Credit Assignment and Fine-Tuning Enhanced Reinforcement Learning for Collaborative Spatial Crowdsourcing

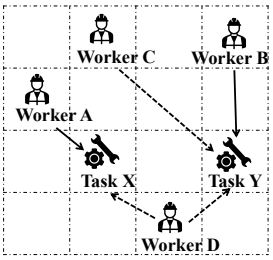
Wei Chen, **Yafei Li***, **Baolong Mei**, **Guanglei Zhu**, **Jiaqi Wu**, **Mingliang Xu**
School of Computer Science and Artificial Intelligence, Zhengzhou University
ischenwei@outlook.com, {ieyfli*, iexumingliang}@zzu.edu.cn, {blmeizzu, ieglzhu, jiaqiwu}@gs.zzu.edu.cn

Abstract

Collaborative spatial crowdsourcing leverages distributed workers' collective intelligence to accomplish spatial tasks. A central challenge is to efficiently assign suitable workers to collaborate on these tasks. Although mainstream reinforcement learning (RL) methods have proven effective in task allocation, they face two key obstacles: delayed reward feedback and non-stationary data distributions, both hindering optimal allocation and collaborative efficiency. To address these limitations, we propose CAFE (credit assignment and fine-tuning enhanced), a novel multi-agent RL framework for spatial crowdsourcing. CAFE introduces a credit assignment mechanism that distributes rewards based on workers' contributions and spatiotemporal constraints, coupled with bi-level meta-optimization to jointly optimize credit assignment and RL policy. To handle non-stationary spatial task distributions, CAFE employs an adaptive fine-tuning procedure that efficiently adjusts credit assignment parameters while preserving collaborative knowledge. Experiments on two real-world datasets validate the effectiveness of our framework, demonstrating superior performance in terms of task completion and equitable reward redistribution.

1 Introduction

With the popularity of smart mobile devices, spatial crowdsourcing (SC) has emerged as a promising computing paradigm [Li *et al.*, 2022c; Li *et al.*, 2022b], where workers perform spatial tasks assigned by the SC platform for payments [Wu *et al.*, 2024b; Mei *et al.*, 2024]. The increasing complexity and diverse requirements of spatial tasks have driven the evolution of a more sophisticated model within SC, namely *Collaborative Spatial Crowdsourcing* (CSC). In this paradigm, multiple workers can collaboratively complete tasks, leveraging a collective workforce to accelerate task completion. Such collaborative approaches are prevalent in various real-world applications, including home renovation, furniture assembly, and venue arrangement [Cheng *et al.*, 2016; Cheng *et al.*, 2019; Zhao *et al.*, 2024], etc.



Unmatched Workers

Worker order	Optional tasks
Worker C	Task Y
Worker D	Task X, Task Y

Matched Workers

Worker order	Assigned tasks
Worker A	Task X
Worker B	Task Y

Figure 1: An example of collaborative spatial crowdsourcing.

To illustrate CSC, consider the example shown in Fig.1, which involves four workers (A-D) and two tasks (X, Y). The initial configuration assigns workers A and B to tasks X and Y respectively, while workers C and D remain unallocated. The scenario is characterized by worker heterogeneity: worker C is exclusively qualified for task Y, while worker D possesses the versatility to perform either task. Task completion times directly impact payment structures, and the assignment of worker D to either task X or Y yields different compensation outcomes. This study aims to optimize the platform's revenue through strategic task allocation. Notably, as the scale of workers and tasks expands in real-world applications, the computational complexity of CSC increases exponentially.

Existing research on CSC has explored diverse approaches, ranging from heuristic algorithms [Cheng *et al.*, 2019; Zhao *et al.*, 2024] to learning-based methods [Zhao *et al.*, 2023]. Cheng *et al.* (2019) introduced a collaborative framework where multiple workers jointly handle space tasks to maximize overall collaboration quality. Zhao *et al.*(2024) developed an equilibrium-based approach that combines simulated annealing with Nash equilibrium refinement to optimize total rewards while maintaining priority-aware fairness in task assignments. Zhao *et al.* (2023) advanced the field by incorporating mutual information and attention mechanisms for group preference modeling, alongside tree decomposition and curriculum learning strategies to enhance task allocation efficiency. Building on this progress, Zhan *et al.*(2024) integrated graph neural networks to assess worker trustworthiness and implemented a specialized Tabu search algorithm for worker assignment.

*Corresponding authors.

Although multi-agent reinforcement learning (MARL) has been widely applied in task allocation problems, its application to CSC faces significant challenges due to delayed and sparse reward feedback. The credit assignment of reward in CSC is particularly complex due to the temporal nature of task completion, where payments are only generated upon task completion, yet workers join and contribute at different timestamps throughout the execution process. This temporal misalignment complicates reward allocation, as simple time-based distribution methods fail to capture the nuanced dynamics of worker participation. Furthermore, workers' heterogeneous contributions to the same task create additional complexity in fair reward distribution - certain workers may provide critical contributions despite shorter participation periods. For instance, strategically assigning a worker to a nearly-completed task rather than a newly published one can accelerate task completion and worker availability, creating a cascade effect that enhances overall platform efficiency. In such cases, although the worker's time investment may be brief, their strategic contribution significantly impacts the platform's revenue, warranting higher rewards despite shorter participation duration.

Another critical challenge in CSC stems from the inherent heterogeneity across tasks. As each task presents unique characteristics and requirements, the learning experiences derived from different tasks may contain significant noise and variability. This non-stationary data distribution makes it challenging to effectively learn knowledge across tasks, as parameters trained on one task may not generalize well to others due to task-specific patterns and biases. Such diversity in task characteristics creates substantial learning interference, potentially compromising the stability and robustness of the learned policies.

To address the aforementioned challenges, this paper presents CAFE (Credit Assignment and Fine-tuning Enhanced), a novel framework for MARL that incorporates two key components: (i) a credit attribution mechanism that accurately evaluates individual worker contributions, and (ii) a robust parameter fine-tuning approach that effectively mitigates task-level sample noise. The key contributions of our work are as follows:

- *A Causality Reward Redistribution Methods for MARL.* We propose a multi-agent reinforcement learning approach to address the CSC problem, and subsequently introduce a causal-perspective reward redistribution scheme that leverages Bayesian surprise to quantify individual agent contributions.
- *A Bi-Level Meta-Optimization Approach.* We propose a bi-level optimization framework leveraging implicit gradients, which concurrently optimizes RL objectives and reward redistribution parameters by strategically integrating historical learning trajectories.
- *Efficient Parameters Fine-Tuning for Diverse Data.* To address the non-stationary data distribution challenge in real-world datasets, we propose a parameters fine-tuning approach built upon implicit learning that enables rapid adaptation without deviating from the original optimization objective.
- *Extensive Empirical Studies.* We evaluate the effectiveness of our proposed approaches on real-world datasets. Extensive experiments demonstrate the effectiveness of our methods.

