Efficient Worker Recruitment Approach Based on Social Incentive Diffusion in Mobile Crowdsourcing

Haojun Teng[1], Zhongwei Zhan[2], Yingjie Wang[3*], Wenhan Hou[4], Yihen Pan[5], Zhipeng Cai[6]
[1,3,4]School of Computer and Control Engineering, Yantai University, Yantai, China
[2]School of Computer Science and Engineering, Southeast University, Nanjing, China
[5]Department of Engineering, Shenzhen MSU-BIT University, Shenzhen, China
[6]Department of Computer Science, Georgia State University, Atlanta, USA

Email: tenghaojun1008@163.com, zhanzhongwei@seu.edu.cn, towangyingjie@163.com, cshwh@ytu.edu.cn, 3362568858@qq.com, zcai@gsu.edu.cn

***Abstract*—With the advancement of mobile devices and wireless communication technology, Mobile Crowdsourcing (MCS) has emerged as a paradigm for recruiting workers to complete tasks at specific locations. As a variant, social MCS has been proposed to facilitate large-scale collaborative tasks among multiple workers, leveraging MCS social network to expand the worker pool and enhance task utility. This paper introduces the Worker Recruitment approach based on Social Diffusion (WRSD). To model and incentivize workers' social diffusion behavior towards task information, we designed a Social Trust Prediction (SoTP) framework using Graph Convolutional Networks (GCNs) and a social incentive mechanism. For effective worker-task matching, we integrated cross-modal social recommendation data using Graph Attention Networks (GATs) within a Social Task Recommendation (SoTR) framework. The WRSD problem is then modeled as a Constrained Multi-attribute Combinatorial Optimization (CMCO) problem under budget constraints. We define a heuristic neighborhood search strategy and propose the Variable Neighborhood Tabu Search (VNTS) algorithm to solve the WRSD problem, achieving an approximately optimal worker solution for each task. Comprehensive experiments conducted on three real-world datasets validate the effectiveness and efficiency of the proposed approach.**

***Index Terms*—Mobile Crowedsourcing, Worker Recruitment, Social Network, Optimization Algorithm, Graph Neural Network**

I. Introduction

With the rapid development and widespread application of wireless sensing devices and wireless communication technologies, MCS has emerged as a prominent paradigm. MCS leverages the sensing and computing capabilities of large-scale mobile devices to collect, process, and analyze data, thereby completing spatio-temporal tasks in the real world [?], [?], [?], [1]. Typically, MCS involves task publishers, workers, and the MCS platform [2], [3]. Task publishers release specific

*Corresponding author. This work was supported in part by the National Natural Science Foundation of China under Grants 92582115 and 62272405, in part by the Youth Innovation Science and Technology Support Program of Shandong Province under Grant 2021KJ080, in part by the Natural Science Foundation of Shandong Province under Grants ZR2024YQ002 and ZR2022MF238, and in part by the Yantai Science and Technology Innovation Development Plan under Grant 2024ZLYJ121.

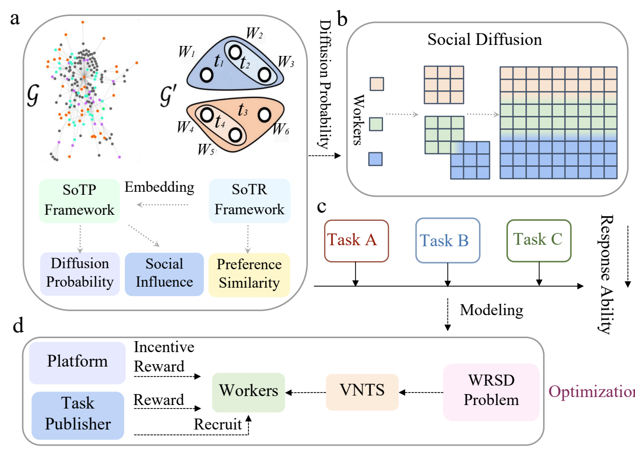


Fig. 1: Illustration of WRSD Framework.

tasks based on their requirements, and qualified workers proceed to designated locations to perform the tasks. Upon task completion, task publishers pay rewards to the workers [?], [4]. The MCS platform is responsible for recruiting qualified workers for tasks, incentivizing value-creating interactions [?], [5], and coordinating the matching between task publishers and workers [?], [6]. A key challenge in MCS worker recruitment research is how to recruit high-quality workers for tasks within a given task budget. Existing recruitment approaches usually consider only the individual attributes of workers, such as spatial, temporal, and ability attributes, while neglecting their social attributes [7]. It has been observed that workers could diffuse task information with cross-platform social friends via social network interactions, improving task execution utility [5], [8]. However, when modeling this social diffusion process, existing approaches often rely on fixed trust rules, which fail to capture the complex nonlinear relationships within social networks [9], [10]. Moreover, incentivizing workers' social diffusion behavior is crucial for maintaining the stability of this process. Therefore, the first challenge identified in this paper is how to effectively model and incentivize workers' social diffusion processes toward task information.

In social MCS, strong social relationships could motivate workers to diffuse task information to social friends and actively participate in the execution of tasks [11]. Studies have shown that higher levels of sociality correspond to higher completion rates of tasks [12]. Additionally, by leveraging social relationships, workers may exert some pressure on their social friends to better execute tasks, thereby improving task quality [13], [14]. Privacy problems are also somewhat alleviated within social networks, as friends are less likely to disclose each other's privacy [**?**], [15]. Existing recruitment approaches often consider only single-hop social relationships between workers [16]. However, trust propagation is inherently multi-hop. For instance, the social relationships between workers are unobserved, but they can build trust indirectly through mutual friends [15]. Therefore, the second challenge identified in this paper is how to predict unobserved social relationships between workers.

Additionally, before accepting tasks, workers typically rate different tasks based on individual rationality. Existing worker recruitment approaches often use historical preference data to match workers with tasks [17]. However, these preference ratings are generally limited, and for unknown tasks, workers' preference ratings are missing. It has been found that a social network can be used to enhance the prediction of missing preference ratings [18]. Workers who are social friends with each other often have similar preferences for tasks. Therefore, how to predict preference ratings between workers and tasks is identified as the third challenge in this paper. Finally, how to model and solve the WRSD problem is another challenge faced in this paper.

To address the aforementioned challenges, this paper proposes a more suitable WRSD framework, as illustrated in Fig. 1. Firstly, SoTR framework based on GATs is designed to predict preference ratings between workers and tasks by integrating cross-modal trust and recommendation features combined with an attention mechanism. Secondly, SoTP framework based on GCNs is designed. Initial embeddings are obtained through pre-training on SoTR framework, followed by the independent propagation of workers as trustor and trustee features to accurately predict social relationships between workers. Based on the workers' social relationships, the social diffusion process towards task information is modeled, and a social incentive mechanism is proposed to motivate the workers' social diffusion behavior. Finally, considering workers' social influence, preference similarity, task response ability and recruited worker scale, the WRSD problem is modeled as a CMCO problem. A heuristic neighborhood search strategy is defined, and a specific VNTS algorithm is proposed to solve the WRSD problem, resulting in the approximate optimal workers solution for tasks.

The main contributions of this paper are summarized as follows:

- SoTR and SoTP frameworks based on GATs and GCNs respectively are designed, integrating cross-modal social network and social recommendation networks to achieve accurate predictions of workers' preference ratings for tasks and social relationships between workers.
- The social diffusion process of workers toward task information is modeled and incentivized based on workers' social relationships, expanding the worker recruitment solution space.
- The WRSD problem is modeled as a CMCO problem under budget constraints, considering workers' social influence, preference similarity, task response ability, and recruited worker scale. A heuristic neighborhood search strategy is defined, and the VNTS algorithm is designed to solve the WRSD problem, obtaining an approximately optimal worker solution.
- Extensive experiments are conducted on three real-world social recommendation datasets, demonstrating that the proposed approach outperforms the state-of-the-art approaches in the literature.

