smart cities

MDPI

Article

A Spatio-Temporal Task Allocation Model in Mobile Crowdsensing Based on Knowledge Graph

Bingxu Zhao †, Hongbin Dong † and Dongmei Yang *

College of Computer Science and Technology, Harbin Engineering University, Harbin 150001, China; zbx_heu@hrbeu.edu.cn (B.Z.); donghongbin@hrbeu.edu.cn (H.D.)

* Correspondence: yangdongmei@hrbeu.edu.cn

† These authors contributed equally to this work.

Abstract: With the increasing popularity of wireless networks and the development of smart cities, the Mobile Crowdsourcing System (MCS) has emerged as a framework for automatically assigning spatiotemporal tasks to workers. The study of mobile crowdsourcing makes a valuable research contribution to community service and urban route planning. However, previous algorithms have faced challenges in effectively addressing task allocation issues with massive spatial data. In this paper, we propose a novel solution to the spatiotemporal task allocation problem using a knowledge graph. Firstly, we construct a robust spatiotemporal knowledge graph (STKG) and employ a knowledge graph embedding algorithm to learn the representations of nodes and edges. Next, we utilize these representations to build a task transition graph, which is a weighted and learning-based graph that highlights important neighbors for each task. We then apply a simplified Graph Convolutional Network (GCN) and an RNN-based model to enhance task representations and capture sequential transition patterns on the task transition graph. Furthermore, we design a similarity function to facilitate personalized task allocation. Through experimental results, we demonstrate that our solution achieves higher accuracy compared to existing approaches when tested on three real datasets. These research findings are significant as they contribute to an 18.01% improvement in spatiotemporal task allocation accuracy.

Keywords: smart city; mobile crowdsensing; task allocation; knowledge graph; GCN; RNN

Citation: Zhao, B.; Dong, H.; Yang, D. A Spatio-Temporal Task Allocation Model in Mobile Crowdsensing Based on Knowledge Graph. *Smart Cities* **2023**, *6*, 1937–1957. https://doi.org/10.3390/smartcities6040090

Academic Editor: Songnian Li

Received: 5 July 2023
Revised: 5 August 2023
Accepted: 8 August 2023
Published: 10 August 2023

1. Introduction

With the rapid development of smart cities, spatio-temporal crowdsensing has attracted more and more attention, which is a novel distributed service model that enables resource sharing and value creation by assigning many tasks to many mobile device workers. However, the problem of task allocation in spatiotemporal crowdsensing has been a major challenge since it directly determines the efficiency of task completion. Therefore, it is of great theoretical significance and practical value to study how to optimize spatio-temporal crowdsensing task allocation by using advanced technology [1].

In recent years, the development of deep learning techniques has provided new ideas and methods for studying the task allocation problem in spatiotemporal crowdsensing. Graph neural networks (GNNs) are deep learning methods that effectively handle graph data. By learning the relationships between nodes and local structural features, GNNs can model and optimize task allocation problems. As a result, spatiotemporal crowdsensing task allocation methods based on GNNs have become a popular research direction [2].

Currently, spatiotemporal task allocation problems [3] can be divided into two categories: sequence-based solutions and knowledge graph-based solutions. Sequence-based solutions encompass Markov chains, recurrent neural networks (RNNs), and attention-based models. Initial research employed Markov chains to model sequence transition patterns. RNN-based methods [4,5] aim to integrate classical RNN architectures, such

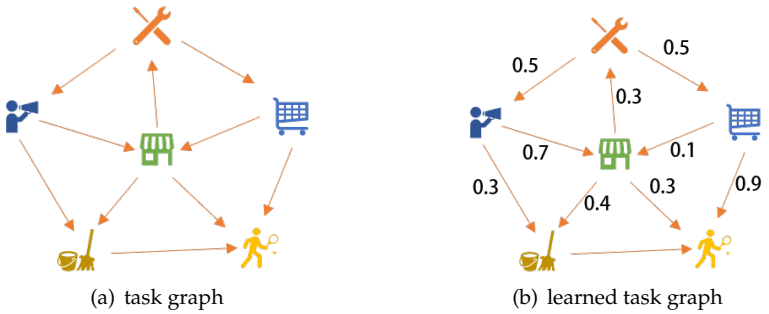
as Long Short-Term Memory networks, with temporal and spatial contexts to enhance the model's ability to capture sequence patterns. Moreover, attention mechanism models have been proposed for task allocation based on the success of Transformers in natural language processing. Recently, knowledge graph-based models have emerged to improve task representations by considering the characteristic that similar users tend to access similar tasks. However, knowledge graph-based spatiotemporal task allocation methods have certain limitations that significantly impact their effectiveness:

Knowledge graph construction: Previous methods primarily constructed custom homogeneous task graphs consisting of only one type of node and edge [6,7]. These methods primarily emphasized task connectivity while neglecting to consider edge weights. Conversely, a limited number of studies have explored the utilization of heterogeneous graphs that incorporate multiple types of nodes and edges. These studies aim to generate learning-based homogeneous task graphs.

Task representation: Current knowledge graph-based models directly employ graph neural networks (GNNs) on custom graphs to enhance the representation of each task through neighborhood sampling strategies. However, these models lack sufficient theoretical support. Additionally, avoiding unnecessary operations (e.g., feature transition) presents challenges in the application of GNNs [8]. Furthermore, it is crucial to clarify the learning process of weight coefficients, which signify the importance of neighbors for each task, as overlooking this aspect may undermine the effectiveness of the model.

Personalized allocation: The majority of existing methods combine user feature vectors, such as embeddings, with the model's output to allocate tasks in a personalized manner. However, this operation must consider users' general preferences for different tasks. Alternatively, other studies combine user and task representations and utilize them as input to the model to reflect personalized user preferences. However, accurately obtaining and incorporating personalized user preferences solely by concatenating user and task representations is challenging due to the multitude of factors that can influence user preferences, including time, location, and task category.

This paper introduces a knowledge graph-based algorithm for spatiotemporal task allocation to overcome the aforementioned limitations. Initially, we construct a robust and versatile spatiotemporal knowledge graph (STKG) to address the limitations of previous methods' custom homogeneous task graphs. By utilizing knowledge graph embedding (KGE) algorithms on the STKG, we learn representations for each node and edge. Next, these learned representations are used to construct a task transition graph, as depicted in Figure 1. Unlike previous methods, our task graph explicitly denotes the importance of different neighbors (excluding itself) for each task. Through the application of a simplified graph convolutional network (GCN) on the task transition graph, we enrich the representation of each task. Finally, the enhanced representations are inputted into an RNN-based model to effectively capture sequence transition patterns, thus leading to improved task allocation services. To tackle the third limitation, we design a similarity function that encompasses various factors, such as time, location, and task category, to measure the preferences of different users and facilitate personalized task allocation.



(a) task graph (b) learned task graph

Figure 1. A simple example that illustrates the difference between customized task graph and learned task transition graph.

In summary, we contribute to spatiotemporal task allocation in the following ways:

- We propose a novel spatiotemporal knowledge graph (STKG) that can be directly used to learn task graphs reflecting the patterns of task transitions. To our knowledge, this is the first method that learns weighted isomorphic task graphs using heterogeneous knowledge graphs in spatiotemporal crowdsensing.
- We introduce a novel Knowledge Graph-based GTA approach that seamlessly integrates the learned task transition graph with an RNN-based model to effectively capture sequence transition patterns. Furthermore, to enhance the effectiveness of personalized task allocation, we introduce a meticulously designed similarity function to measure the preferences of different users.
- We extensively evaluate our approach on three real datasets and demonstrate that GTA outperforms existing solutions with higher accuracy.

The structure of this paper is described as follows: First, we briefly introduce the basic concepts of spatiotemporal crowdsensing and task allocation and explore the current challenges and problems faced in this area in Section 1. Next, we review the research work related to spatiotemporal crowdsensing task allocation in Section 2. In Section 3, we describe the system model and the problem formulation. Subsequently, we describe our proposed knowledge graph-based spatiotemporal crowdsensing task allocation algorithm in detail in Section 4. In Section 5, we verify the effectiveness and feasibility of the proposed algorithm through experiments on three real datasets and discuss potential research directions and application prospects. Finally, we review the main contributions and shortcomings of this paper in the Section 7 and look forward to possible future research directions.

2. Related Works

Spatio-temporal crowdsensing has attracted considerable attention because of its high practicality for real-world applications. We have studied the online allocation of two types of objects in spatio-temporal crowdsensing. In this section, we will briefly review some of the work on task allocation and prediction, as shown in Table 1.

2.1. Task Allocation

In recent years, the problem of task allocation in spatio-temporal crowdsensing systems has become a research hotspot. Early research mainly focused on the aspect of offline task allocation. However, with the advancement of technology and the popularity of mobile devices, the timeliness of tasks is becoming more and more critical. Therefore, the importance and complexity of online task allocation problems in spatio-temporal crowdsensing systems have become increasingly prominent. To address these issues, researchers have proposed many new methods and frameworks. Zhao et al. [9,10] proposed a dynamic delayed binary matching (DDBM) problem and designed value-based task allocation (VBTA) and policy gradient-based task allocation (PGTA) frameworks, respectively. To achieve safe and efficient task allocation, Liu et al. [11] proposed a task allocation framework based on federated preference learning. The framework not only ensures the privacy of the centralized data on each platform but also enables task allocation based on worker preferences. Wang et al. [12] proposed a multi-task allocation problem with the goal of maximizing spatio-temporal coverage, aiming to solve the competition and prioritization problems among tasks while considering temporal and spatial constraints. These research results help us better understand and solve the task allocation problem of spatio-temporal crowdsensing systems, and provide more effective and feasible solutions for practical applications.