2 Related Works

2.1 Reinforcement Learning in Spatial Crowdsourcing

Reinforcement learning (RL) is increasingly applied in crowdsourcing to optimize task assignment and scheduling through adaptive policy learning in dynamic environments [Li *et al.*, 2025; Wu *et al.*, 2024a]. A common approach is to model task assignment as a Markov Decision Process. For instance, Zong *et al.* (2022) developed an end-to-end multi-agent system for pickup and delivery problems. In air-ground spatial crowdsourcing, Wang *et al.* (2023) introduced a multi-center attention-based graph convolutional network for communication. In contrast, Ye *et al.* (2023) proposed a heterogeneous multi-agent framework exploring both individual and collaborative environments. Jiang *et al.* (2023) designed a fairness-aware concurrent dispatch system for instant delivery services. For ride-hailing order dispatching, Zhang *et al.* proposed an offline deep reinforcement learning framework with a method for managing dynamic nondeterministic action spaces [Zhang *et al.*, 2024] Beyond direct application in decision-making, RL enhances established matching methods. It can determine the sliding window size of decision time steps [Li *et al.*, 2022c] and optimize existing algorithms based on game theory [Li *et al.*, 2023], large-scale search [Li *et al.*, 2022a], and combinatorial optimization [Tong *et al.*, 2021].

2.2 Credit Assignment in Reinforcement Learning

Delayed rewards present a fundamental challenge in reinforcement learning, wherein rewards manifest several timesteps after an action, making it difficult for agents to establish clear associations between actions and their consequent outcomes. Reward reshaping has emerged as a widely adopted approach to address this temporal credit assignment problem. Zhu *et al.* (2023) investigated episodic reinforcement learning with trajectory feedback, introducing an adaptive reward redistribution method grounded in bi-level optimization. In a complementary approach, Ren *et al.* (2021) developed a randomized return decomposition technique to transform long-horizon delayed reward problems into more manageable shorter sequences, thereby enabling effective training through mini-batch gradient descent. Advancing the field further, Ma *et al.* (2024) introduced a novel dual-agent framework comprising a policy agent for optimal behavior learning and a reward agent for generating auxiliary reward signals, thus achieving reward design without dependence on expert knowledge or hand-crafted functions. Qu *et al.* (2024) demonstrated that large models can be leveraged to encapsulate valuable decision-making knowledge, offering a promising avenue for reward redistribution. This approach of utilizing large models for processing state features has shown considerable promise in effectively addressing the temporal credit assignment challenge. In addition to credit assignment

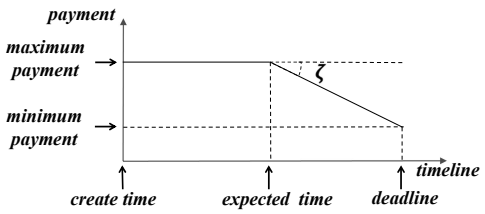


Figure 2: Payment of spatial task.

methods, many studies address the reward distribution problem from other perspectives, such as enhancing agent training in reinforcement learning by combining the advantages of group strategies and individual strategies [Wang *et al.*, 2022].

3 Problem Formulation

Generally, we define the CSC problem on a road network represented by a graph $G = (N, E)$, where $n \in N$ represents road intersections, and $e_{ij} \in E$ denotes roads connecting two intersections. Next, we define other entities related to the CSC problem.

Worker. A worker $w \in W$ is described by the tuple $w = (l, K)$, where l denotes the worker's location and K represents the set of skills. Furthermore, to simplify the problem, we assume that all workers have the same basic attributes, such as movement speed, work speed, etc.

Spatial Task. A spatial task $\tau \in \mathcal{T}$ is described by the tuple $\tau = (t^c, t^e, t^d, l, k, q, \rho)$. The task's temporal aspects are represented by t^c, t^e, and t^d, indicating the creation time, expected completion time, and deadline, respectively. Additionally, l is the location of the task, k represents the skill required to complete the task, each task has exactly one skill requirement, and q represents the task quantity, which is proportional to the workload completed by a worker per unit of time. Furthermore, ρ signifies the payment obtained upon successful completion of the task.

Considering the actual accomplishment of the task, the real benefit ρ of the task is expressed as follows:

$$\rho = \begin{cases} \rho_{max}, & \text{if } t^f \le t^e \\ [1 - \zeta \cdot (t^f - t^e)] \cdot \rho_{max}, & \text{if } t^e < t^f < t^d \\ 0, & \text{if } t^f \ge t^d \end{cases},$$

while t^f is the finish time of the task, $\rho_{max}$is the maximum payment that the task publisher can offer, ζ is the penalty rate, as shown in Fig.2.

CSC Problem. Given a stream of tasks $\mathcal{T}$ and a set of works W, CSC problem aims to find the optimal assignment $\mathcal{M}$ to maximize the global revenue P.

$$P = \max \sum_{\tau \in \mathcal{T}} \rho^{\tau},$$

4 Methodology

4.1 Markov Decision Process

We establish a multi-agent network system based on Markov Decision Process (MDP). This system is represented by the tuple $(\mathcal{N}, \mathcal{S}, \mathcal{O}, \mathcal{A}, \mathcal{R}, \mathcal{P}, \gamma)$, where $\mathcal{P}$ represents the state transition probability matrix and γ is the discount factor. Next, we elaborate on the meanings of other components in this MDP tuple.

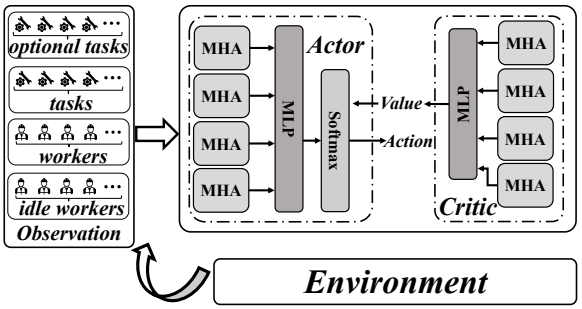


Figure 3: Actor-Critic network structure.

Agent $n \in \mathcal{N}$: Each worker w is modeled as an agent n, and all agents are assumed to share an identical network structure and parameters.