II. RELATED WORK

A. Social Diffusion in MCS

As a large number of workers establish social relationships on the MCS platform, a large and stable social network is formed, allowing task and worker information to be disseminated through these networks [8]. Many researchers have studied task diffusion models. Wang *et al.* [9] proposed the Acceptance-Aware Worker Recruitment (AWR) approach for MCS, defining the probability of task diffusion among workers as the ratio of common friends to total friends, and randomly diffusing tasks based on this probability. Wang *et al.* [19] designed a worker recruitment algorithm using social relationships and mobility trajectories of workers, greedily selecting workers by observing the geographic interdependencies between friends in the social network. Chen *et al.* [10] proposed a task recommendation algorithm for MCS, aiming to maximize task completions by utilizing the social network for task diffusion based on the proportion of common friends among workers. Xiao *et al.* [20] addressed the problem of maximum span-sensitive task allocation in social network, proposing to use workers' mobility to diffuse tasks among workers with overlapping locations, relying on social relationships for autonomous coordination in task execution. Although this paper also considers task diffusion in social network, the fundamental difference lies in the consideration of the diffusion range. In the aforementioned studies, task diffusion is based on observed social relationships, while this paper predicts social relationships between workers who are socially reachable but not observed, thereby significantly expanding the task diffusion range compared to the approaches in the aforementioned studies.

B. Worker Recruitment in MCS

Worker recruitment has always been a critical issue in MCS research, with a focus on how to recruit suitable workers for tasks [21]. Several approaches have been proposed. Wang *et al.* [17] proposed a graph theory-based algorithm to solve the single-round worker recruitment problem and an Multi-armed

Bandit (URMB) model for multi-round recruitment under budget constraints, aiming to maximize data quality. Zhan *et al.* [15] proposed a tabu search recruitment algorithm to maximize task completion under privacy loss constraints. Gao *et al.* [22] proposed a Learning-based Credible Participant Recruitment Strategy (LC-PRS) to recruit trustworthy workers and inspire them to contribute high-quality data, aiming to maximize the benefits of platform and worker. Gao *et al.* [23] converted the problem of unknown worker quality into an multi-armed bandit problem, proposing a UCB-based worker recruitment algorithm. However, the worker recruitment approaches proposed in the aforementioned studies typically target fixed or isolated MCS worker sets, which are not suitable for the WRSD problem that considers the social diffusion of tasks.

III. SYSTEM MODEL AND PROBLEM FORMULATION

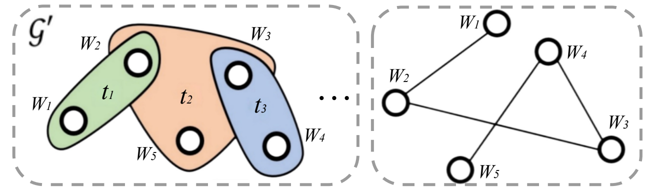
A. System Model

In this section, the definitions of terms and process descriptions involved in the approach in this paper will be elaborated upon.

Definition 1: Worker. The worker set is represented as $W = \{w_1, w_2, ..., w_m\}$. This set is dynamically changing to accommodate worker $w_i \in W$ to invite social friends through social diffusion to join W. The historical task interaction set of $w_i \in W$ is represented as $I(w_i) = \{t_1, ..., t_k\}$. The location of w_i is represented as $loc_i = \{lon_i, lat_i\} \in L$. The MCS social network is denoted as $G = (V_w, E, \Theta)$, where V_w represents the set of social workers, including $w_i \in W$ and their social friends. $e_{ij} \in E$ denotes the directed social relationship between w_i and w_j, and θ_{ij} denotes the trust score of social relationship between w_i and w_j. $\theta_{ij} = 1$ denotes that w_i trusts w_j, while $\theta_{ij} = 0$ denotes that w_i does not trust w_j.

Definition 2: Task. The set of tasks is represented as $T = \{t_1, t_2, ..., t_n\}$. The historical worker interaction set of $t_j \in T$ is represented as $I(t_j) = \{w_1, ..., w_k\}$. The location of t_j is represented as $loc_j = (lon_j, lat_j) \in L$. The budget of t_j for worker recruitment is represented as β_j. Furthermore, a hypergraph $G^{'} = (V_w, V_t, E_w, E_t, R)$ is introduced to represent heterogeneous worker-task elements, as shown in Fig. 2a. In $G^{'}$, V_t represents the set of historical tasks, $E_t = \{I(t_j), \forall t_j \in V_t\}$ represents the set of task hyperedges, $E_w = \{I(w_i), \forall w_i \in V_w\}$ represents the set of worker hyperedges, and R represents the set of preference ratings. It is noteworthy that the preference rating of w_i for t_j is equal to the preference rating of t_j for w_i, i.e., $r_{ij} = r_{ji}$.

Definition 3: Social Reachability. Social reachability refers to workers being connected through direct or indirect social relationships. If w_i can connect to w_j via a series of social friends, they are socially reachable, i.e., $sr_{ij} = 1$. Specifically, as shown in Fig 2.b, w_1 has a direct social relationship with w_2, and w_2 has a direct social relationship with w_3. According to the definition of social reachability, w_1 is socially reachable to w_2 because of their direct social relationship, and w_1 is socially reachable to w_3 despite the lack of a direct social relationship, as they are connected through w_2 via the social



a. Hypergraph b. Social Reachability

Fig. 2: Illustration of hypergraph G' and social reachability.

chain w_1-w_2-w_3.

Definition 4: Social Diffusion. Based on social reachability, the cross-layer social diffusion nature of workers towards tasks in MCS social network is further considered. Specifically, a worker diffuses task information to their social friends, who then further diffuse task information to their own social friends, forming a multi-layer social diffusion model. The dynamic diffusion process is modeled based on social diffusion probabilities. The social diffusion probability df_{ij} from w_i to w_j is calculated as follows:

$$df_{ij} = p(\tilde{\theta}_{ij} = 1) \text{ if } p(\tilde{\theta}_{ij} = 1) > p(\tilde{\theta}_{ij} = 0) \tag{1}$$

where $p(\tilde{\theta}_{ij} = 1)$ represents the predicted probability that $\tilde{\theta}_{ij} = 1$. $\tilde{\theta}_{ij}$ denotes the predicted trust score form w_i to w_j. For t_j, the worker recruitment solution space Sp_j is obtained through social diffusion and calculated as follows:

$$Sp_j = \{W \cup w_v\} \forall w_v \in W_{sr}^u, \forall w_u \in W \text{ if } df_{uv} > rd \tag{2}$$

where $rd \in [0, 1]$ is defined as the random variable for social diffusion determination. W_{sr}^u represents the socially reachable set of w_u. Due to the impact of social diffusion probability, the social diffusion process becomes more difficult with higher diffusion layers.