For heterogeneous task, Li et al. [13] proposed a worker selection problem with the goal of minimizing costs. Specifically, the problem involves selecting a group of workers suitable for a specific task in order to perform the task efficiently and minimize costs while these workers can continue to perform their daily work without changing the original trajectory. The study provides an effective solution for worker selection for heterogeneous sensing tasks. Li et al. [14] further formulate a time-constrained task allocation problem aimed at maximizing the utility of mobile crowdsensing platforms.

Time constraints are considered critical because tasks must be completed within strict time constraints. Their goal is to allocate multiple tasks to the right mix of workers to maximize the utility of the overall platform, thereby increasing system efficiency and productivity. Wang et al. [15] proposed a new task allocation framework from the perspective of task organizers and participants to optimize the overall system utility. The framework aims to address challenges in task allocation, such as unfair distribution, full load of workers, etc. They propose a new task allocation algorithm that considers multiple factors between workers and tasks to maximize the utility of the overall system. The framework helps to optimize task allocation and improve productivity and efficiency while allowing workers to complete mobile crowdsensing tasks in their daily work without changing the original trajectory, improving work flexibility.

2.2. Mobility Prediction

Movement prediction is widely studied in task allocation in spatio-temporal crowdsensing systems. Zhang et al. [16] proposed a novel multi-task allocation method based on mobility prediction, which optimizes the overall task completion rate according to the mobility prediction results while considering the spatio-temporal characteristics of workers and tasks. Yang et al. [17] used a Markov model to predict the probability of a worker completing a task and select a worker set that maximizes the probability of completing multiple tasks while satisfying a budget constraint. The work aims to optimize the task allocation scheme, thereby improving overall efficiency. Wang et al. [18] proposed a multi-objective optimization algorithm using a mobile prediction model, including task coverage maximization and task cost minimization. The model simulates worker flow using statistical models, improving the accuracy and efficiency of task allocations. At the same time, they also take into account the mobility of workers, increasing the flexibility of task allocation. Wang et al. [19] proposed a multi-task allocation algorithm to maximize spatio-temporal coverage under the total budget constraint. The algorithm exploits the Poisson distribution to predict task completion and optimize task allocation to maximize system utility. The work is optimized for overall benefit, taking into account budgetary constraints and task completion. Wang et al. [19] used the poisson distribution to obtain the probability of workers passing through a certain area within a certain period of time and proposed an effective task allocation algorithm. The algorithm aims to optimize task allocation to improve overall efficiency. Through the application of the movement prediction model, this algorithm can more accurately predict the movement trajectory of workers, thereby improving the accuracy and efficiency of task allocation.

Poisson distribution is widely used to predict user mobility in spatio-temporal crowdsensing task allocation. Wang et al. [20] and Xiao et al. [21] used poisson distribution to predict the user's moving trajectory to improve the accuracy and efficiency of task allocation. Guo et al. [22] proposed the task allocation problem in two situations, namely, time-sensitive tasks and delay-tolerant tasks. The latter aims to reduce the workload of employees by predicting the mobility of users to allocate tasks. The work is mainly to consider the mobility of employees and improve the flexibility and accuracy of task allocation. Liu et al. [23] used a Markov model to predict worker mobility and select a set of workers to perform tasks. The algorithm aims to optimize the task allocation scheme, thereby improving system efficiency and work efficiency. Lai et al. [24] proposed a DSTA model that aims to maximize the number of completed tasks under a specific perceived duration requirement. The model models the probability of task completion for each worker based on an exponential distribution to account for worker mobility and task completion time constraints. The work optimizes task allocation by predicting worker mobility, thereby improving system efficiency and productivity. In order to quantify the differences between the trajectories of workers, Deng et al. [25] employed a contrastive learning mechanism to learn the latent representations of the trajectories and proposed a trajectory similarity calculation model called CL-TSim based on contrastive learning.

Table 1. Methods and techniques for task allocation in smart cities.

Category	Authors	Objective
Task Allocation	Zhao et al. [9,10]	Formulate the Dynamic Delayed Binary Matching (DDBM) problem and design Value-Based Task Assignment (VBTA) and Policy Gradient-Based Task Assignment (PGTA) frameworks
	Liu et al. [11]	Propose a task allocation framework based on federated preference learning to ensure data privacy and assign tasks according to worker preferences
	Wang et al. [12]	Solve multi-task competition and priority issues, optimize task allocation to maximize spatio-temporal coverage, and consider spatio-temporal constraints
	Li et al. [13]	Propose a worker selection problem in heterogeneous perception tasks to minimize costs and maintain worker trajectories
	Li et al. [14]	Study the time-constrained task allocation problem, maximize the utility of mobile crowdsourcing platforms, and improve system efficiency and productivity
	Wang et al. [15]	Optimize task allocation from the perspective of task organizers and participants, solve problems such as unfair allocation and worker load, and improve work flexibility
Mobility Prediction	Zhang et al. [16]	Multi-task allocation method based on mobility prediction to optimize task completion rate
	Yang et al. [17]	Use the markov model to predict the probability of workers completing tasks and optimize task allocation
	Wang et al. [18]	Multi-objective optimization algorithms using movement prediction models, including task coverage maximization and task cost minimization
	Wang et al. [19]	Multi-task allocation algorithm for maximizing spatiotemporal coverage under total budget constraint
Prediction and Task Allocation	Cheng et al. [26]	Consider current and future workers/tasks, improve global task allocation, increase task allocation accuracy
	Zhang et al. [27]	A task allocation framework based on worker churn prediction, which optimizes the allocation of individual tasks by predicting worker churn
	Zhao et al. [28]	Considering current and future workers/tasks (location unknown), maximizing the number of task allocation
	Zhai et al. [29]	Use the SeqST-ResNet deep learning model to effectively capture the temporal dependence of historical tasks
	Wang et al. [30]	Research on task allocation problem based on worker churn to achieve the highest total task allocation reward
	Wei et al. [31]	Joint Predictive Model (JPM), considering worker location and preference category, predicts worker location and preference category
	Wang et al. [32]	Use knowledge graph technology to predict task allocation, optimize task allocation, and explore complex spatio-temporal relationships
	Quan et al. [33]	Propose Conv-STAN and CT-Voting prediction methods to predict the future distribution of crowdsourced entities

2.3. Prediction and Task Allocation

Task prediction has great potential to improve the accuracy of task allocations on mobile crowdsensing platforms, and there are many existing works on task prediction. Cheng et al. [26] were the first to carry out research on task prediction. They consider

existing and future (via prediction) workers/tasks to improve global task allocation, design an efficient grid-based prediction method to estimate the spatial distribution of future workers/tasks, and then allcate workers to tasks. Zhang et al. [27] proposed a task allocation framework based on worker turnover prediction, which helps to optimize the allocation of a single task. Researchers have also focused on more complex and realistic perception task settings, such as multi-task competition, heterogeneous perception tasks, time constraints, and party factors. Zhao et al. [28] also consider current and future workers/tasks (location unknown) entering the system to maximize the number of allcated tasks. Zhai et al. [29] proposed a novel deep learning model called SeqST-ResNet, which effectively captures the temporal dependency of historical task occurrences at multiple time scales. To capture complex spatio-temporal correlations and underlying supply-demand relationships, Wang et al. [30] investigated the task allocation problem based on worker attrition, with the aim of achieving the highest total task allocation reward while considering worker attrition. Wei et al. [31] considered both worker location and preference categories and proposed a **j**oint **p**rediction **m**odel (JPM) for location and preference category prediction of workers at each sample timestamp. Some researchers consider using knowledge graph technology to predict tasks. Wang et al. [32] explored the potential relationship between tasks and users, analyzed the characteristics of mobile users and tasks, and constructed a knowledge graph. They proposed a task allocation method based on link prediction. Quan et al. [33] proposed a **c**onvolutional **s**patio-**t**emporal **a**ttention model (Conv-STAN) and a **c**lustering-based **t**ime-weighted **v**oting method (CT-Voting) for predicting the future distribution of crowdsensing entities.

3. Preliminaries

We first introduce the definitions in MCS system.

Definition 1 (**Crowd Worker**). *We use a quaternion* $w_i =< l_i, v_i, md_i, s_i >$ *to denote crowd worker* w_i*. Among them,* l_i *represents the geographical location of the worker,* v_i *represents the moving speed of the worker,* md_i *represents the maximum moving distance of the worker, and* s_i *represents the working skill of the worker. The representation helps to fine-tune the behavioral characteristics and capabilities of workers in crowdsensing systems. Location information helps to understand a worker's geographic location, movement speed, and maximum travel distance reflect a worker's ability to move, and skill information indicates the tasks a worker can perform. The set of workers in mobile crowdsensing is defined as* $\mathcal{WS} = \{w_1, w_2, \ldots, w_m\}$.