State $s \in \mathcal{S}$: The global state s_t is represented by a vector $(\mathcal{T}, W)$, which includes both worker and spatial task real-time information.

Observation $o \in \mathcal{O}$: The observation o_t^n is represented as a triplet $(\mathcal{T}_{near}, W_{near}, W_{idle}, \mathcal{T}_{optinal}, w_{self})$, which captures the conditions of nearby workers, surrounding tasks, idle workers, surrounding optional tasks, and the agent's own real-time information.

Action $a \in \mathcal{A}$: The action is defined as $\{\tau_0, \tau_1, \tau_2, \ldots, \tau_x\}$, where τ_0 represents the option of not selecting any task and τ_x represents the task index of the surrounding optional tasks.

Reward $r \in \mathcal{R}$: Reward serves as the critical factor in evaluating the quality of agent actions and directly influences the model's training outcomes. In the methodology section, we propose a novel reward redistribution approach.

4.2 Neural Network Architecture

Our proposed method, CAFE, builds upon the Actor-Critic framework by introducing a novel neural network architecture. This architecture features a base module that seamlessly integrates with established multi-agent actor-critic approaches such as MAPPO [Yu *et al.*, 2022], IPPO [De Witt *et al.*, 2020], and MADDPG [Lowe *et al.*, 2017] as shown in Fig.3. In the following sections, we present a detailed description of the key components and their interconnections within our proposed architecture.

Actor: The actor network employs four Multi-Head Attention (MHA) [Vaswani, 2017] layers to process the observation, followed by a Multi-Layer Perceptron (MLP) to integrate the outputs of the MHA layers, and finally uses a softmax function to output the action.

Critic: The Critic network has a similar architecture to the actor network, with the key difference being that the Critic network outputs the evaluation of the current state.

4.3 Reward Redistribution

To comprehensively evaluate how individual workers' actions affect overall revenue, we propose a reward redistribution

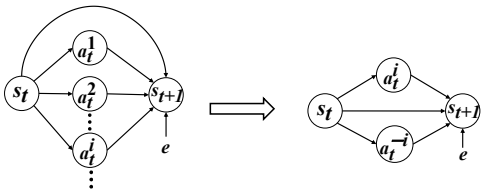


Figure 4: Structural causal model of MARL. Meanwhile, e represents the external disturbance affecting state transitions. For notational convenience, we denote by a_t^{-i} the joint actions of all agents except agent i at time step t.

mechanism that quantifies each worker's contribution.

To systematically analyze agent contributions in MARL, we first develop a structural causal model (SCM) that captures the underlying causal relationships of the process (Fig.4). Within this model, directional arrows represent inherent causal dependencies between components. To quantify how each agent's action a_t^i causally impacts the subsequent state s_{t+1}, we employ conditional mutual information (CMI):

$$CMI_t^i := I(s_{t+1}; a_t^i | s_t, a_t^{-i}). \tag{1}$$

To enhance computational efficiency while preserving the essential causal relationships, we make two key approximations. First, we replace the global state s_t with agent n_i's local observation o_t^i. Second, we consider only the actions of nearby agents $a_{t,o}^{-i}$ instead of the complete joint action set a_t^{-i}. These simplifications are theoretically justified by two established principles: independent causal mechanisms [Peters *et al.*, 2017] and spatiotemporally bounded interactions [Schölkopf *et al.*, 2021] between autonomous entities [Du *et al.*, 2024]. This leads to our simplified CMI formulation:

$$CMI_t^i \approx I(o_{t+1}; a_t^i | o_t, a_{t,o}^{-i}). \tag{2}$$

Since our primary objective is maximizing platform revenue, we focus on how agents' actions influence future task payments within their local observation range. We implement this through a Variational Autoencoder (VAE) [Liu *et al.*, 2023] architecture, where z_ρ encodes the revenue information of nearby tasks. This allows us to reformulate CMI in terms of z_ρ and a_t^i, yielding:

$$\begin{aligned} CMI_t^i &\approx I(z_\rho; a_t^i | o_t, a_{t,o}^{-i}) \\ &\approx D_{KL}\left[q(z_\rho | o_t, a_t^i, a_{t,o}^{-i}) || \; q(z_\rho^{-i} | o_t, a_{t,o}^{-i})\right]. \end{aligned} \tag{3}$$

The final expression employs Bayesian surprise [Li *et al.*, 2024; Mazzaglia *et al.*, 2022] to compute CMI. Our VAE implementation, illustrated in Fig.5, consists of:

$$\begin{aligned} &\text{Encoder:} && q(z_\rho | o_t, a_t^i, a_t^{-i}), \\ &\text{Decoder:} && p(\sum_{\tau \in \mathcal{T}} \rho | z_\rho). \end{aligned}$$

With the corresponding loss function:

$$\begin{aligned} J = -D_{KL}\left[q\left(z | o_t, a_t^i\right) || p(z)\right] + \\ \mathrm{E}_{z \sim q(z|\cdot)}\left[\log p(\sum_{\tau \in \mathcal{T}} \rho | z_\rho)\right]. \end{aligned} \tag{4}$$

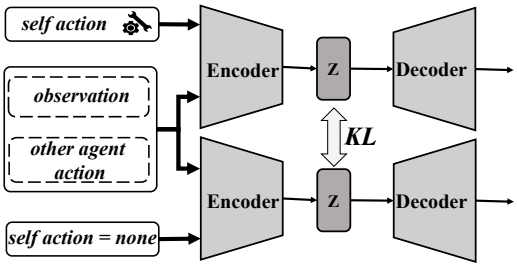


Figure 5: The reward for agent's action a_t^i is constructed using a Variational Autoencoder (VAE). The encoder architecture mirrors the agent's critic structure, while the decoder is implemented as a fully connected neural network.

Relying solely on an action's state-impact assessment as the agent's reward is suboptimal. Traditional reward mechanisms typically incentivize actions that dramatically alter the environment, which may diverge from the goal of maximizing platform revenue. To mitigate this misalignment, we introduce a regularization term into the reward function:

$$r^i(\phi) = \phi_1 \cdot CMI_t^i + \phi_2. \tag{5}$$

While ϕ_1 and ϕ_2 are hyperparameters. In the next subsection. we will introduce the method for determining the specific values of hyperparameters ϕ.

4.4 Implicitly Learning

To optimize the hyperparameters in the reward function, we propose a novel bi-level optimization method based on implicit gradients [Rajeswaran *et al.*, 2019] to tune the reward function's hyperparameters. This approach enables the concurrent optimization of hyperparameters during the reinforcement learning process.