Definition 5: Social Influence. The worker's social influence is a key factor affecting task utility. Through the MCS social network, workers could seek help from their social friends either directly or indirectly, thereby improving task utility. Specifically, a worker's social influence is determined by two factors: the number of socially reachable workers and the social diffusion probability. The social influence of w_i, denoted as si_i, is calculated as follows:

$$si_i = \sum_{w_j \in W_{sr}^i} df_{ij} \tag{3}$$

Definition 6: Preference Similarity. The preference similarity between workers and tasks is crucial for task utility. Higher preference similarity means workers are more willing and better able to complete tasks, resulting in greater task utility, benefiting both parties. Therefore, SoTR framework, based on GATs, is designed to precisely extract preference features between workers and tasks, as detailed in Section 4. After the release of t_j, the predicted preference rating r'_{ij} is obtained based on the preference features of w_i and t_j, and mapped as the preference similarity of w_i for t_j. The preference similarity, denoted as p_{ij}, is calculated as follows:

$$p_{ij} = W_{MLP}[\mathbf{h}_i^w \otimes \mathbf{h}_j^t]_{MLP} \quad (4)$$

where $\otimes$ represents the concatenation operation. $\mathbf{h}_i^w$ represents the preference vector of w_i, and $\mathbf{h}_j^t$ represents the preference vector of t_j. W_{MLP} represents the weight matrix of the Multi-Layer Perceptron (MLP).

Definition 7: Task Response Ability. A worker's task response ability is a key factor affecting task utility. Strong task response ability means that workers could quickly arrive at the task location and execute the task. Therefore, a worker's task response ability is closely related to the distance between the worker and the task as well as the worker's mobility speed. The task response ability of w_i to t_j, denoted as tr_{ij}, is calculated as follows:

$$tr_{ij} = \frac{\max(tm) - \frac{dist_{ij}}{v_i}}{\max(tm) - \min(tm)} \quad (5)$$

where $\frac{dist_{ij}}{v_i}$ represents the task response time of w_i for t_j. $dist_{ij}$ represents the distance between w_i and t_j. v_i represents the mobility speed of w_i. $\max(tm)$ represents the longest task response time among the workers in Sp_j, while $\min(tm)$ represents the shortest task response time among the workers in Sp_j.

Definition 8: Task Utility. After the release of t_j, the task utility Tu_j is calculated by comprehensively considering the social influence si_i, preference similarity p_{ij}, task response ability tr_{ij} of $w_i \in Sp_j$, and the recruited worker scale $|W_j|$. The calculation is as follows:

$$Tu_j = |W_j| \sum_{i \in W_j} \Lambda_{ij}, W_j \subseteq Sp_j \quad (6)$$

$$\Lambda_{ij} = \xi \cdot si_i \cdot p_{ij} \cdot tr_{ij} \quad (7)$$

where ξ represents the value coefficient, W_j represents the set of recruited workers for t_j, and Λ_{ij} represents the value of w_i for t_j.

Definition 9: Reward. Considering the energy consumption of $w_i \in W_j$ during the execution of t_j, the reward cr_{ij} paid by t_j to w_i is calculated as follows:

$$cr_{ij} = \gamma \cdot dist_{ij} \quad (8)$$

where γ represents the distance reward coefficient.

Definition 10: Incentive Reward. The social incentive mechanism based on the economic theory of decreasing rewards is designed, taking into account the dynamic changes in the number of workers. Initially, higher incentive rewards are paid by the MCS platform to those who successfully achieve social recruitment when the number of workers in W is low. Social recruitment is defined as workers inviting their social friends to join the MCS platform and be recruited for tasks. As the number of workers in W increases, new workers are encouraged to join the MCS platform by the same-side network effect in the MCS social network [24]. Subsequently, the MCS platform gradually reduces the incentive rewards to promote earlier participation in social recruitment, with later participation yielding lower rewards. Overall, the worker base of the MCS platform is rapidly expanded by this mechanism. Specifically, after t_j is released, the dynamic incentive reward ci_{ij} for w_i who successfully achieve social recruitment is calculated as follows:

$$ci_{ij} = \kappa \cdot \sum_{w_u \in \{W_j \cap W_{sr}^i\}} \frac{\Lambda_{uj}}{|W|} \quad (9)$$

where κ represents the incentive reward coefficient.

B. Problem Formulation

Given $\forall t_j \in T$, the task utility Tu_j is calculated by considering the preference similarity p_{ij}, social influence si_i, task response ability tr_{ij} of each worker $w_i \in Sp_j$, and the recruited worker scale $|W_j|$. Under the budget β_j constraint, workers are recruited for t_j to maximize Tu_j. Formally, the WRSD problem is formulated as a CMCO problem as follows:

$$\text{Maxmize} \quad \sum_{j \in T} Tu_j \quad (10)$$

$$\text{s.t.} \sum_{i \in W_j} cr_i \leq \beta_j, \forall t_j \in T, W_j \subseteq Sp_j \quad (11)$$

The WRSD problem is NP-Hard.

Proof: In the WRSD problem framework, there exists a series of items (corresponding to different workers), each with a certain value (corresponding to the worker's value) and a certain weight (corresponding to the worker's reward). Each knapsack (corresponding to a task) has a certain weight capacity (representing the task's budget). The objective is to select a combination of items that brings the maximum value without exceeding the knapsack's weight limit. Given that the knapsack problem is known to be NP-hard, the WRSD problem could be transformed into the knapsack problem using polynomial time reduction approaches, thereby confirming that the WRSD problem is also NP-hard.

IV. PROPOSED FRAMEWORK

SoTR and SoTP frameworks, based on GATs and GCNs, respectively, are shown in Fig. 3a and Fig. 3b, respectively.

A. SoTR Framework

SoTR framework comprises four components: 1) The embedding layer for initializing worker embeddings and task embeddings. 2) The worker modeling layer for learning latent features of workers. 3) The task modeling layer for learning latent features of tasks. 4) The prediction layer for predicting workers' preference ratings for tasks to learn model parameters.

1) Embedding Layer: The data in the social task recommendation system includes social data between workers and interaction data between workers and tasks. These data are modeled as a MCS social network G and a worker-task interaction hypergraph G'. The MCS social network captures social relationships between workers, where mutually trusted workers tend to exhibit similar task preferences, reflected in

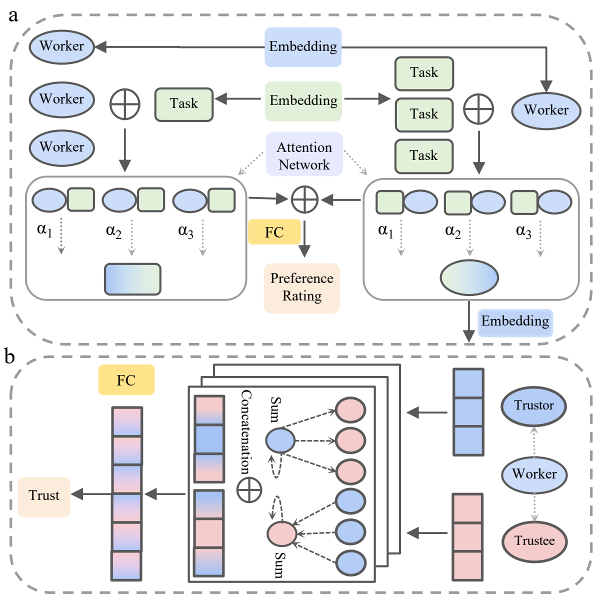


Fig. 3: Illustration of SoTR and SoTP Frameworks.

their closely aligned preference ratings for tasks. To this end, the Node2Vec pre-training method is employed to extract local and global trust structure features of workers in G, which are then used as the initial embeddings for $w_i \in V_w$ in the SoTR framework, denoted as $e_i \in R^{D_e \times 1}$. Additionally, random embeddings are used to generate initial embeddings for $t_j \in V_t$, denoted as $e_j \in R^{D_e \times 1}$.