Definition 2 (**Crowd Task**). *We use a quaternion* $t_j =< l_j, c_j, b_j, dl_j >$ *to denote the crowd task* t_j *to be allocated. Among them,* l_j *represents the location of the task,* c_j *represents the category of the task,* b_j *represents the budget of the task, and* dl_j *represents the deadline of the task. The representation helps to fully consider the requirements and constraints of the task during the task allocation process. The location of a task is critical to assessing worker viability, task category sets represent the skills or skill sets required for the task, budget constraints help ensure tasks are completed at an affordable cost, and deadlines constrain tasks completion time. The set of tasks to be allocated in spatio-temporal crowdsensing is expressed as* $\mathcal{TS} = \{t_1, t_2, \ldots, t_n\}$*, also known as the set of candidate tasks.*

Definition 3 (**Historical Task Set**). *When processing the historical task set* HTS_i *completed by a worker, we observe that each worker-task pair is represented as a tuple* $< workerID, taskID, timestamp, location >$*.* $workerID$ *represents the ID of the worker,* $taskID$ *represents the ID of the task,* $timestamp$ *represents the time when the task was executed, and* $location$ *represents the location where the task was performed. Since the size of the historical task set* HTS *for each worker may vary, we need to process it to fit the model proposed in this paper. To achieve this goal, we transform the worker's historical task set into a fixed-length task sequence, where* ml *is the maximum length we consider. This way, regardless of the number of historical tasks completed by the worker, they can be uniformly represented as a fixed-length task sequence. When dealing with task sets of varying*

lengths, we employ the following strategy: if $length(HTS_i) > ml$, we only keep the most recent ml tasks; if $length(HTS_i) < ml$, we pad zeros to the right of HTS_i until the task sequence reaches ml.

Definition 4 (Knowledge Graph). *A knowledge graph is represented by $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{R})$, where $\mathcal{V}$ denotes the set of entities, $\mathcal{E}$ represents the set of relationship types, and $\mathcal{R}$ denotes a set of subject-relationship-object triples, where each triple $r = (h, p, t)$ represents a relationship p from head entity h to tail entity t. Specifically, (h, p, t) indicates the relationship p between the head entity h and the tail entity t. For example, (Tom, pick up, package) indicates that Tom picked up a package.*

Definition 5 (Worker-Task Matching). *Given the knowledge graph $\mathcal{G}$, the worker w_i, and the historical task set HTS_i of worker w_i, the goal of the task allocation problem is to allocate the tasks that worker w_i is most likely to perform.*

4. Transition Graph Learning

This section introduces an explicit approach to learning the transition patterns between spatiotemporal tasks. This method utilizes a knowledge graph to generate representations for each entity and relationship. It defines neighborhoods based on these representations to obtain a graph representing the transition patterns between spatiotemporal tasks. This transition graph can be used for the spatiotemporal task allocation task. The method learns the transition patterns between spatiotemporal tasks by leveraging the knowledge graph. The knowledge graph consists of spatiotemporal tasks as entities and their relationships as edges or relations. Representations for each entity and relationship in the graph can be derived from various sources such as textual descriptions, worker comments, or other contextual information. We create neighborhoods around each spatiotemporal task based on these representations, capturing the relevant spatiotemporal tasks and relationships in the knowledge graph. Neighborhoods can be defined by considering the proximity of entities and the strength of relationships. By constructing neighborhoods, we form a graph representing the transition patterns between spatiotemporal tasks. The edges in this graph represent the transitions between spatiotemporal tasks. The edges' strength or weight can be determined based on the learned representations to indicate the relevance or likelihood of transitions. This transition graph can be used for the spatiotemporal task allocation. Given the current spatiotemporal task, the graph can provide insights about possible tasks based on observed transition patterns in the neighborhoods. This method explicitly models and captures the transition patterns between spatiotemporal tasks using a knowledge graph and learned representations. This enables the generation of a graph for the spatiotemporal task allocation.

4.1. Spatio-Temporal Knowledge Graph

We propose a novel knowledge graph called Spatio-temporal Knowledge Graph (STKG), as shown in Figure 2. STKG integrates the traditional spatiotemporal task interaction graph, the spatio-temporal correlations between tasks, and the social relationships among workers. Formally, STKG is represented as $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{R})$, where $\mathcal{V}$ represents the union of the worker set and the task set $\mathcal{V} = \mathcal{WS} \cup \mathcal{TS}$, $\mathcal{E}$ represents edges that belong to one of the four types of relationships: execution, temporal, spatial, and social. It is important to note that the execution and temporal relationships are directed. Our goal is to learn the transition patterns between spatiotemporal tasks using the execution information recorded by all workers. Next, we will provide a detailed description of how to construct the four types of relationships mentioned above.

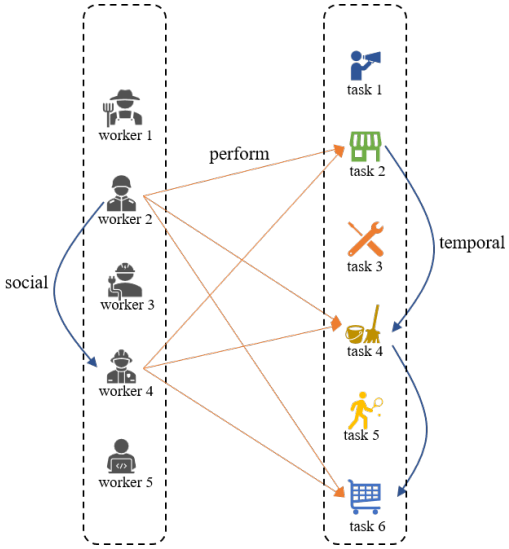


Figure 2. A simple illustration of our Spatial-Temporal Knowledge Graph.

Construction of Execution Relationship: Firstly, we construct the execution relationship represented by the triple $(worker, r_{wt}, task)$, which indicates that worker has executed task.

Construction of Temporal Relationship: Next, we construct the temporal relationship represented by the triple $(task_1, r_{tt}, task_2)$, which indicates that $task_1$ has been executed before $task_2$.

Construction of Spatial Relationship: For a given spatiotemporal task, we employ two methods to construct its spatial relationship: threshold-based method and rank-based method. The threshold-based method constructs spatial relationships between task and all other tasks whose distance to it is smaller than a predefined threshold Δ (e.g., 0.5 km). On the other hand, the rank-based method constructs spatial relationships between task and its n nearest neighboring tasks (e.g., 50 tasks).

Construction of Social Relationship: Finally, we construct the social relationship represented by the triple $(worker_1, r_{ww}, worker_2)$, which indicates that $worker_1$ and $worker_2$ have a friendship relationship. Considering the friendship relationship, when $worker_1$ executes a task, if $worker_1$ and $worker_2$ are friends, then $worker_2$ may also execute similar tasks, refer to Figure 2.

The above describes the construction process of the spatiotemporal task knowledge graph we designed. The goal of this knowledge graph is to learn the transition patterns between spatiotemporal tasks by integrating execution information, in order to better understand the relationships and dependencies among tasks.

4.2. Spatio-Temporal Knowledge Graph Embedding

Our objective, with the given spatiotemporal knowledge graph (STKG) $\mathcal{G}$, is to obtain an embedding function $f : \mathcal{V}(\mathcal{E}) \rightarrow R^m$ that assigns an m-dimensional feature vector (i.e., embedding) to each entity v $\in \mathcal{G}.\mathcal{V}$ or each relationship e $\in \mathcal{G}.\mathcal{E}$. The embedding function $f(\cdot)$ is expected to preserve the intrinsic properties of the graph $\mathcal{G}$. To accomplish this, we employ Knowledge Graph Embedding (KGE) algorithms that rely on translation-based distance models, including TransE [34], TransH [35], and TransR [36], to acquire representations for entities and relationships within the knowledge graph. As an illustration of our embedding scheme, we provide an example using TransH. TransH is based on the concept of using distinct hyperplanes to represent different relationship spaces, considering relationships as translation operations on these hyperplanes. Specifically, for a triplet

$(head, relation, tail)$, the entity embeddings head h and tail t are first projected onto a hyperplane w_r with the constraint $||w_r||_2 = 1$, as shown in Equation (1).

$$\begin{aligned} h_\perp &= h - w_r^T h w_r \\ t_\perp &= t - w_r^T t w_r \end{aligned} \tag{1}$$

where $h_\perp$ and $t_\perp$ represent the projected embeddings of h and t, respectively. Next, we use a scoring function $g(\cdot)$ to measure the credibility of incorrect triplets, as given by Equation (2).

$$g_r(h,t) = \|h_\perp + d_r - t_\perp\|_2^2 \tag{2}$$

here, d_r denotes the translated embedding on the hyperplane. A lower value of $g_r(h,t)$ indicates a higher likelihood of correctness for the triplet, whereas a higher value suggests a reduced likelihood of correctness.

At this stage, representations have been acquired for every entity and relationship. The subsequent goal is to devise a function $s(\cdot)$ that captures the transfer patterns among spatio-temporal tasks. Drawing inspiration from KGE algorithms, we calculate the similarity between spatio-temporal $task_1$ and $task_2$ by defining a spatio-temporal similarity function $s(\cdot)$ that aims to capture the spatio-temporal relationship between $task_1$ and $task_2$. Using TransE as an illustration, the similarity function can be represented by Equation (3):

$$\begin{aligned} s(task_1, task_2) &= e^{-d_t(task_1, task_2)} e^{-d_s(task_1, task_2)} \\ d_t(task_1, task_2) &= \|task_1 + r_t - task_2\|_2^2 \\ d_s(task_1, task_2) &= \frac{e^{|string_3|}}{\sum_{j=1}^{MP} e^j} \end{aligned} \tag{3}$$

where e represents the exponential function, $d_t(\cdot)$ is the time distance function, $task_1$ and $task_2$ represent the learned embeddings, and r_t is the embedding of the time relationship. $d_s(\cdot)$ is the spatial distance function. $|string_3|$ represents the length of $string_3$, MP is maximum precision of GeoHash. The spatial distance function is based on the GeoHash encoding. Specifically, given the latitude and longitude coordinates of $task_1$ and $task_2$, in the format $< longitude, latitude >$, the GeoHash encoding is obtained to correspond to binary codes $geoLog_x$, $geoLat_x$, $geoLog_y$, $geoLat_y$. Following the rule of "even bits for longitude, odd bits for latitude," the GeoHash codes of latitude and longitude are combined to generate a new string $string_1$ and $string_2$. The resulting string is then processed into a sequence of encodings according to the rules of base32 encoding. The two encodings are compared starting from the first position in the sequence until different encodings are encountered, resulting in $string_3$.

