solvability(Layout) then
        Append Layout to L
    end if
end while
```

Hamming distance between the embedded arrays to eliminate any redundancy. Additionally, we verify the solvability of each layout using the A* path-finding algorithm [33], which determines whether the agent can reach each interactive block.

APPENDIX C
CROSS-PLAY RESULTS

Figure 7 represents the detailed results of experiments with AI-AI cross-play on the five evaluation layouts. This is a zero-shot coordination experiment in which different models are coordinated without direct encounters with others during

the training phase. These scores are normalized relative to the highest reward obtained from that layout, with the maximum value set at 1 and the minimum value set to 0. Our method performs better than the baselines in all five evaluation layouts. The final mean graph shows the average value of the normalized scores for the five layouts. Our method also outperforms baselines in self-play where each method plays itself, and in cross-play where each method plays against different methods.

Robust PLR and MAESTRO use a regret-based utility function, which is not suitable for cooperative environments, resulting in worse performance than the random method. In contrast, our method improves the lower bound of the performance by using a return-based utility function. This approach curates challenging environment/co-player pairs for the ego-agent. Our trained ego-agent generally performs well in easy-to-hard environments and is robust to unseen environments and co-players.

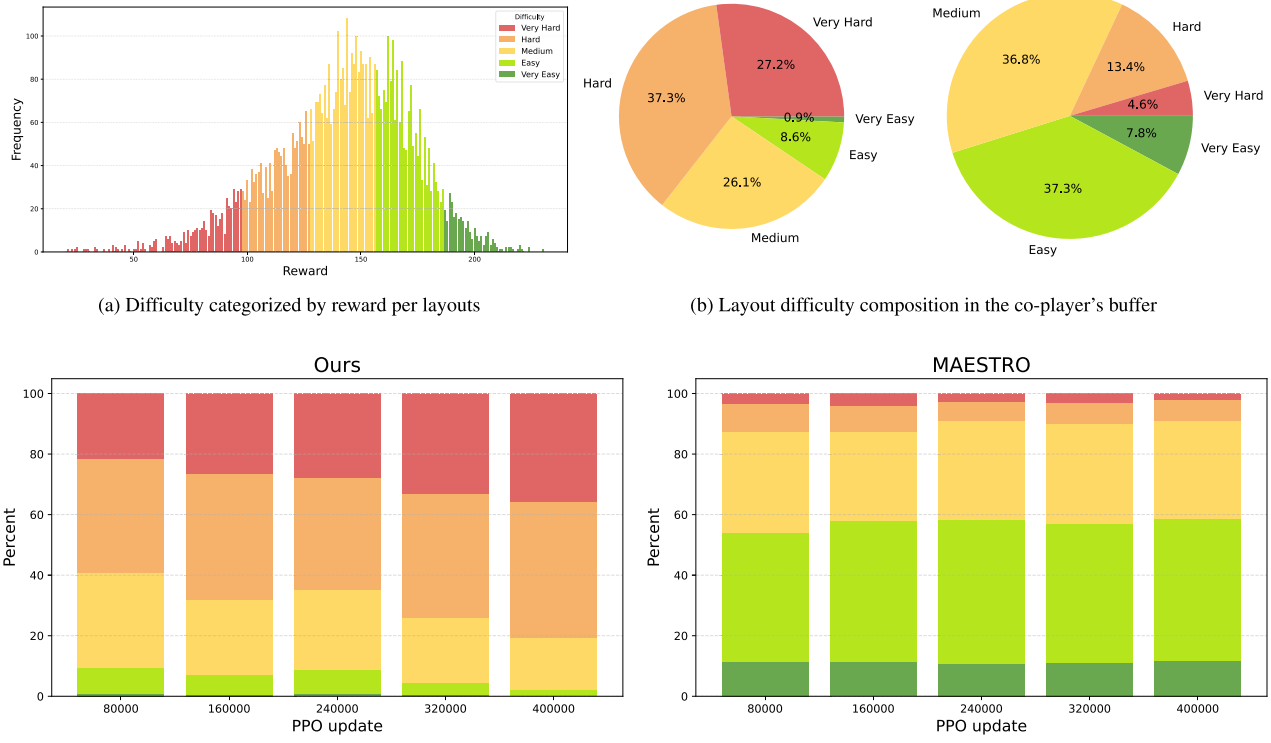
APPENDIX D
ADDITIONAL RESULTS

Define Training Layouts Difficulty: To categorize the difficulty of the layouts, we obtained the average reward (50 runs) evaluated from the trained agent and the human proxy agent. Figure 8(a) shows the distribution of average rewards obtained from the 6,000 layouts used for training. We applied the criteria outlined in Table 4, categorizing the difficulty levels into five groups, from 'Very Easy' to 'Very Hard', based on the mean and standard deviation of the reward distribution. According to this classification scheme, the layouts classified as 'Medium' are the most numerous, while those classified as 'Very Hard' or 'Very Easy' are comparatively rare.