2) Task Modeling Layer: Since G' encompasses both interactions between tasks and workers as well as preference ratings, the modeling of task interaction awareness x_{ji} must account for both interaction features and preference rating features. Specifically, x_{ji} is formulated by fusing the interaction information from $t_j \in V_t$ to $w_i \in I(t_j)$ with the preference rating r_{ji}, with its calculation detailed as follows:

$$\mathbf{x}_{ji} = g_t \cdot \left[e_i \otimes e_{r_{ji}}\right] \tag{12}$$

where e_i represents the embedding vector of w_i, and $e_{r_{ji}}$ represents the embedding vector of r_{ji}. g_t represents the weight matrix of MLP for task interaction. Next, the worker aggregation is employed to learn the task preference vector $\mathbf{h}_j^t$ for $t_j \in V_t$. Specifically, worker aggregation for t_j is performed by weighted aggregation of x_{ji}, where $w_i \in I(t_j)$, to better learn $\mathbf{h}_j^t$. Mathematically, the worker aggregation for t_j to obtain $\mathbf{h}_j^t$ is calculated as follows:

$$\mathbf{h}_j^t = \sigma\left(\mathbf{W}_{agg}^t \cdot \left\{\sum_{w_i \in I(t_j)} \frac{\exp\left(\alpha_{ji}\right)}{\sum\limits_{w_i \in I(t_j)} \exp\left(\alpha_{ji}\right)} x_{ji}\right\} + \mathbf{b}\right) \tag{13}$$

$$\alpha_{ji} = \mathbf{W}_2 \cdot \sigma\left(\mathbf{W}_1 \cdot \left[x_{ji} \otimes e_j\right] + \mathbf{b}_1\right) + \mathbf{b}_2 \tag{14}$$

where α_{ji} represents the attention score of x_{ji}, parameterized by a two-layer neural network. e_j represents the embedding vector of t_j. $\mathbf{W}_{agg}^t$, $\mathbf{W}_1$, and $\mathbf{W}_2$ represents the trainable parameters of the SoTR model. $\mathbf{b}_1$ and $\mathbf{b}_2$ represents the trainable bias terms of the SoTR model.

3) Worker Modeling Layer: By introducing the worker interaction awareness x_{ij} form $w_i \in V_w$ to $t_j \in I(w_i)$, the interaction process from w_i to t_j is formulated. x_{ij} integrates the interaction information from w_i to t_j with the preference rating r_{ij}, calculated as follows:

$$x_{ij} = g_w \cdot \left[e_j \otimes e_{r_{ij}}\right] \tag{15}$$

where $e_{r_{ij}}$ represents the embedding vector of r_{ij}, and g_w represents the weight matrix of MLP for worker interaction. Next, task aggregation is used to learn the preference vector $\mathbf{h}_i^w$ of w_i. Specifically, task aggregation for w_i is performed by weighted aggregation of x_{ij}, where $t_j \in I(w_i)$, to better learn $\mathbf{h}_j^t$. Mathematically, the task aggregation for w_i to obtain $\mathbf{h}_i^w$ is calculated as follows:

$$\mathbf{h}_i^w = \sigma\left(\mathbf{W}_{agg}^w \cdot \left\{\sum_{t_j \in I(w_i)} \frac{\exp\left(\alpha_{ij}\right)}{\sum\limits_{t_j \in I(w_i)} \exp\left(\alpha_{ij}\right)} \mathbf{x}_{ij}\right\} + \mathbf{b}\right) \tag{16}$$

$$\alpha_{ij} = \mathbf{W}_2 \cdot \sigma\left(\mathbf{W}_1 \cdot \left[\mathbf{x}_{ij} \otimes \mathbf{e}_i\right] + \mathbf{b}_1\right) + \mathbf{b}_2 \tag{17}$$

where α_{ij} represents the attention score of x_{ij}, parameterized by a two-layer neural network.

4) Prediction Layer: After aggregation, the preference vectors of workers and tasks (*i.e.*, $\mathbf{h}_i^w, \forall w_i \in V_w$ and $\mathbf{h}_j^t, \forall t_j \in V_t$) are obtained, concatenated, and then input into an MLP for preference rating prediction:

$$h_0 = \left[\mathbf{h}_i^w \otimes \mathbf{h}_j^t\right] \tag{18}$$

$$h_l = \sigma\left(\mathbf{W}_l \cdot h_{l-1} + \mathbf{b}_l\right) \tag{19}$$

$$r'_{ij} = \mathbf{W}_p \cdot h_l \tag{20}$$

where l denotes the number of hidden layers. r'_{ij} represents the predicted preference rating of w_i for t_j. $\mathbf{W}_l$, $\mathbf{W}_p$ denotes the trainable weight matrices of MLP, and $\mathbf{b}_l$ represents the trainable bias terms of MLP. To update model parameters, the mean squared error loss function L' is used as the objective function, defined as follows:

$$L' = \frac{1}{|O|} \sum_{i,j \in O} \left(r'_{ij} - r_{ij}\right)^2 \tag{21}$$

where $|O|$ is the number of observed preference ratings. r_{ij} denotes the ground truth preference rating of w_i for t_j.

B. SoTP Framework

SoTP framework consists of four parts: 1) the embedding layer for initializing worker embedding; 2) the trust aggregation layer to learn trust features of workers; 3) the trusted aggregation layer to learn the trusted features of workers; 4) the prediction layer to learn model parameters by predicting social relationships between workers.

1) Embedding Layer: The initial embedding of each $w_u \in V_w$ in SoTP model is obtained through pre-training on SoTR model, denoted as $\mathbf{h}_u^w \in R^{D_e \times 1}$. This ensures that the initial embeddings of workers contain task preference features, enriching the trust features of workers during the training of SoTP model. Specifically, the initial trust vector $h_O^0[u]$ and initial trusted vector $h_I^0[u]$ of $w_u \in V_w$ are given as follows:

$$h_O^0[u] = \mathbf{h}_u^w \tag{22}$$

$$h_I^0[u] = \mathbf{h}_u^w \tag{23}$$

2) Trust Aggregation Layer: Due to the asymmetric nature of social relationships, workers could play both the roles of trustor and trustee [15], [25]. First, considering workers as trustors propagating trust, the trust interaction vector At_{uv}^l of w_u trusting w_v is calculated as follows:

$$\mathrm{At}_{uv}^l = h_O^{l-1}[v] \otimes \left\{ W_{uv}^l \cdot \theta_{\mathrm{uv}} \right\} \tag{24}$$

where l represents the propagation layers. W_{uv}^l denotes the trainable weight matrix of the trust interaction at layer l. Further, trust aggregation is used to learn the trust features of workers as trustors. Specifically, trust aggregation is used to calculate the degree to which a worker trusts other workers. The more a worker trusts others, the higher their trust degree. Mathematically, the trust aggregation to obtain the worker trust vector h_O^l is calculated as follows:

$$h_O^l = \sigma\left(W_O^l\left(\widetilde{A}At^l\right) + b_O^l\right) \tag{25}$$

where W_O^l denotes the trainable weight matrix of the trust aggregation at layer l, b_O^l represents the trainable bias term of the trust aggregation at layer l, and $\widetilde{A}$ represents the adjacency matrix.

3) Trusted Aggregation Layer: Considering workers as trustees propagating trust, the trusted interaction vector Pt_{vu}^l of w_u trusted by w_v is calculated as follows:

$$\mathrm{Pt}_{vu}^l = h_I^{l-1}[v] \otimes \left\{ W_{vu}^l \cdot \theta_{vu} \right\} \tag{26}$$

where W_{vu}^l denotes the trainable weight matrix of the trusted interaction at layer l. Trusted aggregation is used to learn the trusted features of workers as trustees. Trusted aggregation is used to calculate the degree to which a worker is trusted by other workers. The more a worker is trusted by others, the higher their trusted degree. Mathematically, the trusted aggregation to obtain the worker trusted vector h_I^l is calculated as follows:

$$h_I^l = \sigma\left(W_I^l\left(\widetilde{A}^T Pt^l\right) + b_I^l\right) \tag{27}$$

where W_I^l denotes the trainable weight matrix of the trusted aggregation at layer l, and b_I^l represents the trainable bias term of the trusted aggregation at layer l.