Next, we construct the spatio-temporal task transfer matrix $M \in \mathbb{R}^{|\mathcal{TS}| \times |\mathcal{TS}|}$ based on Equation (3). However, if the number of spatio-temporal tasks is large, storing M in memory can be memory-consuming, which is a common issue in task allocation scenarios. To mitigate the space cost, we construct a sparse transfer graph represented as G. More specifically, we compute the n-nearest neighbor set $\mathcal{N}_n$ for each spatio-temporal task. G is presented by Equation (4):

$$G(i,j) = \begin{cases} M(i,j), & if \quad task_j \in \mathcal{N}_n(task_i) \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

where $i, j = 1, 2, \ldots, |TS|$. Finally, to normalize the sparse graph G between 0 and 1, we select the maximum value in each row to construct the diagonal matrix D. By Equation (5), we obtain the learned spatio-temporal task transfer graph A.

$$A = D^{-1} G \tag{5}$$

5. Model Framework

This section provides a detailed introduction to the framework we propose. The framework consists of several key components: (1) Embedding layer, used to learn dense representations of workers and spatiotemporal tasks. Through this layer, we can convert workers and spatiotemporal tasks into high-dimensional vector representations to capture their feature information. (2) GCN layer, used to enrich the representation of spatiotemporal tasks. With the spatiotemporal task transfer graph learned in advance, the GCN layer can propagate and fuse information from neighboring spatiotemporal tasks, thereby enhancing the expressive power of spatiotemporal task representation. (3) Aggregation layer, used to learn weighted spatiotemporal and worker preference effects as the output of the aggregated hidden state. In this layer, we consider the influence of spatiotemporal distance and worker preferences on spatiotemporal task selection, and model these factors through weight adjustment to generate the final aggregated hidden state. (4) Prediction layer, used to allocate the most probable spatiotemporal task based on the aggregated output and worker embeddings.

Figure 3 illustrates the network architecture of the proposed GTA model. Through this network, we can effectively learn and model the information of workers and spatiotemporal tasks, and make allocation for the spatiotemporal task based on the learned representations. The design of the entire network framework aims to fully utilize the spatiotemporal relationships and worker preferences to improve the accuracy and personalization of spatiotemporal task allocation.

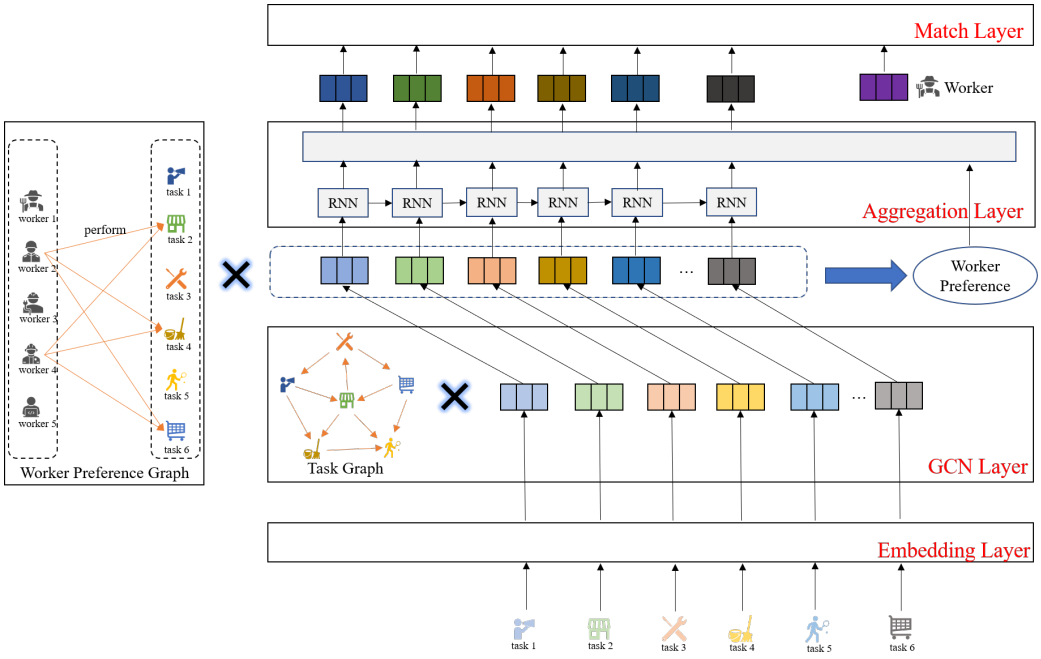


Figure 3. The overview of GTA.

5.1. Multimodal Embedding Layer

To better represent workers and spatiotemporal tasks, we propose a multimodal embedding layer that jointly encodes information of workers and spatiotemporal tasks, and integrates with the subsequent allocation module to more effectively accomplish task allocation. During the allocation process, vector representations play a crucial role in the overall system performance, so we need to employ appropriate methods to make full use of the available information. Initially, the information of workers and spatiotemporal tasks is represented as one-hot vectors, but due to sparsity, this representation fails to accurately capture workers' task preferences. To address this, we enhance the representation by learning low-dimensional dense embeddings, incorporating more useful information onto the standard sparse representation. Specifically, we first partition each worker's historical task

sequence (HTS) into several equally sized subsequences, which are then processed as inputs to the embedding layer to learn a low-dimensional dense representation. Additionally, to obtain more task-related information, we use a one-hot vector of dimension $|TS|$ to represent each spatiotemporal task. Furthermore, since different workers possess distinct characteristics and preferences, we utilize a one-hot vector of dimension $|WS|$ to represent each worker's unique features, better capturing the differences among workers. Through this approach, the embedding layer can transform one-hot vectors into corresponding low-dimensional dense representations, $Embedding(worker) \in \mathbb{R}^d$ and $Embedding(task) \in \mathbb{R}^d$, respectively, to better accomplish task allocation, d is the dimension of the embedding layer.

5.2. GCN Layer

To adequately capture the characteristics of each spatiotemporal task, relying solely on the learned low-dimensional dense representations proves insufficient. Hence, we employ a Graph Convolution Network (GCN) to further optimize the task representation. Building upon the accomplishments of LightGCN [8], our focus lies solely on neighbor aggregation within the GCN layer to enhance the precision and efficiency of the representation. It is important to note that the transition graph A does not account for the influence of each task on itself. To tackle this limitation, we introduce a unit matrix I (also known as a self-connection) to A, resulting in a new transition graph $\hat{A}$. This allows us to comprehensively capture the relationships between tasks while considering the tasks' individual characteristics. By conducting neighboring aggregation and transforming $\hat{A}$ through the GCN layer, we acquire more accurate spatiotemporal task representations, establishing a foundation for subsequent worker-task allocation.

$$\hat{A} = A + I \tag{6}$$

Next, we normalize the new transition graph using Equation (7):

$$\hat{A} = \hat{D}^{-1}\hat{A} \tag{7}$$

where $\hat{D} = diag(\hat{A})$ represents the out-degree diagonal matrix derived from $\hat{A}$. We employ the transition graph $\hat{A}$ to enrich the representation of each spatiotemporal task. Finally, we have:

$$\hat{X} = \hat{A}X \tag{8}$$

where $X \in \mathbb{R}^{|TS| \times d}$ represents the previous embeddings of all spatiotemporal tasks, and $\hat{X} \in \mathbb{R}^{|TS| \times d}$ is the updated spatiotemporal task embedding.

5.3. Aggregation Layer

The aggregation layer comprises a recurrent module and an aggregation module, each serving distinct roles. The recurrent module captures sequential patterns, aiding in the comprehension of workers' check-in trajectories and enabling more precise task allocation. On the other hand, the aggregation module considers the influence of historical hidden states on the current hidden state, as it is crucial to consider workers' past behavioral records to accurately predict future behavior during task allocation.