TABLE 4. Difficulty classification based on reward conditions.

Difficulty	Reward Condition
Very Easy	$reward > \mu + 1.5\sigma$
Easy	$\mu + 0.5\sigma < reward \leq \mu + 1.5\sigma$
Medium	$\mu - 0.5\sigma < reward \leq \mu + 0.5\sigma$
Hard	$\mu - 1.5\sigma < reward \leq \mu - 0.5\sigma$
Very Hard	$reward \leq \mu - 1.5\sigma$

Co-player's Buffer Composition: Figure 8(b) shows the cumulative composition of layout difficulties in the environment buffer of the co-player, which is first added to the co-player population during the training process. Both our method and MAESTRO prioritize layout(environment) sampling and store layouts in the co-player's buffer of size 1,000 according to their own criteria. In our method, the



(a) Difficulty categorized by reward per layouts

(b) Layout difficulty composition in the co-player's buffer

(c) Changing the layout difficulty composition of buffers

FIGURE 8. (a) Distribution of rewards obtained by the trained model on training layouts, playing with a human proxy. Difficulty is categorized based on these reward distributions. (b) Difficulty distribution of layouts sampled during the training process for our method and MAESTRO. (c) The difficulty configuration of the buffer changes as training progresses.

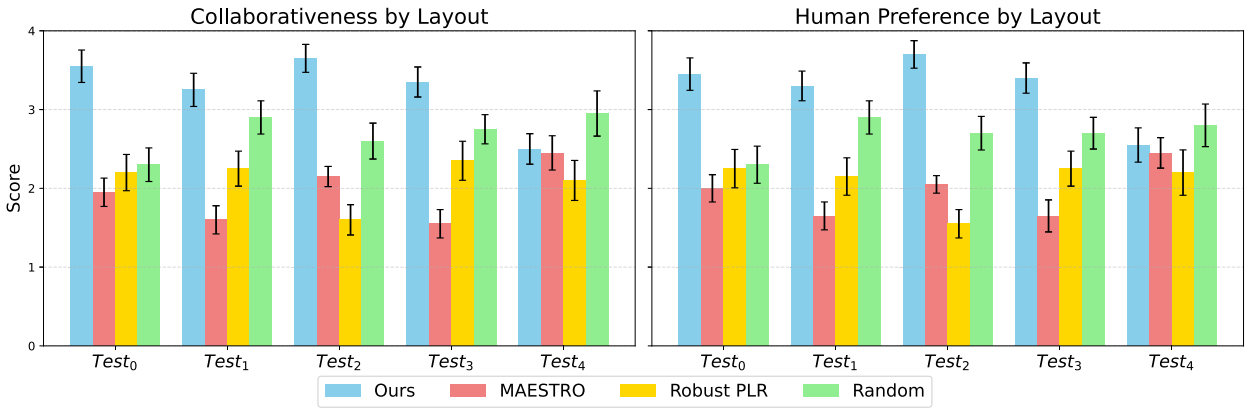


FIGURE 9. **Human survey results by test layouts Results from a survey of 20 human players on five evaluation layouts. In the layouts of $test_0$, $test_1$, $test_2$ and $test_3$, human players tended to prefer our model, but in the hardest layout of $test_4$, the human player had different preferences, and each model scored similarly.**

percentage of 'Very Hard' layouts in the co-player's buffer increases as the training progresses from start to finish. This indicates that our method samples the more difficult layouts as the agent's performance improves, similar to curriculum learning. However, MAESTRO was likely to sample easier environments as training progressed, so the percentage of 'Easy' layouts in the co-player buffer increased. Figure 8(c) represents details of the changing layout composition in the co-player's buffer. These results support the idea that the return-based utility function is more appropriate than the regret-based utility function to assess learning potential in a coordination setting.