4) Prediction Layer: The trust vector $h_O^l[u]$ of $w_u \in V_w$ at layer l and the trusted vector $h_I^l[v]$ of $w_v \in V_w$ at layer l are concatenated, and then input into a Fully Connected (FC) layer for predicting the social relationship, the predict trust score $\tilde{\theta}_{uv}$ of social relationship is calculated as fellows:

$$\tilde{\theta}_{uv} = \sigma\left(W_{fc} \cdot (h_O^l[u] \otimes h_I^l[v])\right) \tag{28}$$

where W_{fc} represents the FC matrix. Then, cross-entropy loss is utilized as the optimization objective to train the model parameters, quantifying the discrepancy between the predicted trust score and the ground truth trust score. The loss function L is defined as follows:

$$L = -\frac{1}{|\Omega|}\mathrm{CE}(\theta, \tilde{\theta}) + \lambda|\Theta|_2^2 \tag{29}$$

where CE represents cross-entropy function. $|\Omega|$ represents the number of observed trust scores. Θ represents the SoTP model parameters. λ represents the weight of the regularization term.

V. Worker Recruitment: Proposed Approach

After the tasks are released by task publishers, the social diffusion process of workers towards the task information is modeled based on their social relationships, thereby obtaining the worker recruitment solution space for tasks. Next, the preference similarity of workers towards tasks and their social influence are calculated using the trained SoTR and SoTP models. Subsequently, the task response ability of workers is determined by considering their mobility speeds and the distances between workers and tasks. The WRSD problem is then modeled as a CMCO problem by comprehensively considering preference similarity, social influence, task response ability, and recruited worker scale. Finally, the WRSD problem is solved using the VNTS algorithm, and workers achieving social recruitment are paid incentive rewards to promote their active participation in task information diffusion. The specific steps are as follows:

Step 1:Social Diffusion. For each $t_j \in T$, df_{uv} of between $w_u \in W$ and $w_v \in W_{sr}^u$ is calculated using Eq. (1). A random variable $rd \in [0, 1]$ is iteratively generated, and social diffusion is performed using Eq. (2) to obtain Sp_j for t_j.

Step 2:WRSD Problem Modeling. For each $w_i \in Sp_j$, p_{ij} and si_i are obtained using the trained SoTP and SoTR models with Eq. (3) and Eq. (4). tr_{ij} is calculated using Eq. (5). Tu_j is calculated using Eq. (6), and the WRSD problem is modeled as a CMCO problem using Eq. (10) and Eq. (11).

Step 3: Solving WRSD Problem with VNTS Algorithm. A specific VNTS algorithm is proposed to solve the WRSD problem, aiming to maximize task utility by recruiting an approximately optimal workers solution under budget constraints. The VNTS algorithm steps are as follows:

Substep 1: Initialization. Calculate Λ_{ij} using Eq. (7), cr_{ij} using Eq. (8), and $\frac{\Lambda_{ij}}{cr_{ij}}$ for all $w_i \in Sp_j$. Sort Sp_j in descending order by $\frac{\Lambda_{ij}}{cr_{ij}}$. Initialize the neighboring solutions set $W_j^{\text{neighbor}} = \{\}$.

Substep 2: Updating. Add $w_i \in Sp_j$ to W_j^{initial} in order until $\sum_{w_i \in W_j^{\text{initial}}} cr_{ij} \leq \beta_j$, where W_j^{initial} represents the initial worker solution. Set $W_j^{\text{current}} = W_j^{\text{initial}}$, where W_j^{current} represents the current worker solution. Update W_j^{current} to the tabu list L_m. Calculate $Tu_j(W_j^{\text{current}})$ of W_j^{current}.

Substep 3: Neighborhood Search. The n-th neighborhood ($n \in [1, |W_j^{\text{current}}|]$) is as follows: (a) Initialize $L_n = \{\}$ and set $W_j^{\text{current}_n} = W_j^{\text{current}}$, where $W_j^{\text{current}_n}$ represents the current worker solution for the n-th neighborhood, and L_n represents the tabu sublist for the n-th neighborhood. (b) Set $Sp_j^n = Sp_j \setminus W_j^{\text{current}_n}$, where $\setminus$ represents the set difference operation, and Sp_j^n represents the worker recruitment solution space in the n-th neighborhood. (c) Set $S_n^{\text{incomplete}} = W_j^{\text{current}_n} \setminus \{\max_n(cr_{ij}), w_i \in W_j^{\text{current}_n}\}$, where $S_n^{\text{incomplete}}$ represents the incomplete solution for the n-th neighborhood, and $\{\max_n(cr_{ij}), w_i \in W_j^{\text{current}_n}\}$ represents the set of top n workers in terms of cr_{ij} from $W_j^{\text{current}_n}$. (d) Update $\beta_j = \beta_j - \sum_{w_i \in S_n^{\text{incomplete}}} cr_{ij}$ and set $Sp_j^n = \{w_i \mid w_i \in Sp_j^n, cr_{ij} \leq \beta_j\}$. (e) Add $w_i \in Sp_j^n$ to $S_n^{\text{incomplete}}$ in order until $\sum_{w_i \in S_n^{\text{incomplete}}} cr_{ij} \leq \beta_j$. Set $S_n = S_n^{\text{incomplete}}$, S_n represents the solution for the n-th neighborhood. If $S_n \in L_n$ then update $Sp_j^n = Sp_j^n \setminus Sp_j^n[|\text{repeat}(S_n)|]$ and repeat step (e), where $|\text{repeat}(S_n)|$ represents the number of times S_n is repeated in L_n. $Sp_j^n[|\text{repeat}(S_n)|]$ represents the top $|\text{repeat}(S_n)|$ elements in terms of $\frac{\Lambda_{ij}}{cr_{ij}}$ in Sp_j^n. Otherwise, update $W_j^{\text{neighbor}} = W_j^{\text{neighbor}} \cup S_n$ and set $W_j^{\text{current}_n} = S_n$. Repeat steps (b)-(e) until the maximum iteration count $Iterations_{\text{max}}^n$ for the n-th neighborhood is exceeded.

Substep 4: Neighborhood Evaluation and Update. Set $W_{j_{\max}}^{\text{neighbor}} = \arg\max(Tu_j(S), \forall S \in W_j^{\text{neighbor}})$, where $W_{j_{\max}}^{\text{neighbor}}$ represents the neighborhood solution with the highest task utility. If $Tu_j(W_{j_{\max}}^{\text{neighbor}}) > Tu_j(W_j^{\text{current}})$ and $W_{j_{\max}}^{\text{neighbor}} \notin L_m$, then set $W_j^{\text{current}} = W_{j_{\max}}^{\text{neighbor}}$ and add $W_{j_{\max}}^{\text{neighbor}}$ to L_m. Otherwise, set $Sp_j = Sp_j \setminus Sp_j[0]$. Iterate back to the substep 2 until the maximum iteration count $Iterations_{\text{max}}$ is exceeded. After the iteration ends, set $W_j = W_j^{\text{current}}$, $W = W \cup W_j$.

Step 4:Payment and Social Incentive.

Substep 1: Payment. For each $t_j \in T$, calculate cr_{ij} using Eq. (8) for all $w_i \in W_j$. Task publishers then pay cr_{ij} to each $w_i \in W_j$.