Within the output of the GCN layer, the updated spatiotemporal task embeddings and worker preference embeddings are inputted into the aggregation layer. To enhance allocation accuracy, we employ a basic recurrent neural network (RNN) as the recurrent module to obtain all hidden states. However, using these hidden states directly for allocation fails to fully exploit the temporal periodicity and spatial context embedded within workers' check-in trajectories. Notably, workers commonly engage in periodic spatiotemporal tasks, such as those close to their residential areas or tasks with shorter distances. Hence, taking inspiration from Flashback [4], we explicitly devise a similarity function $w(\cdot)$ that incorporates temporal and spatial context to reflect the correlation between the historical hidden state h_j and the current hidden state h_i. Through this function, we can

more precisely analyze workers' behavioral records and recommend more suitable tasks based on them. The definition of $w(\cdot)$ can be found in Equation (9):

$$w(\Delta T_{i,j}, \Delta D_{i,j}) = hvc(2\pi\Delta T_{i,j})e^{-\alpha\Delta T_{i,j}}e^{-\beta\Delta D_{i,j}} \tag{9}$$

where, $hvc(x) = (1+\cos x)/2$ represents temporal periodicity. Secondly, $\Delta D_{i,j}$ and $\Delta T_{i,j}$ are used to denote the spatial and temporal intervals between two spatiotemporal tasks i and j. Parameters α and β are used to represent the temporal and spatial decay weights. It should be noted that the similarity function $w(\cdot)$ only focuses on the similarity correlation between spatiotemporal tasks, neglecting the general preference of workers for spatiotemporal tasks. To better reflect the overall task preferences of workers, we propose to integrate worker preferences into the function $w(\cdot)$. Specifically, we can use the access relationship to construct a sparse worker-spatiotemporal task preference graph $\mathcal{G}_p \in R^{|WS|\times|TS|}$, which is the same as the method introduced earlier. Then, we can obtain the general preference of each worker for spatiotemporal tasks through Equation (10), thereby better considering the overall task preferences of workers and providing more personalized task allocation. When integrating worker preferences, we also need to ensure that the output of the function $w(\cdot)$ is constrained within the range [0, 1] to ensure the rationality of the allocation results. With these improvements, the similarity function $w(\cdot)$ can more accurately calculate workers' task preferences and spatiotemporal relationships, thus providing better task allocation. Equation (10) calculates $\mathcal{P}$:

$$\mathcal{P} = \mathcal{G}_p\hat{X} \tag{10}$$

where $\mathcal{P} \in \mathbb{R}^{|TS|\times d}$ is the worker preference matrix. Finally, for each worker, we obtain a new similarity function $\hat{w}(\cdot)$, Equation (11), which simultaneously considers worker preference weights and spatiotemporal weights:

$$\hat{w}(\Delta T_{i,j}, \Delta D_{i,j}) = w(\Delta T_{i,j}, \Delta D_{i,j})e^{-\left\|\mathcal{P}_{worker} - e^{task_j}\right\|} \tag{11}$$

where $\mathcal{P}_{worker}$ worker represents the worker's preference embedding, $||\cdot||$ represents the $L2$ distance. The aggregation module combines the similarity function $\hat{w}(\cdot)$ and historical hidden states into the current hidden state at each time step i. Thus, we have:

$$\widehat{h}_i = \frac{\sum_{j=0}^{i} \hat{w}_j * h_j}{\sum_{j=0}^{i} \hat{w}_j} \tag{12}$$

where $\hat{w}_j$ indicates the similarity $\hat{w}(\Delta T_{i,j}, \Delta D_{i,j})$.

5.4. Prediction Layer

At each time step t, the output $\hat{h}_t$ of the aggregation layer and the worker embedding $E(worker)$ are concatenated into a new vector, which is then passed through a fully connected layer to generate the final output using Equation (13):

$$\hat{y} = W_f[\widehat{h}_t||E(worker)] \tag{13}$$

where $W_f \in R^{|TS|\times 2d}$ is a learnable weight matrix, $||$ represents the concatenation operation. We use the cross-entropy function, represented by Equation (14), as our loss function:

$$-\sum_{ws=1}^{|WS|}\sum_{i=1}^{ml}\left(\log\sigma(y_k^{ws}) + \sum_{j=1,j\neq k}^{|TS|}\log\left(1-\sigma\left(\hat{y}_j^{ws}\right)\right)\right) \tag{14}$$

where ml represents the length of the spatiotemporal task sequence for each worker, and represent the predicted values for the current task i of the worker with respect to the label task k and other spatiotemporal tasks $j \neq$ task k, respectively. σ is the softmax function.

6. Experimental

6.1. Dataset

We evaluated our GTA model on three widely used real-world datasets: Foursquare-NYC, TKY [37], and Gowalla (http://snap.stanford.edu/data/loc-gowalla.html (accessed on 4 July 2023)) [38]. These datasets originate from different geographical regions and contain a large amount of worker, location, and check-in information. Each spatiotemporal task record includes user ID, task ID, latitude, longitude, and timestamp. Our experiments are based on these original datasets. For experimental convenience, we consider workers as mobile crowdsensing workers and enterprises as mobile crowdsensing tasks. We sort the task records of each user in ascending order of timestamps. The task records of each user are divided into multiple equal-length sequences as the training set, which accounts for 80% of the records. Similarly, the remaining 20% is considered as the test set. Furthermore, the training set is used to construct our spatiotemporal knowledge graph. Table 1 shows the statistical information of the two datasets used in our experiments. Although all three datasets have four types of relationships, we only utilize the execution relationship.

By conducting experiments on these four real-world datasets, we can comprehensively evaluate the performance of our proposed model in different scenarios and applications. This helps us further optimize the model and improve its prediction accuracy and generalization ability in practical mobile crowdsensing tasks.

6.2. Evaluation Metrics

We use the Top-K accuracy metric commonly used in recommender systems to evaluate our proposed algorithm. This metric has multiple variants, including ACC@1, ACC@5, and ACC@10, which represent the proportion of correctly allocated positive samples among all positive samples in the Top 1, Top 5, and Top 10 allocation results, respectively. In the context of recommender systems, ACC@K is used to measure the accuracy of predicting worker preferences. In the context of crowdsensing systems, it is used to evaluate the accuracy of task allocation, examining how many tasks of interest to the workers are included in the top K allocation list. Therefore, ACC@K can help us understand the performance of the model under different lengths of allocation task lists. Clearly, a higher ACC@K indicates better allocation performance of the model.

6.3. Parameter Settings

When constructing the knowledge graph, we utilize ranking information to determine the strength of relationships among entities within the graph. We regard the ranking information as the weight or distance measure of the edges in the graph. A higher ranking signifies a stronger correlation between entities, whereas a lower ranking implies a weaker correlation, thereby capturing significant relationships and patterns among entities. To construct the spatiotemporal task transition graph, we employ TransE. To build the spatiotemporal task transition graph A and the user-spatiotemporal task preference graph $\mathcal{G}_p$, we consider the n_t and $n_p = \{20, 40, 60, 80, 100\}$ nearest neighbors for each spatiotemporal task (user). The default number of neighbors is set to 80. Additionally, we set the embedding dimensions d to 60 for hidden states, workers, and spatiotemporal tasks for the TKY and NYC datasets, and 30 for the Gowalla dataset. The time decay factor α and space decay factor β adhere to the default settings in [4]. We employ the Adam optimizer with default parameter settings, a learning rate of 0.001, a dropout rate of 0.3, and a training period of 80 epochs. We assess the model's performance every 10 epochs, and the maximum length of the historical task list is limited to 100. The batch size is 1, as displayed in Table 2.

Table 2. Parameters.

Parameters	Value
Learning rate	0.001
Embedding dimension	50
Batch size	1
Dropout rate	0.3
Epochs	80
number of neighbors	50

6.4. Comparative Algorithms

Here are the descriptions of the comparative algorithms:

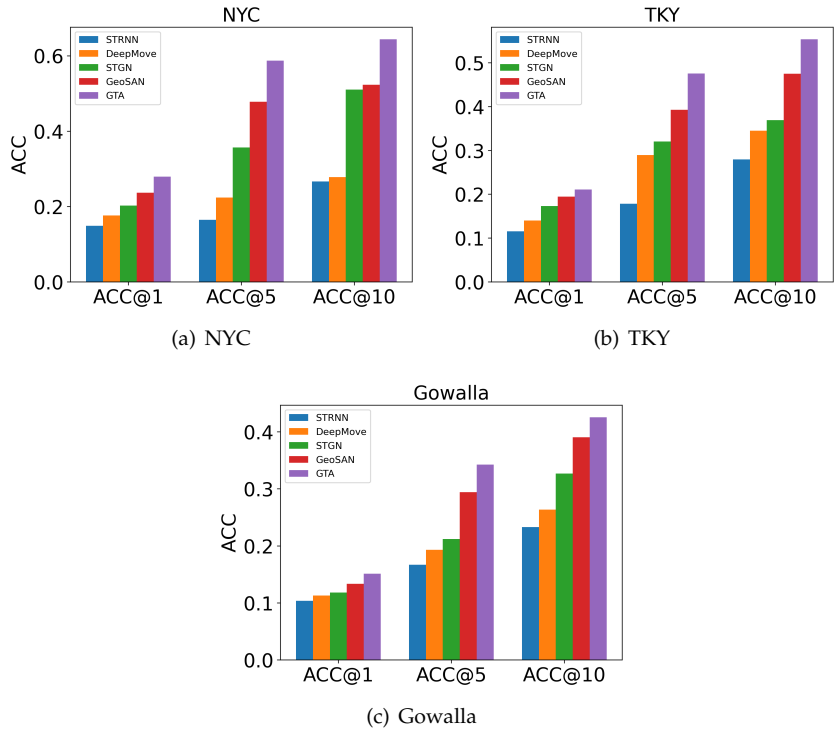
- STRNN [39]: An invariant RNN model that incorporates spatio-temporal features between consecutive visits.
- DeepMove [40]: A state-of-the-art model that utilizes recurrent and attention layers to capture periodicity.
- STGN [41]: Learn the long-term and short-term preferences of users and extend the LSTM model using spatial and temporal gates.
- GeoSAN [42]: A state-of-the-art model that employs hierarchical gridding of GPS locations for spatial discretization and utilizes self-attention layers for matching, without explicit use of spatio-temporal intervals.

6.5. Allocation Performance

In this study, we conducted a comparative analysis of various allocation models and thoroughly explored their key characteristics. The results obtained by our proposed GTA model are highlighted in bold in Table 3, while the best-performing results among the comparative models are underlined. The findings from the Foursquare-NYC, Foursquare-TKY, and Gowalla datasets demonstrate the significant superiority of our proposed GTA model over other state-of-the-art baseline methods across all evaluation metrics, particularly on the Foursquare-NYC dataset. Remarkably, on the Foursquare-NYC dataset, GTA exhibits relative improvements of 18.01%, 22.95%, and 23.17% over the second-best method GeoSAN in terms of ACC@1, ACC@5, and ACC@10, respectively. On the Foursquare-TKY dataset, GTA achieves an average improvement of 16.82% compared to GeoSAN. Moreover, on the Gowalla dataset, GTA demonstrates an average improvement of 12.38% over GeoSAN. These results clearly indicate the superiority of GTA in effectively handling spatiotemporal tasks, as shown in Figure 4.