APPENDIX E
ADDITIONAL PAIRED T-TEST

See Tables 5 and 6.

TABLE 5. **Paired t-test results between Ours and baseline methods on each evaluation layout with a human proxy.**

Layout	Comparison	t-statistic	p-value
$test_0$	MAESTRO	4.8088	0.0406*
	Robust PLR	4.4569	0.0468*
	Random	4.9371	0.0387*
$test_1$	MAESTRO	29.0707	0.0012**
	Robust PLR	9.4667	0.0110*
	Random	10.1263	0.0096**
$test_2$	MAESTRO	7.6474	0.0167*
	Robust PLR	207.0434	<0.0001***
	Random	25.8937	0.0015**
$test_3$	MAESTRO	17.2566	0.0033**
	Robust PLR	10.8919	0.0083**
	Random	4.1389	0.0537
$test_4$	MAESTRO	3.5168	0.0722
	Robust PLR	4.0035	0.0571
	Random	1.7264	0.2264

TABLE 6. **Paired t-test results between Ours and baseline methods on each layout evaluated with a human partner.**

Layout	Method	t-statistic	p-value
$test_0$	MAESTRO	3.2151	0.0046**
	Robust_PLR	3.6306	0.0018**
	Random	2.9939	0.0075**
$test_1$	MAESTRO	6.1094	<0.0001***
	Robust_PLR	3.3245	0.0036**
	Random	0.9126	0.3729
$test_2$	MAESTRO	4.8766	0.0001***
	Robust_PLR	5.6865	<0.0001***
	Random	3.9149	0.0009***
$test_3$	MAESTRO	4.3184	0.0004***
	Robust_PLR	3.3836	0.0031**
	Random	4.2080	0.0005***
$test_4$	MAESTRO	-0.4229	0.6771
	Robust_PLR	0.5977	0.5571
	Random	-1.6296	0.1197

REFERENCES

[1] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel, T. Lillicrap, K. Simonyan, and D. Hassabis, "Mastering chess and shogi by self-play with a general reinforcement learning algorithm," 2017, *arXiv:1712.01815*.

[2] C. Berner et al., "Dota 2 with large scale deep reinforcement learning," 2019, *arXiv:1912.06680*.

[3] O. Vinyals, I. Babuschkin, J. Chung, M. Mathieu, M. Jaderberg, W. M. Czarnecki, A. Dudzik, A. Huang, P. Georgiev, R. Powell, and T. Ewalds, "Alphastar: Mastering the real-time strategy game starcraft II," *DeepMind blog*, vol. 2, p. 20, Jan. 2019.

[4] B. R. Kiran, I. Sobh, V. Talpaert, P. Mannion, A. A. A. Sallab, S. Yogamani, and P. Pérez, "Deep reinforcement learning for autonomous driving: A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 6, pp. 4909–4926, Jun. 2022.

[5] M. Fang, T. Zhou, Y. Du, L. Han, and Z. Zhang, "Curriculum-guided hindsight experience replay," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 12602–12613.

