Active Sensing for Transformer Model in Sparse Mobile CrowdSensing

Deran Hao[1], En Wang[1], Wenbin Liu[1,*], Weiting Liu[1], Jiang Yuan[1], Yongjian Yang[1], and Jie Wu[2]
[1]College of Computer Science and Technology, Jilin University, Changchun, China
[2]Department of Computer and Information Sciences, Temple University, Philadelphia, USA
haodr21@mails.jlu.edu.cn, {wangen,liuwenbin}@jlu.edu.cn,
liuwt20@mails.jlu.edu.cn, 1733519754@qq.com, yyj@jlu.edu.cn, jiewu@temple.edu

***Abstract*—With the popularization of mobile devices and the development of 5G networks, Mobile Crowd-Sensing (MCS) has emerged as a new paradigm for data collection. As its variant, Sparse MCS is favored by researchers because of its practicality and low costs, which only needs to select a few sub-areas and then infer the entire map. To achieve good performance on the complex sensing tasks, Sparse MCS has to utilize the powerful deep learning methods, e.g., Transformer, which actually has high requirements for training data sets. However, most existing works randomly select spatiotemporal positions to sense training data, which not only ignores the importance of the spatiotemporal positions but also may lead to unbalance sensing data distribution and affect the training of the model. Consequently, in this paper, we propose an active sensing method in Sparse MCS for the Transformer-like models. First, we consider the data correlation at different spatiotemporal positions and use it to evaluate the representativeness of each spatiotemporal position. Secondly, we assess the informativeness of each spatiotemporal position by using the newly proposed spatiotemporal attention mechanism. Then we use these two aspects to evaluate the importance of each spatiotemporal position. Finally, we evaluate the performance of our proposed method through two typical urban sensing tasks with three real-world datasets.**

***Index Terms*—Mobile crowdsensing, Active Learning, Transformer**

I. Introduction

With the popularization of mobile devices and the development of 5G technology, Mobile CrowdSensing (MCS) [1], [2], a new data collection paradigm, has garnered significant attention and played a pivotal role in environmental monitoring [3], traffic monitoring [4], etc. Due to the cost constraints and practical factors such as geography, as a variant of MCS, Sparse MCS [5], [6] has become a more practical approach to data collection. In Sparse MCS, only a few sub-areas of the entire sensing map are sensed. Based on this limited data, inference algorithms or complex neural networks are used to infer the whole sensing map. In addition, various deep learning models with powerful learning capabilities have been proposed to solve various spatiotemporal sequence prediction problems. Benefiting from their powerful ability on inference, prediction, and other sensing problems, deep learning models are gradually being integrated with Sparse MCS to handle various complex sensing tasks. Among the various deep learning models, the large-scale model Transformer [7] and its variants such as Informer [8] perform great in handling time series tasks and are particularly effective for long time series.

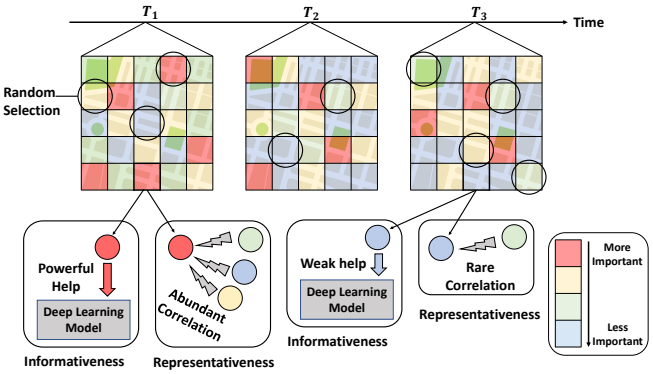


Fig. 1. Random selection of spatiotemporal positions may ignore important spatiotemporal positions and affect model accuracy. The importance of spatiotemporal positions generally includes two aspects: firstly, the potential spatiotemporal correlation with others, and secondly, the importance of the data at each spatiotemporal position to the model.

However, for Sparse MCS oriented to deep learning models, most existing works have focused on designing models and constructing neural networks while ignoring another important issue: how to select the spatiotemporal areas to be sensed. In general, training deep learning models often requires a large amount of data. However, with a limited budget, only a few spatiotemporal positions can be sensed. It is important to note that different spatiotemporal positions may have different importance. For example, in traffic flow data, measurements taken during weekday morning and evening peak hours often represent the maximum traffic volume of the day in many regions, conveying additional information compared to measurements at other hours. However, as shown in Fig.1, previous works usually select spatiotemporal sub-areas in a random way, which is inefficient and costly. This is because random selection may ignore the more important spatiotemporal positions and may also lead to an unbalanced distribution of training data, thereby negatively affecting the training accuracy. Although a few existing works [9]–[11] have explored

*Wenbin Liu is the corresponding author.

active selection methods for spatiotemporal positions, they only consider the impact of data at spatiotemporal positions on training results, ignoring the diversity and representativeness of spatiotemporal positions, which is not a comprehensive consideration. To address this issue, we combine the data collection in Sparse MCS with model training by *actively selecting spatiotemporal positions to sense data for model training under a fixed sensing budget to improve the performance of the model.*

For this goal, how to actively select different spatiotemporal positions is the key issue of this paper. First, the correlation between spatiotemporal data is one of the most typical characteristics of spatiotemporal data. When we actively select spatiotemporal positions, we should consider the impact of data correlations implied between them. However, this correlation is affected by factors such as cycles, holidays, and characteristics of the sub-area (such as population density and geographical environment within the area), which is complex and difficult to measure. Therefore, *how to evaluate the correlation of spatiotemporal data between different spatiotemporal positions* is the first challenge. Secondly, data at different spatiotemporal positions have different importance for the model. Due to the data correlation, actively selecting some spatiotemporal positions to sense the training data may provide extra help for the model to characterize other spatiotemporal data, which can be described as the informativeness of the spatiotemporal positions. Then, as a large-scale model with complex neural networks, *how to evaluate the informativeness of the spatiotemporal positions for the Transformer-like model* is our second challenge.