Table 3. Allocation performance comparison.

	NYC			**TKY**			**Gowalla**		
	ACC@1	**ACC@5**	**ACC@10**	**ACC@1**	**ACC@5**	**ACC@10**	**ACC@1**	**ACC@5**	**ACC@10**
STRNN	0.1487	0.1645	0.2661	0.1153	0.1783	0.2793	0.1037	0.1669	0.2327
DeepMove	0.1761	0.2235	0.2781	0.1398	0.2893	0.3451	0.1129	0.1931	0.2637
STGN	0.2023	0.3566	0.5102	0.1728	0.3203	0.3689	0.1181	0.2118	0.3268
GeoSAN	<u>0.2365</u>	<u>0.4775</u>	<u>0.5226</u>	<u>0.1942</u>	<u>0.3925</u>	<u>0.4747</u>	<u>0.1333</u>	<u>0.2942</u>	<u>0.3905</u>
GTA	**0.2791**	**0.5871**	**0.6437**	**0.2105**	**0.4757**	**0.5537**	**0.1512**	**0.3425**	**0.4256**



(a) NYC (b) TKY

(c) Gowalla

Figure 4. Allocation accuracy on Foursquare NYC, TKY and Gowalla.

Upon comparing the performance improvements on different datasets, we observed that GTA performs exceptionally well on the Foursquare-NYC dataset. We attribute this to the dataset's richer relationships in comparison to the Gowalla dataset. Compared to GeoSAN, our GTA model consistently exhibits significant improvements. By leveraging the spatiotemporal task transition graph A to optimize task embeddings, we are able to better capture patterns of task transitions. Additionally, the considered user preferences significantly enhance the performance of personalized spatiotemporal task allocation.

6.6. Ablation Study

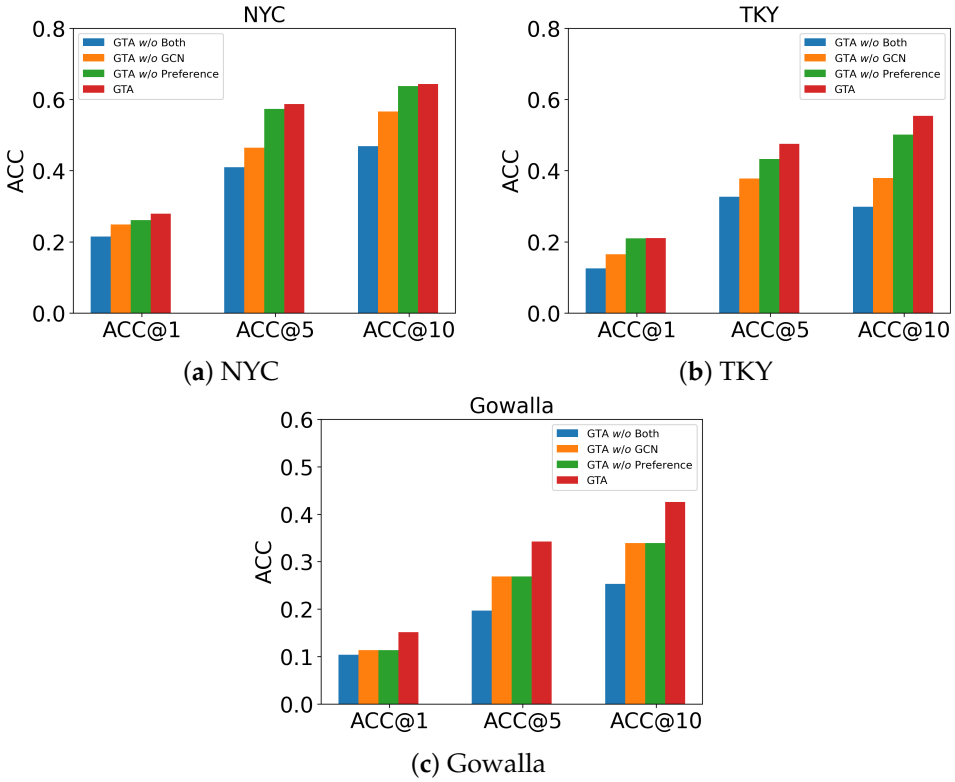
Our framework comprises two primary components: (i) the Graph Convolutional Network (GCN) layer and (ii) the aggregation layer. To showcase the impact of these components, we performed a sensitivity analysis using three datasets, the results of which are outlined in Table 4. Based on our analysis, we draw the following conclusions:

Table 4. Ablation experiments on NYC, TKY and Gowalla dataset.

	NYC			**TKY**			**Gowalla**		
	ACC@1	**ACC@5**	**ACC@10**	**ACC@1**	**ACC@5**	**ACC@10**	**ACC@1**	**ACC@5**	**ACC@10**
GTA *w/o* Both	0.2147	0.4095	0.4687	0.1253	0.3269	0.2983	0.1037	0.1969	0.2529
GTA *w/o* GCN	0.2487	0.4645	0.5661	0.1653	0.3783	0.3792	0.1137	0.2689	0.3394
GTA *w/o* Preference	0.2612	0.5734	0.6378	0.2098	0.4327	0.5013	0.1229	0.3019	0.4037
GTA	**0.2791**	**0.5871**	**0.6437**	**0.2105**	**0.4757**	**0.5537**	**0.1512**	**0.3425**	**0.4256**

From our observations, we have noted that the utilization of the spatiotemporal task transition graph within the GCN layer leads to an enhancement in model performance. The learned graph contributes to the enrichment of spatiotemporal task representations, aiding the sequence model in capturing the patterns of transition between these tasks. Furthermore, our findings reveal that the correct incorporation of user preferences plays a significant role in improving model performance. The integration of the learned user-

spatiotemporal task preference graph with the spatiotemporal context results in noticeable enhancements in personalized spatiotemporal task allocation. We have also observed a slight performance advantage in GTA over GTA w/o Preference. This outcome can be attributed to the partial reflection of user preferences within the GCN layer, as shown in Figure 5. Our study emphasizes the importance of the spatiotemporal task transition graph and worker preferences in boosting allocation accuracy. These essential factors should be thoroughly considered during the task allocation process to achieve higher-quality and more precise allocation outcomes.



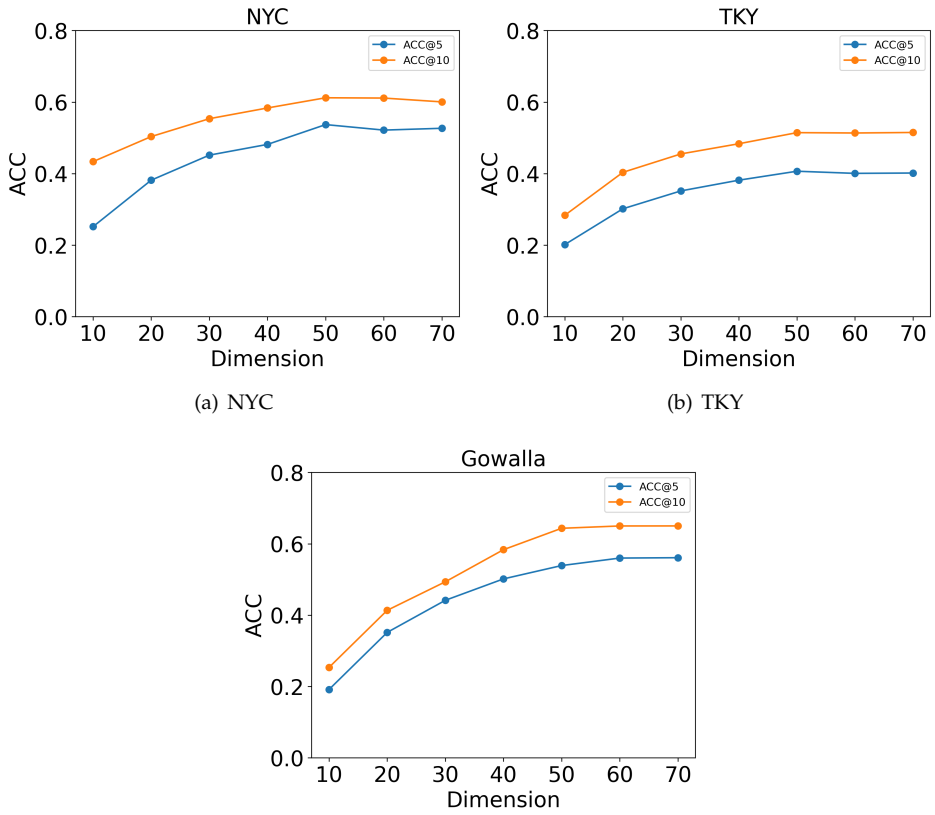
(**a**) NYC (**b**) TKY

(**c**) Gowalla

Figure 5. Influence of different modules on allocation performance.