The bi-level optimization method consists of two levels: an inner-level that optimizes the reinforcement learning model and an outer-level that optimizes the reward function's hyperparameters. The loss function can be expressed as:

$$\begin{aligned} &\text{inner-level:} && \theta^* = \arg\max_\theta J(\theta), \\ &\text{outer-level:} && \phi^* := \arg\max_\phi \left[J(\theta, \phi) - \mathcal{L}_\Lambda\right], \end{aligned}$$

while J is the target function of the actor's network. And we aim to ensure that the rewards after redistribution still maintain the same long-term return as the undivided rewards, thereby maintaining the invariance of the optimal policy. To achieve this, we use $\mathcal{L}_\Lambda$ to regulate the long-term reward, which is expressed as:

$$\mathcal{L}_\Lambda(\phi) = \mathrm{E}_{\lambda \sim \Lambda}\left[\left(\sum_{t=1}^{T} r(\phi) - \eta \cdot \sum_{t=1}^{T} \rho\right)^2\right]. \tag{6}$$

η is a hyperparameter used to control the magnitude of the reward, and $r(\phi)$ is the reward function of ϕ.

We define $F(\phi)$ as the outer-level loss function, and its gradient is:

$$\frac{\mathrm{d}F(\phi)}{\mathrm{d}\phi} = \frac{\mathrm{d}J}{\mathrm{d}\phi} - \frac{\mathrm{d}\mathcal{L}_\Lambda}{\mathrm{d}\phi}, \tag{7}$$

while $\mathrm{d}\mathcal{L}_\Lambda/\mathrm{d}\phi$ is straightforward to compute, $\mathrm{d}J(\theta^*,\phi)/\mathrm{d}\phi$ is more challenging to calculate. We can derive $\mathrm{d}J(\theta^*,\phi)/\mathrm{d}\phi$ using the chain rule as:

$$\frac{\mathrm{d}J}{\mathrm{d}\phi} = \frac{\partial J}{\partial \phi} + \frac{\partial J}{\partial \theta}\cdot\frac{\partial \theta}{\partial \phi}. \tag{8}$$

The key to calculating $\mathrm{d}J(\theta^*,\phi)/\mathrm{d}\phi$ is to solve for $\frac{\partial\theta}{\partial\phi}$. First, consider the situation within the inner level, where the inner loss function has been optimized, meaning that θ attains a reasonable parameter value. In this case, the gradient of θ is close to 0:

$$\frac{\mathrm{d}J}{\mathrm{d}\theta} = \frac{\partial J}{\partial \theta} = 0. \tag{9}$$

Thus, in order to get $\partial J/\partial\theta$, we differentiate both sides of the above equation with respect to ϕ:

$$\frac{\partial}{\partial\phi}\left(\frac{\partial J}{\partial\theta}\right) = 0. \tag{10}$$

Next, apply the chain rule to solve the resulting equation:

$$\frac{\partial^2 J}{\partial\theta_i\partial\theta_j}\cdot\frac{\partial\theta}{\partial\phi} + \frac{\partial^2 J}{\partial\theta\partial\phi} = 0. \tag{11}$$

Rearranging the above equation, we can get:

$$\frac{\partial\theta}{\partial\phi} = -\left(\frac{\partial^2 J}{\partial\theta_i\partial\theta_j}\right)^{-1}\cdot\frac{\partial^2 J}{\partial\theta\partial\phi}. \tag{12}$$

Here, $\partial^2 J/\left(\partial\theta_i\partial\theta_j\right)$ represents the Hessian matrix. We calculate $\partial\theta/\partial\phi$ by combining the inverse of the Hessian matrix with the secondary gradient $\partial^2 J/\left(\partial\theta\partial\phi\right)$. Therefore, $\mathrm{d}J/\mathrm{d}\phi$ is computed as:

$$\frac{\mathrm{d}J}{\mathrm{d}\phi} = \frac{\partial J}{\partial\phi} - \frac{\partial J}{\partial\theta}\cdot\left(\frac{\partial^2 J}{\partial\theta_i\partial\theta_j}\right)^{-1}\cdot\frac{\partial^2 J}{\partial\theta\partial\phi}. \tag{13}$$

Computing second-order derivatives requires substantial computational resources, particularly for neural networks with numerous parameters, which significantly increases the computational complexity of secondary gradient calculations. Yosinski *et al.* demonstrated that the initial layers of neural networks capture transferable knowledge, while the later layers specialize in task-specific features. Building upon these findings, we propose replacing the complete network θ with only its later layers θ^l, focusing our optimization efforts on these task-specific components. This approach substantially reduces the computational overhead by limiting calculations to a subset of the network parameters. So the gradient of ϕ can be represented as:

$$\frac{\mathrm{d}F}{\mathrm{d}\phi} \approx \frac{\partial J}{\partial\phi} - \frac{\partial J}{\partial\theta^l}\cdot\left(\frac{\partial^2 J}{\partial\theta_i^l\partial\theta_j^l}\right)^{-1}\cdot\frac{\partial^2 J}{\partial\theta^l\partial\phi} - \frac{\mathrm{d}\mathcal{L}_\Lambda}{\mathrm{d}\phi}. \tag{14}$$

4.5 Swift Parameter Refinement

In real-world crowdsourcing scenarios, learning from large-scale, complicated, and noisy data presents significant challenges. When task data distributions are non-stationary, a Reward Redistribution network with fixed parameters may not generalize effectively across diverse conditions. However, by leveraging the inherent correlations between tasks, rapid parameter adaptation allows the network to capture trajectory-specific patterns better and improve performance across varying contexts. Therefore, in this subsection, we propose a method for rapidly adapting parameters to better handle variations in real-world scenarios.

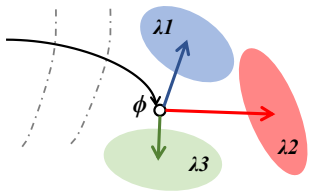


Figure 6: Fine-tuning the parameters to adapt to changes in crowdsourcing tasks quickly.