Substep 2: Social Incentive. Identify each $w_u \in T_j$ who is socially recruited. Calculate ci_{ij} using Eq. (9) for $w_i \in W$ who diffused t_j to w_u. The platform then pays ci_{ij} to w_i.

VI. PERFORMANCE EVALUATION

This section primarily introduces the experimental setup, including datasets, baselines, metrics, parameters, and evaluation results. The hardware configuration for the experiment includes an Intel Core i5-12490F CPU, an NVIDIA GeForce RTX 4060Ti GPU with 8GB VRAM, and 32GB RAM.

A. Datasets

Extensive experiments are conducted in this paper based on three real-world datasets: Ciao, Epinions, and FilmTrust. These datasets are sourced from popular social websites Ciao (http://www.ciao.co.uk), Epinions (www.epinions.com), and FilmTrust. Each dataset contains a large volume of user preference ratings for items and social information among users. User preference ratings for items are mapped to worker preference ratings for tasks, and user social information is mapped to MCS social network. The preference ratings range from 0 to 5. Detailed statistical information about these datasets is provided in Table I.

TABLE I: Statistics of the datasets

Dataset	Ciao	Epinions	FilmTrust
Users	7,317	18,088	1,508
Items	104,975	261,649	2,071
Preference Ratings	283,319	764,351	35,497
of Density(Preference Ratings)	0.0368%	0.0161%	1.14%
Trust Scores	111,781	335,813	1,853
of Density(Trust Scores)	0.2087%	0.1087%	0.0688%

TABLE II: Statistics of SoTP and SoTR Models Performance

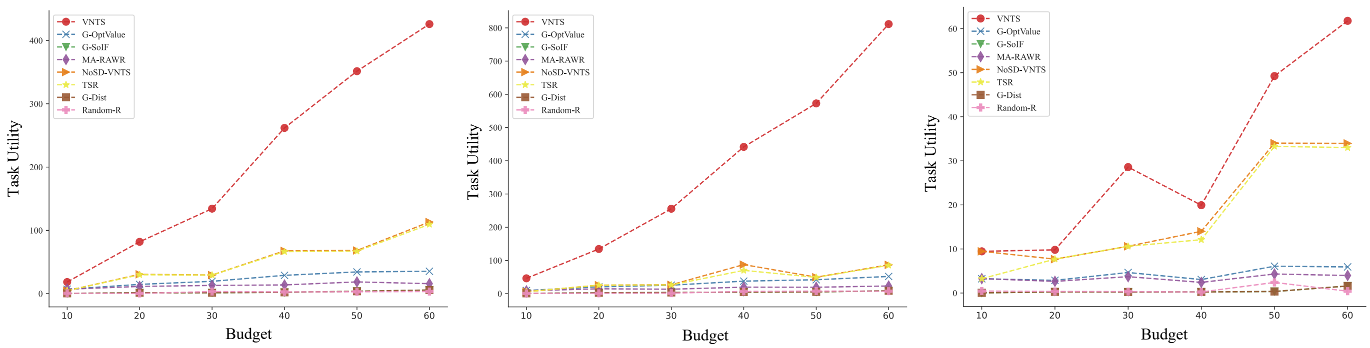
Datasets	Ciao	Epinions	FilmTrust
Acc(SoTP)	90.2%	93.4%	73.5%
F1-Score(SoTP)	82.7%	86.0%	69.2%
MAE(SoTR)	0.8179	0.9499	0.7099
RMSE(SoTR)	1.0426	1.1790	0.9321

B. Baselines

TSR Algorithm (TMC 2024) [15]: Adjusted for the WRSD problem under budget constraints, the algorithm utilizes a tabu list and heuristic neighborhood strategy, dynamically adjusting the solution structure driven by the value-to-reward ratio to seek near-optimal solutions. **MA-RAWR** Algorithm (TMC 2021) [9]: Adjusted to solve the WRSD problem under budget constraints, worker solution chromosomes are divided into high, medium, and low segments based on workers' values, employing differential evolution and improved neighborhood structures to find near-optimal solutions. Greedy Nearest Distance Algorithm (**G-Dist**) [20]: Selects the group of workers closest to the task under budget constraints. Greedy Optimal Value-to-Reward Ratio Algorithm (**G-OptValue**): Under budget constraints, workers with the highest value-to-reward ratio are recruited through this algorithm. Greedy Social Influence Algorithm (**Greedy-SI**): Under budget constraints, workers with the highest social influence are recruited through this algorithm. Non-Social Diffusion VNTS (**NoSD-VNTS**): VNTS algorithm without social diffusion. Random Recruitment Algorithm (**Random-R**): Under budget constraints, workers are randomly recruited in multiple rounds.

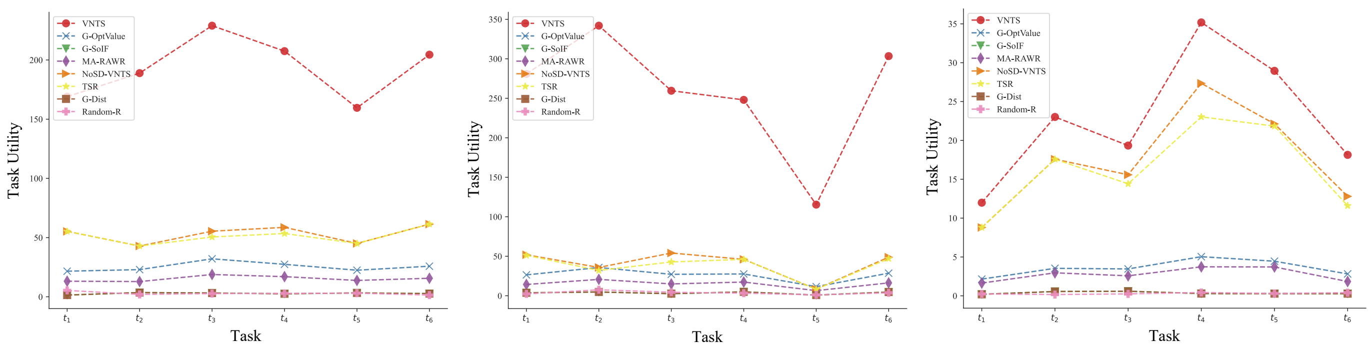
C. Metrics

F1-Score: A metric that combines precision and recall to evaluate the performance of a classification model. **Acc:** The ratio of correctly classified samples to the total number of samples is indicated by the classifier. **MAE:** The average absolute error between the predicted values and the actual values is measured in a regression model. **RMSE:** The evaluation error is reflected by calculating the square root of the mean of the squared differences between the predicted and actual values.



(a) Ciao (b) Epinions (c) FilmTrust

Fig. 4: Task Utility of Different Algorithms Under Various Budgets



(a) Ciao (b) Epinions (c) FilmTrust

Fig. 5: Task Utility of Different Algorithms Under Various Tasks

Task Utility: The quality of the worker solution recruited for the task. **NoW:** The number of workers after social diffusion.