[6] K. Zhang, Z. Yang, and T. Başar, "Multi-agent reinforcement learning: A selective overview of theories and algorithms," in *Handbook of Reinforcement Learning and Control*. Springer, 2021, pp. 321–384.

[7] A. Oroojlooy and D. Hajinezhad, "A review of cooperative multi-agent deep reinforcement learning," *Int. J. Speech Technol.*, vol. 53, no. 11, pp. 13677–13722, Jun. 2023.

[8] D. Strouse, K. R. McKee, M. Botvinick, E. Hughes, and R. Everett, "Collaborating with humans without human data," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, pp. 14502–14515.

[9] A. Lupu, B. Cui, H. Hu, and J. Foerster, "Trajectory diversity for zero-shot coordination," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 7204–7213.

[10] 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 *Proc. AAAI Conf. Artif. Intell.*, vol. 37, Jun. 2023, pp. 6145–6153.

[11] 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 *Proc. Int. Conf. Auto. Agents Multiagent Syst.*, 2023, pp. 679–688.

[12] M. D. Dennis, N. Jaques, E. Vinitsky, A. M. Bayen, S. Russell, A. Critch, and S. Levine, "Emergent complexity and zero-shot transfer via unsupervised environment design," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 13049–13061.

[13] M. Jiang, M. D. Dennis, J. Parker-Holder, J. Foerster, E. Grefenstette, and T. Rocktäschel, "Replay-guided adversarial environment design," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, pp. 1884–1897.

[14] M. Jiang, E. Grefenstette, and T. Rocktäschel, "Prioritized level replay," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 4940–4950.

[15] J. Parker-Holder, M. Jiang, M. J. Dennis, M. Samvelyan, J. Foerster, E. Grefenstette, and T. Rocktäschel, "Evolving curricula with regret-based environment design," in *Proc. Int. Conf. Mach. Learn.*, 2022, pp. 17473–17498.

[16] M. Samvelyan, A. Khan, M. J. Dennis, M. Jiang, J. Parker-Holder, J. Foerster, R. Raileanu, and T. Rocktäschel, "MAESTRO: Open-ended environment design for multi-agent reinforcement learning," in *Proc. 11th Int. Conf. Learn. Represent.*, 2023, pp. 1–18. [Online]. Available: https://openreview.net/forum?id=sKWlRDzPfd7

[17] N. Jakobi, "Evolutionary robotics and the radical envelope-of-noise hypothesis," *Adapt. Behav.*, vol. 6, no. 2, pp. 325–368, Sep. 1997.

[18] I. Gur, N. Jaques, Y. Miao, J. Choi, M. Tiwari, H. Lee, and A. Faust, "Environment generation for zero-shot compositional reinforcement learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2022, pp. 4157–4169.

[19] H.-S. Chung, J. Lee, M. S. Kim, D. Kim, and S. Oh, "Adversarial environment design via regret-guided diffusion models," in *Proc. 38th Annu. Conf. Neural Inf. Process. Syst.*, 2024, pp. 63715–63746.

[20] A. Rutherford, M. Beukman, T. Willi, B. Lacerda, N. Hawes, and J. N. Foerster, "No regrets: Investigating and improving regret approximations for curriculum discovery," in *Proc. 38th Annu. Conf. Neural Inf. Process. Syst.*, 2024, pp. 16071–16101.

[21] C. Ruhdorfer, M. Bortoletto, A. Penzkofer, and A. Bulling, "The overcooked generalisation challenge," 2024, *arXiv:2406.17949*.

[22] H. Hu, A. Lerer, A. Peysakhovich, and J. Foerster, "'Other-play' for zero-shot coordination," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 4399–4410.