Firstly, we approximate Equ.6 using the Taylor series expansion to the second-order term[Wu *et al.*, 2024c]:

$$\begin{aligned}\mathcal{L}_\Lambda(\phi) \approx & \mathcal{L}_\Lambda(\hat{\phi}) + \left(\phi-\hat{\phi}\right)^T\cdot\left[\frac{\mathrm{d}\mathcal{L}_\Lambda(\phi)}{\mathrm{d}\phi}\bigg|_{\phi=\hat{\phi}}\right] + \\ & \frac{1}{2}\left(\phi-\hat{\phi}\right)^T\cdot\left[\frac{\mathrm{d}^2\mathcal{L}_\Lambda(\phi)}{\mathrm{d}\phi^2}\bigg|_{\phi=\hat{\phi}}\right]\cdot\left(\phi-\hat{\phi}\right).\end{aligned} \tag{15}$$

While $\hat{\phi}$ is the best possible initialization obtained through optimal multi-trajectory optimization. And then we take the derivative of the above expression with respect to ϕ:

$$\frac{\mathrm{d}\mathcal{L}_\Lambda}{\mathrm{d}\phi} \approx \left[\frac{\mathrm{d}\mathcal{L}_\Lambda(\phi)}{\mathrm{d}\phi}\bigg|_{\phi=\hat{\phi}}\right] + \left[\frac{\mathrm{d}^2\mathcal{L}_\Lambda(\phi)}{\mathrm{d}\phi^2}\bigg|_{\phi=\hat{\phi}}\right]\cdot\left(\phi-\hat{\phi}\right). \tag{16}$$

Consider the concept of implicit gradients for optimizing the network parameters. If ϕ^* is the optimal neural network parameter for the dataset, then the gradient of the loss function at ϕ^* is zero, i.e.:

$$\frac{\mathrm{d}\mathcal{L}_\Lambda(\phi^*)}{\mathrm{d}\phi} = 0, \tag{17}$$

This leads to the following expression:

$$\left[\frac{\mathrm{d}\mathcal{L}_\Lambda(\phi)}{\mathrm{d}\phi}\bigg|_{\phi=\hat{\phi}}\right] + \left[\frac{\mathrm{d}^2\mathcal{L}_\Lambda(\phi)}{\mathrm{d}\phi^2}\bigg|_{\phi=\hat{\phi}}\right]\cdot\left(\phi^*-\hat{\phi}\right) = 0. \tag{18}$$

Therefore, we can observe the optimal parameters ϕ^* in a single step as follows:

$$\phi^* = \hat{\phi} - \left[\frac{\mathrm{d}\mathcal{L}_\Lambda(\phi)}{\mathrm{d}\phi}\bigg|_{\phi=\hat{\phi}}\right]\cdot\left[\frac{\mathrm{d}^2\mathcal{L}_\Lambda(\phi)}{\mathrm{d}\phi^2}\bigg|_{\phi=\hat{\phi}}\right]^{-1}. \tag{19}$$

4.6 Model Training

The Independent Proximal Policy Optimization (IPPO) algorithm [De Witt *et al.*, 2020] is utilized as an example. The pseudocode is presented in Algorithm 1.

In IPPO, the objective function of the actor network and the loss function of the critic network respectively are as follows:

$$\begin{aligned}&\text{actor:} \quad J(\theta) = \mathrm{E}_{a\sim\pi_\theta(\cdot|o)}\min\left[\psi A^n, \mathrm{clip}\left(\psi, 1\pm\epsilon\right)A^n\right],\\ &\text{critic:} \quad \mathcal{L}(\varphi) = \mathrm{E}[(V_\varphi(o_t) - V^{target})^2],\end{aligned}$$

Algorithm 1 Model Training and Fine-Tuning Algorithm

Input: actor network θ and critic network ϕ
Output: θ,ϕ

Initialize policy network θ, value network φ, hyperparameters ϕ_1 and ϕ_2.
Initialize replay buffer D
for episode $\leftarrow 1, \ldots, E$ **do**
  Use θ to execute actions and collect experience, then store transitions $(s_t, \{o_t^n\}, \{a_t^n\}, \{r_t^n\}, s_{t+1})$ in the data buffer D
  if parameter ϕ has been sufficiently trained **then**
    Fine-tune ϕ by Equ.(19)
  end if
  for $iteration \leftarrow 1$ **to** M **do**
    Sample mini-batch from D
    for each agent $n \in N$ **do**
      Update actor network θ and critic network ψ
      Update hyper parameter ϕ by Equ.(14)
    end for
  end for
end for
return θ, ϕ

where $\psi = \frac{\pi_\theta(a_t|o_t)}{\pi_{\theta_{old}}(a_t|o_t)}$ is the probability ratio between the current policy and the old policy and ϵ is a hyperparameter that controls the clipping range. $V_\phi(o_t)$ is the current estimate of the value function and V^{target} is the target value.

5 Experimental Evaluation

In this section, we evaluate CAFE on some Simulation experiments to answer the following questions: **Q1:** Is the credit assignment performance of CAFE superior to that of state-of-the-art frameworks in delayed reward settings? **Q2:** Is the fine-tuning method for CAFE accurate and effective in facilitating reinforcement learning? **Q3:** Is the performance of CAFE superior to that of state-of-the-art collaborative spatial crowdsourcing methods?

5.1 Experimental Setup

Datasets. The road network data was extracted from OpenStreetMap for two cities: Chengdu (17,378 nodes, 26,961 edges) and Haikou (9,667 nodes, 14,458 edges). For spatial tasks, orders were extracted from DiDi, comprising 7,065,937 orders for Chengdu and 14,160,170 orders for Haikou, where the pick-up location of the order corresponds to the location of the spatial tasks. The task skill model was developed based on Point-of-Interest (POI) types from AutoNavi Maps, classified into 8 distinct categories. Each task derives skill requirements from nearby POIs. Worker skills were modeled according to a normal distribution ($\mu = 2.5, \sigma^2 = 1$), while worker initial locations were also initialized by DiDi order.

Implementation Details. The Adam optimizer [Kingma, 2014] was employed for training all models, with a learning rate of 1×10^{-3}. And the hyperparameter update rate is maintained at 1×10^{-3}. In the simulation experiments, worker speed was set to 10 m/s, with an acceptable task range of 5 km. All experimental procedures are executed on a computational system operating Windows 10 with Python 3.8, equipped with Intel Core i9-13900K CPU @ 5.80 GHz, NVIDIA GeForce RTX 4080 super GPU, and 64 GB RAM.