6.7. Comparison Experiment on Embedding Layer Dimension

We also investigated the impact of the embedding layer dimension hyperparameter on the allocation performance. We varied the embedding dimension d of the multimodal embedding module from 10 to 70 in increments of 10. Figure 6 presents the experimental results on the sensitivity of the performance to the embedding layer dimension d for the four datasets. It can be concluded that as d increases, the effectiveness of the model predictions initially improves and then stabilizes, as higher-dimensional embeddings can represent more complex interaction information and capture more latent features. In the case of optimal performance, further increasing the dimension would be wasteful of system resources. From Figure 6a,b, it can be observed that on the NYC dataset, the model performance is almost optimal when the dimension d reaches 50, and there is only a minimal change of 0.5% in the allocation performance when the embedding dimension d exceeds 50, indicating stability. From Figure 6c, it can be seen that on the Gowalla dataset, the model performance is optimal when the dimension d reaches 60 and stabilizes.



(a) NYC (b) TKY

(c) Gowalla

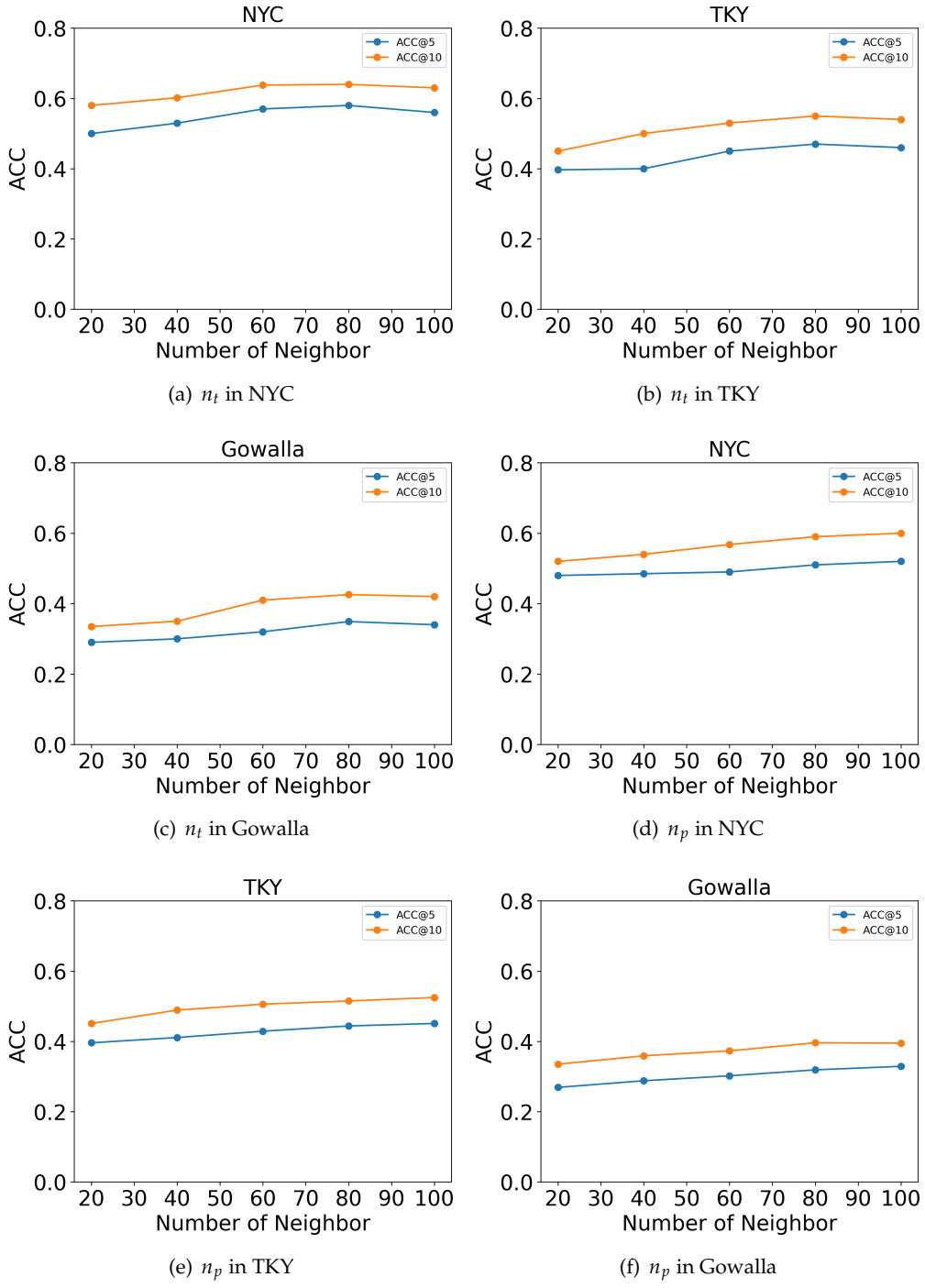
Figure 6. Parameter sensitivity results of embedding dimension.

This is because the Gowalla dataset requires a larger embedding layer dimension to extract features. From Figure 6, it can be observed that although the embedding layer dimension varies from 10 to 70 and the allocation accuracy increases, the improvement is relatively small. For practical applications, the impact on the model's allocation performance can be overlooked to some extent. When the embedding dimension is set to 50, the model can achieve optimal trajectory and spatiotemporal embedding performance, providing strong support for subsequent allocation tasks. Therefore, when selecting the embedding dimension, as long as it is greater than 50, we can ensure the model's allocation performance to a large extent.

6.8. Comparison Experiment on Graph Neighbor

We conducted a comparison experiment on the number of graph neighbors and provided an analysis of the experimental results. According to the results in Figure 7a–c, we found that $n_t = 80$ are the optimal numbers of nearest neighbors for spatiotemporal tasks in the NYC, TKY and Gowalla, respectively. On the Gowalla dataset, as n_t increases, so does the accuracy, flattening out when n_t reaches 60. However, on the NYC and TKY datasets, model performance improves correspondingly as n_t increases, but it decreases slightly after reaching 80. This may be due to Gowalla's sparser spatiotemporal data compared to Foursquare. Therefore, for each task, the Gowalla and Foursquare datasets need more and appropriate neighbors to enrich their representations.

Furthermore, based on the results in Figure 7d–f, we observe that $n_p = 100$ achieves better model performance on both the Gowalla and Foursquare datasets, and the model's performance on the ACC@5 and ACC@10 metric is relatively stable.



(a) n_t in NYC (b) n_t in TKY

(c) n_t in Gowalla (d) n_p in NYC

(e) n_p in TKY (f) n_p in Gowalla

Figure 7. The performance comparison about the number of neighbors n_t and n_p.

7. Conclusions

This paper and its method are highly significant for journals and fields related to urban planning. Using a knowledge graph-based approach, the study successfully learns spatiotemporal crowdsensing graphs, capturing transition patterns between different tasks. This novel research avenue enhances our understanding of spatiotemporal correlations and evolutionary patterns in urban planning. The innovative GTA model integrates the learned graph into existing sequence models, greatly improving transition pattern capture. This has valuable applications in urban planning and resource optimization, aiding decision-makers in understanding task evolution and formulating effective development strategies. Addressing potential participant preference variations, our carefully designed similarity

functions enhance the model's practicality in urban planning scenarios. Experimental results confirm the accuracy of the GTA model, showcasing its potential in urban planning. Ablation experiments validate each component's effectiveness. Future research will explore supplementary information integration, like task categories and participant data, and delve deeper into examining participant similarity relationships, further enhancing the model's performance. This paper and method offer fresh perspectives and tools for urban planning, providing valuable references for professionals in the academic and practical communities.

Author Contributions: Conceptualization, B.Z. and H.D.; methodology, B.Z.; software, B.Z.; validation, B.Z., H.D. and D.Y.; formal analysis, B.Z.; investigation, B.Z.; resources, D.Y.; data curation, H.D.; writing—original draft preparation, B.Z.; writing—review and editing, B.Z. and H.D.; visualization, B.Z.; supervision, H.D. and D.Y.; project administration, H.D. and D.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Liu, S.; Yu, Y.; Guo, L.; Yeoh, P.L.; Ni, Q.; Vucetic, B.; Li, Y. Truthful online double auctions for mobile crowdsourcing: An on-demand service strategy. *IEEE Internet Things J.* **2022**, *9*, 16096–16112. [CrossRef]
2. Xu, Y.; Wang, L.; Wang, Y.; Fu, Y. Adaptive trajectory prediction via transferable gnn. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 6520–6531.
3. Ray, A.; Chowdhury, C.; Bhattacharya, S.; Roy, S. A survey of mobile crowdsensing and crowdsourcing strategies for smart mobile device users. *CCF Trans. Pervasive Comput. Interact.* **2023**, *5*, 98–123. [CrossRef]
4. Yang, D.; Fankhauser, B.; Rosso, P.; Cudre-Mauroux, P. Location prediction over sparse user mobility traces using rnns. In Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, Yokohama, Japan, 7–15 January 2021; pp. 2184–2190.
5. Rao, X.; Chen, L.; Liu, Y.; Shang, S.; Yao, B.; Han, P. Graph-flashback network for next location recommendation. In Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, 14–18 August 2022; pp. 1463–1471.
6. Han, P.; Wang, J.; Yao, D.; Shang, S.; Zhang, X. A graph-based approach for trajectory similarity computation in spatial networks. In Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, Singapore, 14–18 August 2021; pp. 556–564.
7. Li, Y.; Chen, T.; Luo, Y.; Yin, H.; Huang, Z. Discovering collaborative signals for next POI recommendation with iterative Seq2Graph augmentation. *arXiv* **2021**, arXiv:2106.15814.
8. He, X.; Deng, K.; Wang, X.; Li, Y.; Zhang, Y.; Wang, M. Lightgcn: Simplifying and powering graph convolution network for recommendation. In Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, Xi'an, China, 25–30 July 2020; pp. 639–648.
9. Zhao, B.; Dong, H.; Wang, Y.; Pan, T. A task allocation algorithm based on reinforcement learning in spatio-temporal crowdsourcing. *Appl. Intell.* **2022**, *53*, 13452–13469. [CrossRef]
10. Zhao, B.; Dong, H.; Wang, Y.; Pan, T. PPO-TA: Adaptive task allocation via Proximal Policy Optimization for spatio-temporal crowdsourcing. *Knowl.-Based Syst.* **2023**, *264*, 110330. [CrossRef]
11. Liu, J.; Deng, L.; Miao, H.; Zhao, Y.; Zheng, K. Task Assignment with Federated Preference Learning in Spatial Crowdsourcing. In Proceedings of the 31st ACM International Conference on Information & Knowledge Management, Atlanta, GA, USA, 17–21 October 2022; pp. 1279–1288.
12. Wang, J.; Wang, Y.; Zhang, D.; Wang, F.; Xiong, H.; Chen, C.; Lv, Q.; Qiu, Z. Multi-task allocation in mobile crowd sensing with individual task quality assurance. *IEEE Trans. Mob. Comput.* **2018**, *17*, 2101–2113. [CrossRef]
13. Li, H.; Li, T.; Wang, W.; Wang, Y. Dynamic participant selection for large-scale mobile crowd sensing. *IEEE Trans. Mob. Comput.* **2018**, *18*, 2842–2855. [CrossRef]
14. Li, X.; Zhang, X. Multi-task allocation under time constraints in mobile crowdsensing. *IEEE Trans. Mob. Comput.* **2019**, *20*, 1494–1510. [CrossRef]
15. Wang, J.; Wang, F.; Wang, Y.; Zhang, D.; Lim, B.; Wang, L. Allocating heterogeneous tasks in participatory sensing with diverse participant-side factors. *IEEE Trans. Mob. Comput.* **2018**, *18*, 1979–1991. [CrossRef]