D. Parameter Settings

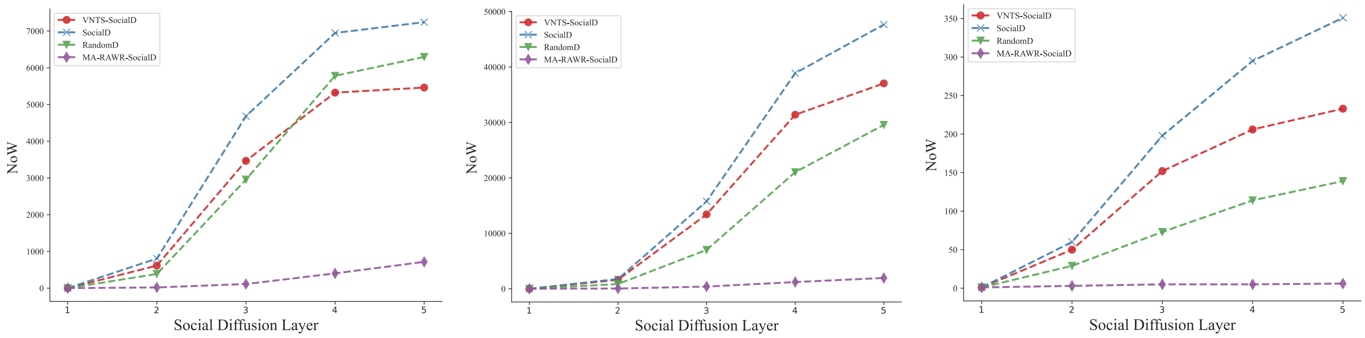
In this paper, SoTR and SoTP are implemented using PyTorch. SoTR generates 64-dimensional worker and task embeddings using Node2Vec and random initialization, respectively. The model uses a batch size of 256, with a learning rate of 0.001, a decay factor of 0.1, and a decay step of 30. All neural components use three hidden layers. For each dataset, SoTR uses 80% of the data for training, 10% for validation, and 10% for testing. SoTP pre-trained with SoTR, produces 64-dimensional initial worker embeddings for trustors and trustees. The learning rate is 0.01, and the output dimensions of the trust convolution layers are [64, 64, 64]. SoTP uses 80% of the data for training and 20% for testing. For the VNTS algorithm, $Iterations^{n}_{\text{max}}(n \in [1, |W_j^{\text{current}}|])$ is set to 100, $Iterations_{\text{max}}$ to 50, and the tabu list size is adaptively set. The default number of social diffusion layers is 1. $|W|$ is set to 50, with task and worker locations randomly generated within Manhattan, New York City. Worker mobility speeds range from 4 km/h to 40 km/h. ξ is set to 0.02. γ and κ are set to 1.The maximum social influence per worker is 100. Parameter settings for TSR and MA-RAWR are referenced from [15] and [9].

E. Evaluation Results: Evaluation performance of SoTR and SoTP models

To realistically evaluate the performance of SoTR and SoTP models in predicting preference ratings and trust scores, experiments are conducted on three real datasets. SoTP model demonstrated excellent performance in terms of F1-Score and Acc, while SoTR model exhibited outstanding performance in terms of MAE and RMSE.The specific evaluation results are shown in Table II. It is noteworthy that trust prediction with SoTP model performs poorly on the FilmTrust dataset. The low density, sparse data, and unbalanced distribution of the FilmTrust social network prevent SoTP model from adequately learning the underlying patterns and features of worker trust. In contrast, the SoTR model demonstrated excellent performance in preference rating prediction on the FilmTrust dataset. With its focus on social recommendation networks and higher preference rating density, FilmTrust provided clearer and more effective data for the SoTR model, facilitating improved learning of worker-task preference relationships.

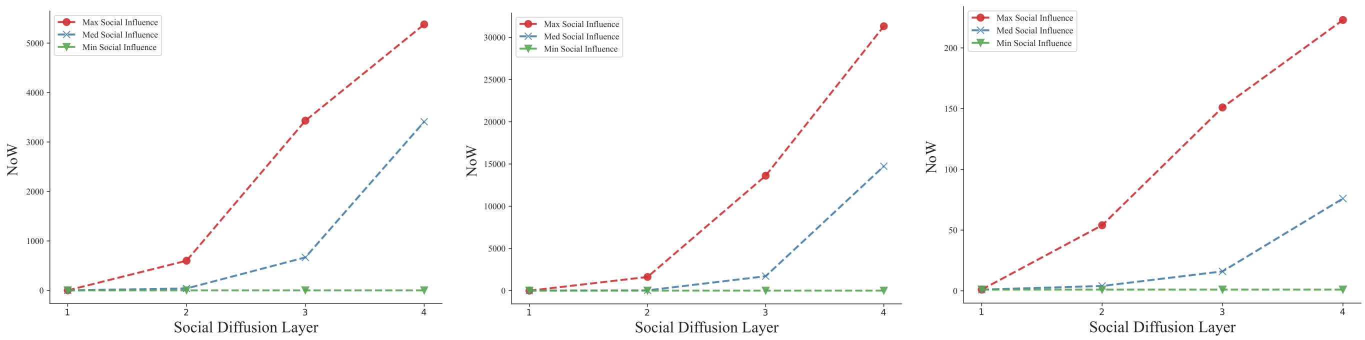
F. Evaluation Results: Task Utility of Different Algorithms Under Various Budgets

The evaluation is conducted on three real datasets to evaluate the task utility of different algorithms under varying budgets, ranging from 10 to 60 with a step size of 10. Fig. 4



(a) Ciao (b) Epinions (c) FilmTrust

Fig. 6: NoW of Different Algorithms Under Various Social Diffusion Layers



(a) Ciao (b) Epinions (c) FilmTrust

Fig. 7: NoW of VNTS with Different Social Influences Across Various Social Diffusion Layers

shows each algorithm's task utility across varying budgets. The proposed VNTS algorithm consistently outperforms others, due to its comprehensive consideration of workers' preference similarity, social influence, task response ability, recruited worker scale and an effective heuristic neighborhood search strategy. Task utility also rises with larger budgets, as they enable the selection of more high-quality workers. In contrast, MA-RAWR performs poorly in solving the WRSD problem due to its limited neighborhood strategy. Worker solutions are categorized into high, medium, and low segments based on value, focusing on crossover mutation within the high and medium segments. This leads to minimal changes in neighborhood solutions and inefficient exploration of the search space.

G. Evaluation Results: Task Utility of Different Algorithms Under Various Tasks

The task utility of different algorithms was validated across various tasks using three real datasets, with six tasks released sequentially. Fig. 5 shows that the proposed VNTS algorithm outperforms others in task utility across different tasks. However, VNTS's performance varies across datasets due to differences in the number of social friends among workers. In the FilmTrust dataset, which has a small-scale social network, the limited number of social diffusion friends results in minimal differences in recruitment solutions before and after diffusion. Thus, VNTS shows only slight improvement over NoSD-VNTS, as seen in Fig. 5c. In contrast, for datasets with large-scale social networks, social diffusion enables more high-quality worker solutions, amplifying VNTS's advantages and significantly surpassing algorithms without social diffusion in task utility.

H. Evaluation Results: NoW of Different Algorithms Under Various Social Diffusion Layers

To compare the NoW of different diffusion models, the initial number of workers was set to 1. VNTS-SocialD and MA-RAWR-SocialD were established as the diffusion models for VNTS and MA-RAWR, respectively. As shown in Fig. 6, the NoW of VNTS-SocialD was found to rank second only to the optimal baseline SocialD as social diffusion layers increased. In contrast, SocialD and RandomD used simpler strategies, setting diffusion probabilities for all socially reachable friends to 1 and 0.5, respectively. VNTS-SocialD determined probabilities by capturing complex nonlinear relationships in the MCS social network and used a social incentive mechanism, making diffusion more realistic. MA-RAWR-SocialD defined its diffusion probability as the ratio of mutual to total friends, leading to lower efficiency and unsuitability for the WRSD problem based on social incentives.