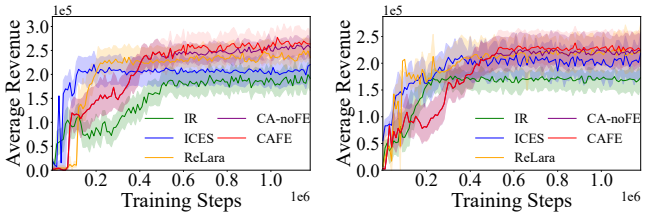
Baselines of Reward Redistribution. We use the following algorithm to compare CAFE's performance in multi-agent reinforcement learning with delayed rewards: ***Individual Reward (IR)***, uses the remaining task working time as the reward; ***ICES*** [Li *et al.*, 2024], uses individual contributions as intrinsic exploration scaffolds to motivate exploration by assessing each agent's contribution from a global perspective; ***ReLara*** [Ma *et al.*, 2024], a dual-agent reward shaping framework composed of two synergistic agents, a policy agent to learn the optimal behavior and a reward agent to generate reward. ***CA-noFE***, derived from CAFE but without the fine-tuning component.

Baselines of Task Assignment. We use the following algorithm to compare CAFE's ability for task assignment in the context of the simulated spatial crowdsourcing experiment: ***Greedy***, assigns workers greedily to the nearest tasks that match the required skills; ***IPPO*** [De Witt *et al.*, 2020], utilizes the same network as CAFE, but with individual reward training; ***PAU*** [Zhao *et al.*, 2024], achieves task assignment through the formation of worker coalitions, modeling the CSC problem as an exact potential game; ***CA-noFE***, derived from CAFE but without the fine-tuning component.

5.2 Comparison of Reward Redistribution

In this subsection, we evaluate the performance of different reward redistribution methods under identical crowdsourcing environments. Using platform revenue as the evaluation metric, we compare the performance variations of models trained with different methods across various training batches. The experimental results are presented in Fig.7. Notably, in the training of CAFE, we first train the model using the CA-noFE method to reduce computational requirements. Once the model training stabilizes, we then introduce the fine-tuning method to improve the reinforcement learning training.

During model training, the IR method progressed slowly due to its simple reward assignment approach. In contrast, ICES and ReLara utilized more advanced reward allocation policies, resulting in faster training. Although CAFE initially showed significant performance fluctuations and required more iterations, it eventually outperformed the other methods. This improvement is attributed to CAFE's second-order gradient optimization, which takes longer to converge



(a) Chengdu datasets (b) Haikou datasets

Figure 7: Comparison of reward redistribution approaches in model training.

Dataset	Methods	$N_\tau = 50$	$N_\tau = 100$	$N_\tau = 150$	$N_\tau = 200$	$N_\tau = 250$	$N_\tau = 300$	$N_\tau = 350$
Chengdu	Greedy	6.53 ±0.16	12.14 ±0.17	17.77 ±0.37	21.85 ±0.15	26.07 ±0.39	29.30 ±0.28	32.94 ±0.81
	IPPO	6.47 ±0.22	12.42 ±0.26	17.25 ±0.42	23.74 ±0.80	27.01 ±0.36	34.40 ±1.16	40.09 ±1.25
	PAU	6.57 ±0.22	11.65 ±0.20	18.36 ±0.88	24.18 ±0.79	26.94 ±0.81	33.57 ±0.92	39.88 ±1.86
	CA-noFE	6.59 ±0.18	12.52 ±0.31	18.10 ±0.55	24.40 ±0.80	27.43 ±0.73	34.55 ±1.06	41.14 ±1.53
	CAFE	**6.69 ±0.17**	**12.67 ±0.22**	**18.48 ±0.80**	**24.66 ±0.71**	**27.63 ±0.74**	**35.37 ±0.98**	**41.55 ±1.42**
Haikou	Greedy	6.42 ±0.14	11.78 ±0.40	17.03 ±0.22	21.52 ±0.39	26.16 ±0.50	29.18 ±0.39	31.25 ±0.42
	IPPO	6.17 ±0.18	12.23 ±0.43	17.30 ±0.32	22.84 ±0.32	27.19 ±0.24	32.48 ±0.88	35.03 ±1.38
	PAU	6.68 ±0.15	11.88 ±0.19	18.25 ±0.79	23.28 ±0.90	25.95 ±0.90	32.87 ±1.01	35.48 ±0.88
	CA-noFE	6.64 ±0.15	12.14 ±0.31	**18.52 ±0.59**	23.28 ±0.90	27.55 ±0.38	33.21 ±0.57	36.09 ±1.50
	CAFE	**6.71 ±0.15**	**12.43 ±0.35**	18.51 ±0.47	**23.40 ±0.87**	**28.37 ±0.45**	**34.15 ±0.69**	**36.99 ±0.97**

Table 1: Revenue performance across varying numbers of tasks (with $N_w = 200$, scaled by $\times 10^4$).

Dataset	Methods	$N_w = 50$	$N_w = 100$	$N_w = 150$	$N_w = 200$	$N_w = 250$	$N_w = 300$	$N_w = 350$
Chengdu	Greedy	7.63 ±0.42	15.67 ±0.51	19.81 ±0.38	21.85 ±0.15	22.88 ±0.37	23.76 ±0.36	24.22 ±0.28
	IPPO	10.45 ±0.34	20.82 ±1.14	23.48 ±1.01	23.74 ±0.80	22.79 ±0.31	24.42 ±0.20	23.71 ±0.42
	PAU	11.48 ±0.49	21.54 ±1.05	23.96 ±0.90	24.18 ±0.79	24.73 ±0.85	24.17 ±0.62	24.94 ±0.55
	CA-noFE	11.94 ±0.62	21.82 ±0.89	24.24 ±1.04	24.40 ±0.80	25.29 ±0.93	24.90 ±0.51	25.63 ±0.60
	CAFE	**12.17 ±0.49**	**21.85 ±1.02**	**24.60 ±0.93**	**24.66 ±0.71**	**25.66 ±0.86**	**25.69 ±0.58**	**26.56 ±0.59**
Haikou	Greedy	7.71 ±0.30	15.89 ±0.44	19.75 ±0.45	21.52 ±0.39	23.10 ±0.28	23.65 ±0.26	23.98 ±0.38
	IPPO	15.92 ±1.90	21.80 ±0.60	22.16 ±1.24	22.84 ±0.32	24.12 ±0.25	26.68 ±0.33	27.63 ±0.28
	PAU	16.38 ±1.73	22.58 ±0.38	23.14 ±1.02	23.28 ±0.90	23.98 ±0.91	26.89 ±0.60	**30.31 ±1.21**
	CA-noFE	16.60 ±1.83	22.87 ±0.45	23.21 ±0.79	23.28 ±0.90	24.51 ±0.80	27.26 ±0.49	29.87 ±0.55
	CAFE	**16.62 ±1.76**	**23.32 ±0.33**	**23.60 ±0.87**	**23.40 ±0.87**	**24.75 ±0.91**	**27.65 ±0.45**	30.02 ±0.54

Table 2: Revenue performance across varying numbers of workers (with $N_\tau = 200$, scaled by $\times 10^4$).

but enables more precise reward allocation. The results confirm CAFE's effectiveness in credit assignment, allowing for more nuanced reward distribution to agents. Compared to the non-fine-tuned CA-noFE method, CAFE enhances training dynamics and generates higher platform revenue, highlighting the importance of its fine-tuning methodology.