[23] P. Stone, G. Kaminka, S. Kraus, and J. Rosenschein, "Ad hoc autonomous agent teams: Collaboration without pre-coordination," in *Proc. AAAI Conf. Artif. Intell.*, vol. 24, Jul. 2010, pp. 1504–1509.

[24] M. Carroll, R. Shah, M. K. Ho, T. L. Griffiths, S. A. Seshia, P. Abbeel, and A. D. Dragan, "On the utility of learning about humans for human-AI coordination," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 1–15.

[25] S. Devlin, D. Kudenko, and M. Grześ, "An empirical study of potential-based reward shaping and advice in complex, multi-agent systems," *Adv. Complex Syst.*, vol. 14, no. 2, pp. 251–278, Apr. 2011.

[26] L. Yuan, L. Li, Z. Zhang, F. Chen, T. Zhang, C. Guan, Y. Yu, and Z.-H. Zhou, "Learning to coordinate with anyone," in *Proc. 5th Int. Conf. Distrib. Artif. Intell.*, Nov. 2023, pp. 1–9.

[27] X. Yan, J. Guo, X. Lou, J. Wang, H. Zhang, and Y. Du, "An efficient end-to-end training approach for zero-shot human-ai coordination," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 36, 2023, pp. 2636–2658.

[28] B. Sarkar, A. Y. Shih, and D. Sadigh, "Diverse conventions for human-AI collaboration," in *Proc. Adv. Neural Inf. Process. Syst.*, 2023, pp. 23115–23139.

[29] 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," 2023, *arXiv:2311.00416*.

[30] J. Liu, C. Yu, J. Gao, Y. Xie, Q. Liao, Y. Wu, and Y. Wang, "LLM-powered hierarchical language agent for real-time human-AI coordination," in *Proc. 23rd Int. Conf. Auto. Agents Multiagent Syst.*, 2023, pp. 1219–1228.

[31] M. Yang, M. Carroll, and A. Dragan, "Optimal behavior prior: Data-efficient human models for improved human-AI collaboration," 2022, *arXiv:2211.01602*.

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

[33] P. Hart, N. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Trans. Syst. Sci. Cybern.*, vol. SSC-4, no. 2, pp. 100–107, Feb. 1968.

[34] R. Wang, J. Lehman, A. Rawal, J. Zhi, Y. Li, J. Clune, and K. O. Stanley, "Enhanced POET: Open-ended reinforcement learning through unbounded invention of learning challenges and their solutions," in *Proc. Int. Conf. Mach. Learn.*, vol. 1, 2020, pp. 9940–9951.

WON-SANG YOU received the B.S. degree in economy and statistics from Yonsei University, Wonju, South Korea, in 2019. He is currently pursuing the Ph.D. (M.S. and Ph.D. integrated program) degree with the Department of AI Convergence, Gwangju Institute of Science and Technology, Gwangju, South Korea. His research interests include human-AI coordination, reinforcement learning, and multi-agent.

TAE-GWAN HA received the B.S. degree in mechanical engineering from Gwangju Institute of Science and Technology, in 2021, and the M.S. degree from the School of Integrated Technology, Gwangju Institute of Science and Technology, in 2023. His research interests include reinforcement learning and game artificial intelligence.

SEO-YOUNG LEE received the B.S. degree in computer science and engineering from Dongguk University, Seoul, South Korea, in 2023. He is currently pursuing the M.S. degree with the Department of AI Convergence, Gwangju Institute of Science and Technology, Gwangju, South Korea. His research interests include meta-reinforcement learning and instructed reinforcement learning.

KYUNG-JOONG KIM (Member, IEEE) received the B.S., M.S., and Ph.D. degrees in computer science from Yonsei University, in 2000, 2002, and 2007, respectively. He was a Postdoctoral Researcher with the Department of Mechanical and Aerospace Engineering, Cornell University, in 2007. He is currently a Professor with the Department of AI Convergence, Gwangju Institute of Science and Technology (GIST). His research interests include artificial intelligence, game, and robotics.