I. Evaluation Results: NoW of VNTS with Different Social Influences Across Various Social Diffusion Layers

To distinctly compare the NoW trends of VNTS under different social influences as social diffusion layers increase, the initial number of workers was set to 1, with Max Social Influence, Med Social Influence, and Min Social Influence representing the initial worker having maximum, medium, and minimum social influence, respectively. As shown in Fig. 7, NoW increases with diffusion layers and correlates positively with social influence, with Max Social Influence performing best. This is because social influence correlates positively with diffusion probability and the number of socially reachable friends. Greater social influence implies a higher probability of spreading task information to more friends, with this advantage further amplified as diffusion layers increase. In contrast, when social influence is low, diffusion may be interrupted, causing NoW to drop to zero.

VII. Conclusion

The SoTR framework is designed to handle social recommendation data by extracting preference features through task and worker aggregation to predict workers' preference ratings for tasks. Following this, the SoTP framework handles social data by extracting worker trust and trusted features through trust and trusted aggregation, respectively, to predict social relationships among workers. The social diffusion process of workers towards tasks is then modeled and incentivized. Under budget constraints, the WRSD problem is modeled as a CMCO problem. A heuristic neighborhood search strategy is developed, and the VNTS algorithm is proposed to solve the WRSD problem, yielding an approximately optimal worker solution for tasks. Extensive evaluations on three real-world datasets demonstrate that the proposed approach outperforms state-of-the-art methods in the literature.

References

[1] Weilong Wang, Yingjie Wang, Peiyong Duan, Tianen Liu, Xiangrong Tong, and Zhipeng Cai. A triple real-time trajectory privacy protection mechanism based on edge computing and blockchain in mobile crowdsourcing. *IEEE Transactions on Mobile Computing*, 22(10):5625–5642, 2022.

[2] Yingjie Wang, Mingze Wang, Lingkang Meng, Qi Zhang, Xiangrong Tong, and Zhipeng Cai. New crowd sensing computing in space-air-ground integrated networks. In *2021 International Conference on Space-Air-Ground Computing (SAGC)*, pages 143–149. IEEE, 2021.

[3] Zice Sun, Yingjie Wang, Zhipeng Cai, Tianen Liu, Xiangrong Tong, and Nan Jiang. A two-stage privacy protection mechanism based on blockchain in mobile crowdsourcing. *International Journal of Intelligent Systems*, 36(5):2058–2080, 2021.

[4] Xinhao Li, Yingjie Wang, Bingyi Xie, Lingkang Meng, Zhaowei Liu, Xiangrong Tong, Ao Zhou, and Zhipeng Cai. Planning-based mobile crowdsourcing bidirectional multi-stage online task assignment. *Computer Networks*, 233:109879, 2023.

[5] Zhiyuan Wang, Lin Gao, and Jianwei Huang. Socially-optimal mechanism design for incentivized online learning. In *IEEE INFOCOM 2022-IEEE Conference on Computer Communications*, pages 1828–1837. IEEE, 2022.

[6] Qi Zhang, Yingjie Wang, Guisheng Yin, Xiangrong Tong, Akshita Maradapu Vera Venkata Sai, and Zhipeng Cai. Two-stage bilateral online priority assignment in spatio-temporal crowdsourcing. *IEEE Transactions on Services Computing*, 16(3):2267–2282, 2022.

[7] Zhuoran Lu, Yingjie Wang, Xiangrong Tong, Chunxiao Mu, Yu Chen, and Yingshu Li. Data-driven many-objective crowd worker selection for mobile crowdsourcing in industrial iot. *IEEE Transactions on Industrial Informatics*, 19(1):531–540, 2021.

[8] Sitong Chen, Xujia Zhao, Jiahao Liu, Guoju Gao, and Yang Du. Social-network-assisted task recommendation algorithm in mobile crowd sensing. In *Proceedings of the 7th international conference on information and education innovations*, pages 136–142, 2022.

[9] Liang Wang, Dingqi Yang, Zhiwen Yu, Qi Han, En Wang, Kuang Zhou, and Bin Guo. Acceptance-aware mobile crowdsourcinag worker recruitment in social networks. *IEEE Transactions on Mobile Computing*, 22(2):634–646, 2021.

[10] Sitong Chen, Xujia Zhao, Jiahao Liu, Guoju Gao, and Yang Du. Social-network-assisted task recommendation algorithm in mobile crowd sensing. In *Proceedings of the 7th International Conference on Information and Education Innovations*, pages 136–142, 2022.

[11] Jiangtian Nie, Jun Luo, Zehui Xiong, Dusit Niyato, Ping Wang, and Mohsen Guizani. An incentive mechanism design for socially aware crowdsensing services with incomplete information. *IEEE Communications Magazine*, 57(4):74–80, 2019.

[12] Stefano Chessa, Antonio Corradi, Luca Foschini, and Michele Girolami. Empowering mobile crowdsensing through social and ad hoc networking. *IEEE Communications Magazine*, 54(7):108–114, 2016.

[13] Ankur Mani, Iyad Rahwan, and Alex Pentland. Inducing peer pressure to promote cooperation. *Scientific reports*, 3(1):1735, 2013.

[14] Chao Gao and Jiming Liu. Network-based modeling for characterizing human collective behaviors during extreme events. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 47(1):171–183, 2016.

[15] 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.

[16] Aymen Hamrouni, Hakim Ghazzai, Turki Alelyani, and Yehia Massoud. Low-complexity recruitment for collaborative mobile crowdsourcing using graph neural networks. *IEEE Internet of Things Journal*, 9(1):813–829, 2021.

[17] Hengzhi Wang, Yongjian Yang, En Wang, Wenbin Liu, Yuanbo Xu, and Jie Wu. Truthful user recruitment for cooperative crowdsensing task: A combinatorial multi-armed bandit approach. *IEEE Transactions on Mobile Computing*, 22(7):4314–4331, 2022.

[18] Wenqi Fan, Yao Ma, Qing Li, Yuan He, Eric Zhao, Jiliang Tang, and Dawei Yin. Graph neural networks for social recommendation. In *The world wide web conference*, pages 417–426, 2019.

[19] Jiangtao Wang, Feng Wang, Yasha Wang, Daqing Zhang, Leye Wang, and Zhaopeng Qiu. Social-network-assisted worker recruitment in mobile crowd sensing. *IEEE Transactions on Mobile Computing*, 18(7):1661–1673, 2018.

[20] Mingjun Xiao, Jie Wu, Liusheng Huang, Ruhong Cheng, and Yunsheng Wang. Online task assignment for crowdsensing in predictable mobile social networks. *IEEE Transactions on Mobile Computing*, 16(8):2306–2320, 2016.

[21] Xiangwan Fu, Anfeng Liu, Neal N Xiong, Tian Wang, and Shaobo Zhang. Atwr-smr: an area-constrained truthful worker recruitment based sensing map recovery scheme for sparse mcs in extreme-environment internet-of-things. *IEEE Internet of Things Journal*, 2023.

[22] Hui Gao, Yu Xiao, Han Yan, Ye Tian, Danshi Wang, and Wendong Wang. A learning-based credible participant recruitment strategy for mobile crowd sensing. *IEEE Internet of Things Journal*, 7(6):5302–5314, 2020.

[23] Guoju Gao, Jie Wu, Mingjun Xiao, and Guoliang Chen. Combinatorial multi-armed bandit based unknown worker recruitment in heterogeneous crowdsensing. In *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*, pages 179–188. IEEE, 2020.

[24] Jiaping Xie, Weijun Zhu, Lihong Wei, and Ling Liang. Platform competition with partial multi-homing: When both same-side and cross-side network effects exist. *International Journal of Production Economics*, 233:108016, 2021.

[25] Raquel Urena, Gang Kou, Yucheng Dong, Francisco Chiclana, and Enrique Herrera-Viedma. A review on trust propagation and opinion dynamics in social networks and group decision making frameworks. *Information Sciences*, 478:461–475, 2019.