5.3 Performance Analysis and Evaluation

We evaluated the proposed algorithms by conducting experiments with different numbers of workers and tasks, using the platform's total revenue as the main metric. All results are averages of five independent runs.

Effect of the number of Tasks. Table 1 compares the revenue performance of various algorithms across different numbers of tasks for two datasets. The experiments were conducted with a fixed number of workers ($N_w = 200$), and the results highlight the revenue trends as N_w increases from 50 to 350. The CAFE algorithm consistently achieves the highest revenue across both datasets, demonstrating its superiority in optimizing task allocation. The results demonstrate that the CAFE outperforms all other methods across different task numbers and datasets, particularly excelling in large-scale task scenarios. This confirms its effectiveness in optimizing task allocation policies.

Effect of the number of workers. Table 2 compares the revenue performance of various algorithms across different numbers of workers, with a fixed number of tasks ($N_\tau = 200$). The results demonstrate that CAFE consistently outperforms other methods, achieving the highest revenues across most configurations, particularly a sN_w increase. Notably, CAFE exhibits superior scalability and stability, with lower variance in performance compared to alternatives such as CA-noFE and PAU. While Greedy performs the worst due to its inability to optimize task-worker assignments effectively, IPPO provides moderate results but fails to match the advanced methods. These findings highlight the robustness of CAFE in leveraging increased worker availability to maximize platform revenue.

The experimental results conclusively demonstrate CAFE's superior performance compared to state-of-the-art collaborative spatial crowdsourcing methods, as evidenced by its higher platform revenue and improved task assignment efficiency. The fine-tuning approach effectively captures each task's unique characteristics, enabling precise reward allocation and enhancing training dynamics.

6 Conclusion

This paper introduces a novel credit assignment method to address delayed reward challenges in cooperative spatial crowdsourcing using multi-agent reinforcement learning. By leveraging causality and Bayesian theories, we analyze agents' contributions and develop a meta-optimization approach for training reward functions. We further propose a rapid parameter adjustment technique to mitigate non-stationary data distributions across spatial tasks. Extensive experiments on two datasets validate the superior performance of our method compared to existing approaches.

Acknowledgments

This work is supported by the following grants: NSFC Grants 62372416, 61972362, 62036010, and 62325602; HNSF Grant 242300421215.

References

[Cheng *et al.*, 2016] Peng Cheng, Xiang Lian, Lei Chen, Jinsong Han, and Jizhong Zhao. Task assignment on multi-skill oriented spatial crowdsourcing. *IEEE Transactions on Knowledge and Data Engineering*, 28(8):2201–2215, 2016.

[Cheng *et al.*, 2019] Peng Cheng, Lei Chen, and Jieping Ye. Cooperation-aware task assignment in spatial crowdsourcing. In *2019 IEEE 35th International Conference on Data Engineering (ICDE)*, pages 1442–1453. IEEE, 2019.

[De Witt *et al.*, 2020] Christian Schroeder De Witt, Tarun Gupta, Denys Makoviichuk, Viktor Makoviychuk, Philip HS Torr, Mingfei Sun, and Shimon Whiteson. Is independent learning all you need in the starcraft multi-agent challenge? *arXiv preprint arXiv:2011.09533*, 2020.

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

[Jiang *et al.*, 2023] Lin Jiang, Shuai Wang, Baoshen Guo, Hai Wang, Desheng Zhang, and Guang Wang. Faircod: a fairness-aware concurrent dispatch system for large-scale instant delivery services. In *Proceedings of the 29th ACM SIGKDD Conference on knowledge discovery and data mining*, pages 4229–4238, 2023.

[Kingma, 2014] Diederik P Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[Li *et al.*, 2022a] Yafei Li, Huiling Li, Xin Huang, Jianliang Xu, Yu Han, and Mingliang Xu. Utility-aware dynamic ridesharing in spatial crowdsourcing. *IEEE Transactions on Mobile Computing*, 23(2):1066–1079, 2022.

[Li *et al.*, 2022b] Yafei Li, Yifei Li, Yun Peng, Xiaoyi Fu, Jianliang Xu, and Mingliang Xu. Auction-based crowdsourced first and last mile logistics. *IEEE Transactions on Mobile Computing*, 23(1):180–193, 2022.

[Li *et al.*, 2022c] Yafei Li, Qingshun Wu, Xin Huang, Jianliang Xu, Wanru Gao, and Mingliang Xu. Efficient adaptive matching for real-time city express delivery. *IEEE Transactions on Knowledge and Data Engineering*, 35(6):5767–5779, 2022.

[Li *et al.*, 2023] Yafei Li, Huiling Li, Baolong Mei, Xin Huang, Jianliang Xu, and Mingliang Xu. Fairness-guaranteed task assignment for crowdsourced mobility services. *IEEE Transactions on Mobile Computing*, 2023.

[Li *et al.*, 2024] Xinran Li, Zifan Liu, Shibo Chen, and Jun Zhang. Individual contributions as intrinsic exploration scaffolds for multi-agent reinforcement learning. *arXiv preprint arXiv:2405.18110*, 2024.

[Li *et al.*, 2025] Yafei Li, Wei Chen, Jinxing Yan, Huiling Li, Lei Gao, and Mingliang Xu. Gradient-guided credit assignment and joint optimization for dependency-aware spatial crowdsourcing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 14301–14308, 2025.

[Liu *et al.*, 2023] Boyin Liu, Zhiqiang Pu, Yi Pan, Jianqiang Yi, Yanyan Liang, and Du Zhang. Lazy agents: a new perspective on solving sparse reward problem in multi-agent reinforcement learning. In *International Conference on Machine Learning*, pages 21937–21950. PMLR, 2023.

[Lowe *et al.*, 2017] Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, OpenAI Pieter Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. *Advances in Neural Information Processing Systems*, 30, 2017.

[Ma *et al.*, 2024] Haozhe Ma, Kuankuan Sima, Thanh Vinh Vo, Di Fu, and Tze-Yun Leong. Reward shaping for reinforcement learning with an assistant reward agent. In *Proceedings of the 41st International Conference on Machine Learning*, pages 33925–33939, 2024.