16. Zhang, J.; Zhang, X. Multi-task allocation in mobile crowd sensing with mobility prediction. *IEEE Trans. Mob. Comput.* **2021**, *22*, 1081–1094. [CrossRef]
17. Yang, Y.; Liu, W.; Wang, E.; Wu, J. A prediction-based user selection framework for heterogeneous mobile crowdsensing. *IEEE Trans. Mob. Comput.* **2018**, *18*, 2460–2473. [CrossRef]
18. Wang, L.; Yu, Z.; Han, Q.; Guo, B.; Xiong, H. Multi-objective optimization based allocation of heterogeneous spatial crowdsourcing tasks. *IEEE Trans. Mob. Comput.* **2017**, *17*, 1637–1650. [CrossRef]
19. Wang, L.; Yu, Z.; Zhang, D.; Guo, B.; Liu, C.H. Heterogeneous multi-task assignment in mobile crowdsensing using spatiotemporal correlation. *IEEE Trans. Mob. Comput.* **2018**, *18*, 84–97. [CrossRef]
20. Wang, J.; Wang, Y.; Zhang, D.; Wang, F.; He, Y.; Ma, L. PSAllocator: Multi-task allocation for participatory sensing with sensing capability constraints. In Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing, Portland, OR, USA, 25 February–1 March 2017; pp. 1139–1151.
21. Xiao, M.; Wu, J.; Huang, L.; Wang, Y.; Liu, C. Multi-task assignment for crowdsensing in mobile social networks. In Proceedings of the 2015 IEEE Conference on Computer Communications (INFOCOM), Hong Kong, 26 April–1 May 2015; IEEE: Piscataway, NJ, USA, 2015; pp. 2227–2235.
22. Guo, B.; Liu, Y.; Wu, W.; Yu, Z.; Han, Q. ActiveCrowd: A framework for optimized multitask allocation in mobile crowdsensing systems. *IEEE Trans.-Hum.-Mach. Syst.* **2016**, *47*, 392–403. [CrossRef]
23. Liu, W.; Yang, Y.; Wang, E.; Han, Z.; Wang, X. Prediction based user selection in time-sensitive mobile crowdsensing. In Proceedings of the 2017 14th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON), San Diego, CA, USA, 12–14 June 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 1–9.
24. Lai, C.; Zhang, X. Duration-sensitive task allocation for mobile crowd sensing. *IEEE Syst. J.* **2020**, *14*, 4430–4441. [CrossRef]
25. Deng, L.; Zhao, Y.; Fu, Z.; Sun, H.; Liu, S.; Zheng, K. Efficient Trajectory Similarity Computation with Contrastive Learning. In Proceedings of the 31st ACM International Conference on Information & Knowledge Management, Atlanta, GA, USA, 17–21 October 2022; pp. 365–374.
26. Cheng, P.; Lian, X.; Chen, L.; Shahabi, C. Prediction-based task assignment in spatial crowdsourcing. In Proceedings of the 2017 IEEE 33rd International Conference on Data Engineering (ICDE), San Diego, CA, USA, 19–22 April 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 997–1008.
27. Zhang, D.; Xiong, H.; Wang, L.; Chen, G. CrowdRecruiter: Selecting participants for piggyback crowdsensing under probabilistic coverage constraint. In Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing, Seattle, WA, USA, 13–17 September 2014; pp. 703–714.
28. Zhao, Y.; Zheng, K.; Cui, Y.; Su, H.; Zhu, F.; Zhou, X. Predictive task assignment in spatial crowdsourcing: A data-driven approach. In Proceedings of the 2020 IEEE 36th International Conference on Data Engineering (ICDE), Dallas, TX, USA, 20–24 April 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 13–24.
29. Zhai, D.; Liu, A.; Chen, S.; Li, Z.; Zhang, X. SeqST-ResNet: A sequential spatial temporal ResNet for task prediction in spatial crowdsourcing. In Proceedings of the Database Systems for Advanced Applications: 24th International Conference, DASFAA 2019, Chiang Mai, Thailand, 22–25 April 2019; Proceedings, Part I 24; Springer: Berlin/Heidelberg, Germany, 2019; pp. 260–275.
30. Wang, Z.; Zhao, Y.; Chen, X.; Zheng, K. Task assignment with worker churn prediction in spatial crowdsourcing. In Proceedings of the 30th ACM International Conference on Information & Knowledge Management, Queensland, Australia, 1–5 November 2021; pp. 2070–2079.
31. Wei, X.; Sun, B.; Cui, J.; Qiu, M. Location-and-Preference Joint Prediction for Task Assignment in Spatial Crowdsourcing. *IEEE Trans.-Comput.-Aided Des. Integr. Circuits Syst.* **2022**, *42*, 928–941. [CrossRef]
32. Wang, J.; Liu, J.; Zhao, G. Dynamic link prediction method of task and user in Mobile Crowd Sensing. *Comput. Commun.* **2022**, *189*, 110–119. [CrossRef]
33. Quan, J.; Wang, N. An optimized task assignment framework based on crowdsourcing knowledge graph and prediction. *Knowl.-Based Syst.* **2023**, *260*, 110096. [CrossRef]
34. Bordes, A.; Usunier, N.; Garcia-Duran, A.; Weston, J.; Yakhnenko, O. Translating embeddings for modeling multi-relational data. *Adv. Neural Inf. Process. Syst.* **2013**, *2*, 2787–2795.
35. Wang, Z.; Zhang, J.; Feng, J.; Chen, Z. Knowledge graph embedding by translating on hyperplanes. In Proceedings of the AAAI Conference on Artificial Intelligence, Quebec City, QC, Canada, 27–31 July 2014; Volume 28.
36. Lin, Y.; Liu, Z.; Sun, M.; Liu, Y.; Zhu, X. Learning entity and relation embeddings for knowledge graph completion. In Proceedings of the AAAI Conference on Artificial Intelligence, Austin, TX, USA, 25–30 January 2015; Volume 29.
37. Yang, D.; Qu, B.; Yang, J.; Cudre-Mauroux, P. Revisiting user mobility and social relationships in lbsns: A hypergraph embedding approach. In Proceedings of the World Wide Web Conference, San Francisco, CA, USA, 13–17 May 2019; pp. 2147–2157.
38. Cho, E.; Myers, S.A.; Leskovec, J. Friendship and mobility: Friendship and mobility: User movement in location-based social networks. In Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, CA, USA, 21–24 August 2011; pp. 1082–1090.
39. Liu, Q.; Wu, S.; Wang, L.; Tan, T. Predicting the next location: A recurrent model with spatial and temporal contexts. In Proceedings of the AAAI Conference on Artificial Intelligence, Phoenix, AZ, USA, 12–17 February 2016; Volume 30.
40. Feng, J.; Li, Y.; Zhang, C.; Sun, F.; Meng, F.; Guo, A.; Jin, D. Deepmove: Predicting human mobility with attentional recurrent networks. In Proceedings of the 2018 World Wide Web Conference, Lyon, France, 23–27 April 2018; pp. 1459–1468.

41. Zhao, P.; Luo, A.; Liu, Y.; Xu, J.; Li, Z.; Zhuang, F.; Sheng, V.S.; Zhou, X. Where to Go Next: A Spatio-Temporal Gated Network for Next POI Recommendation. *IEEE Trans. Knowl. Data Eng.* **2022**, *34*, 2512–2524. [CrossRef]
42. Lian, D.; Wu, Y.; Ge, Y.; Xie, X.; Chen, E. Geography-aware sequential location recommendation. In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Virtual, 6–10 July 2020; pp. 2009–2019.