[Mazzaglia *et al.*, 2022] Pietro Mazzaglia, Ozan Catal, Tim Verbelen, and Bart Dhoedt. Curiosity-driven exploration via latent bayesian surprise. In *Proceedings of the AAAI conference on artificial intelligence*, pages 7752–7760, 2022.

[Mei *et al.*, 2024] Baolong Mei, Yafei Li, Wei Chen, Linshen Luan, Guanglei Zhu, Yuanyuan Jin, and Jianliang Xu. Catcher: A cache analysis system for top-k pub/sub service. *Proceedings of the VLDB Endowment*, 17(12):4389–4392, 2024.

[Peters *et al.*, 2017] Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. *Elements of causal inference: foundations and learning algorithms*. The MIT Press, 2017.

[Qu *et al.*, 2024] Yun Qu, Yuhang Jiang, Boyuan Wang, Yixiu Mao, Cheems Wang, Chang Liu, and Xiangyang Ji. Latent reward: Llm-empowered credit assignment in episodic reinforcement learning. *arXiv preprint arXiv:2412.11120*, 2024.

[Rajeswaran *et al.*, 2019] Aravind Rajeswaran, Chelsea Finn, Sham M Kakade, and Sergey Levine. Meta-learning with implicit gradients. *Advances in neural information processing systems*, 32, 2019.

[Ren *et al.*, 2021] Zhizhou Ren, Ruihan Guo, Yuan Zhou, and Jian Peng. Learning long-term reward redistribution via randomized return decomposition. *arXiv preprint arXiv:2111.13485*, 2021.

[Schölkopf *et al.*, 2021] Bernhard Schölkopf, Francesco Locatello, Stefan Bauer, Nan Rosemary Ke, Nal Kalchbrenner, Anirudh Goyal, and Yoshua Bengio. Toward causal representation learning. *Proceedings of the IEEE*, 109(5):612–634, 2021.

[Tong *et al.*, 2021] Yongxin Tong, Dingyuan Shi, Yi Xu, Weifeng Lv, Zhiwei Qin, and Xiaocheng Tang. Combinatorial optimization meets reinforcement learning: Effective taxi order dispatching at large-scale. *IEEE Transactions on Knowledge and Data Engineering*, 35(10):9812–9823, 2021.

[Vaswani, 2017] A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

[Wang *et al.*, 2022] Li Wang, Yupeng Zhang, Yujing Hu, Weixun Wang, Chongjie Zhang, Yang Gao, Jianye Hao, Tangjie Lv, and Changjie Fan. Individual reward assisted multi-agent reinforcement learning. In *International Conference on Machine Learning*, pages 23417–23432. PMLR, 2022.

[Wang *et al.*, 2023] Yu Wang, Jingfei Wu, Xingyuan Hua, Chi Harold Liu, Guozheng Li, Jianxin Zhao, Ye Yuan, and Guoren Wang. Air-ground spatial crowdsourcing with uav carriers by geometric graph convolutional multi-agent deep reinforcement learning. In *2023 IEEE 39th International Conference on Data Engineering (ICDE)*, pages 1790–1802, 2023.

[Wu *et al.*, 2024a] Qingshun Wu, Yafei Li, Jinxing Yan, Mei Zhang, Jianliang Xu, and Mingliang Xu. Adaptive task assignment in spatial crowdsourcing: A human-in-the-loop approach. *IEEE Transactions on Mobile Computing*, 2024.

[Wu *et al.*, 2024b] Qingshun Wu, Yafei Li, Guanglei Zhu, Baolong Mei, Jianliang Xu, and Mingliang Xu. Prediction-aware adaptive task assignment for spatial crowdsourcing. *IEEE Transactions on Mobile Computing*, 2024.

[Wu *et al.*, 2024c] Yichen Wu, Long-Kai Huang, Renzhen Wang, Deyu Meng, and Ying Wei. Meta continual learning revisited: Implicitly enhancing online hessian approximation via variance reduction. In *The Twelfth international conference on learning representations*, 2024.

[Ye *et al.*, 2023] Yuxiao Ye, Chi Harold Liu, Zipeng Dai, Jianxin Zhao, Ye Yuan, Guoren Wang, and Jian Tang. Exploring both individuality and cooperation for air-ground spatial crowdsourcing by multi-agent deep reinforcement learning. In *2023 IEEE 39th International Conference on Data Engineering (ICDE)*, pages 205–217, 2023.

[Yosinski *et al.*, 2014] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? *Advances in neural information processing systems*, 27, 2014.

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

[Zhan *et al.*, 2024] Zhongwei Zhan, Yingjie Wang, Peiyong Duan, Akshita Maradapu Vera Venkata Sai, Zhaowei Liu, Chaocan Xiang, Xiangrong Tong, Weilong Wang, and Zhipeng Cai. Enhancing worker recruitment in collaborative mobile crowdsourcing: A graph neural network trust evaluation approach. *IEEE Transactions on Mobile Computing*, 2024.

[Zhang *et al.*, 2024] Hongbo Zhang, Guang Wang, Xu Wang, Zhengyang Zhou, Chen Zhang, Zheng Dong, and Yang Wang. Nondbrem: Nondeterministic offline reinforcement learning for large-scale order dispatching. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 401–409, 2024.

[Zhao *et al.*, 2023] Yan Zhao, Jiaxin Liu, Yunchuan Li, Dalin Zhang, Christian S Jensen, and Kai Zheng. Preference-aware group task assignment in spatial crowdsourcing: Effectiveness and efficiency. *IEEE Transactions on Knowledge and Data Engineering*, 35(10):10722–10734, 2023.

[Zhao *et al.*, 2024] Yan Zhao, Kai Zheng, Ziwei Wang, Liwei Deng, Bin Yang, Torben Bach Pedersen, Christian S Jensen, and Xiaofang Zhou. Coalition-based task assignment with priority-aware fairness in spatial crowdsourcing. *The VLDB Journal*, 33(1):163–184, 2024.

[Zhu *et al.*, 2023] Tianchen Zhu, Yue Qiu, Haoyi Zhou, and Jianxin Li. Towards long-delayed sparsity: Learning a better transformer through reward redistribution. In *IJCAI*, pages 4693–4701, 2023.

[Zong *et al.*, 2022] Zefang Zong, Meng Zheng, Yong Li, and Depeng Jin. Mapdp: Cooperative multi-agent reinforcement learning to solve pickup and delivery problems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 9980–9988, 2022.