In response to the above challenges, we introduce the Active Learning (AL) into Sparse MCS to select the best data to sense for large-scale models. AL is a research branch of machine learning that aims to reduce the training cost of models by actively selecting samples to label to participate in model training. Similar to selecting samples in the sample pooling in AL, Sparse MCS actively selects a few spatiotemporal subareas in the entire sensing map. In addition, sensing data at selected subareas is also similar to labeling the selected samples in AL. Therefore, we can refer to the sample selection criteria in AL and use them in Sparse MCS. In AL, the sample selection strategy generally follows three criteria: informativeness, representativeness, and diversity. The detailed definition can be found in Section II. Referring to these criteria, we propose an evaluation standard for the importance of spatiotemporal positions in Sparse MCS to actively select more important spatiotemporal positions to sense data for the training of large-scale models.

For the first challenge, we link the representativeness in AL and the correlation between spatiotemporal data by using the similarity of spatiotemporal positions. Then we provide a quantization formula for the similarity of spatiotemporal positions using the dot product operation in mathematics. For the second challenge, during the research, we discovered that the self-attention mechanism plays a crucial role in the Transformer and its variants, which can reduce the loss of network signals in the traveling and help the model better extract the dependencies between data. Consequently, to better extract the informativeness of spatiotemporal positions, we propose spatiotemporal attention and design a standard for evaluating the importance of spatiotemporal positions for models based on spatiotemporal attention.

Our work has the following contributions:

- We apply AL to Sparse MCS and propose an active sensing method. As far as we know, this is the first active sensing method for the Transformer model in Sparse MCS.
- We introduce the representativeness in AL to Sparse MCS by the data correlation at different spatiotemporal positions and provide a quantization formula for the representativeness of spatiotemporal positions.
- We propose the spatiotemporal attention mechanism to extract the information of spatiotemporal positions to the model. Based on spatiotemporal attention, we evaluate the importance of spatiotemporal positions for the model.
- We evaluate our active sensing method on two typical urban sensing tasks with three real-world data sets, which shows that our work can improve the accuracy of the Transformer-based model.

II. Related Work

A. Sparse MCS

Mobile Crowd-Sensing [1], [2], as a new data collection paradigm, uses mobile devices carried by users to perform large-scale urban sensing tasks. Based on the collected data, various sensing tasks [12] have been solved. Limited by its cost and various factors in the real world (such as geographical factors), a more practical paradigm, Sparse Mobile Crowd-Sensing [5], [6] has been proposed. Sparse MCS only needs to sense the data in a few areas, and then data inference is used to infer the data that are not sensed. In recent years, many Sparse MCS systems [13], [14] have been developed for urban sensing. Meanwhile, with the development of deep learning, Sparse MCS has started to use deep learning models to solve various complex sensing tasks with high quality. For instance, Wang et al. [15] proposed a deep learning-enabled industrial sensing, and prediction scheme based on Sparse MCS, to achieve high-precision prediction of future moments under the hypothesis of sparse historical data.

However, most existing works have only focused on data inference and data prediction, and only a few works consider the selection of spatiotemporal positions. Wang et al. [9] proposed a deep reinforcement learning-based Cell selection mechanism for Sparse MCS that used a reinforcement learning method to select area-aware data. Xie et al. [10] proposed an Active Sparse MCS scheme that included a bipartite-graph-based sensing scheduling scheme to actively determine the sampling positions in each upcoming timeslot. Wang et al. [11] used Query By Committee to select cells to be sensed in Sparse MCS. These existing works only consider the impact of spatiotemporal data on the model or algorithm, ignoring

the correlation between spatiotemporal data. Liu et al. [16] proposed a cell selection method in Sparse MCS based on AL which considered the similarity of data. But it is only for air pollution monitoring which is not suitable for other complex sensing tasks.

B. Transformer and its variants

With the development of deep learning, Sparse MCS is gradually being integrated with deep learning models to achieve good performance on the complex sensing tasks. Transformer [7], a large-scale deep learning model, was born in the NLP field. With its powerful performance advantages, Transformer has produced many variants and developed into various research fields. When dealing with time series prediction tasks, Transformer and its variants are still outstanding. Autoformer [17] is an upgraded version of the Transformer, which optimizes the original Transformer according to the feature of time series problems. Pyraformer [18] proposes a tree-structured Transformer to solve the prediction problem of long-period time series. Informer [8] optimizes the Transformer from the perspective of efficiency for long-period time series prediction. However, these deep learning models need a large number of complete time series as training data to train the model, which means a high cost in MCS.

C. Active Learning

In supervised learning, to obtain more accurate learning models, a large number of labeled data are often required to participate in the training of learning models. However, in many fields, obtaining labeled data is usually difficult, time-consuming, and expensive. To save the cost of labeling and get a high-accuracy learning model, AL [19], [20] is proposed to maximize the performance of the learning model by actively selecting more valuable samples in the unlabeled sample set for labeling. As the most important part of AL, the design of the sampling algorithm generally considers three criteria: informativeness [21], [22], representativeness [23], and diversity [24]. The informativeness means that the selected sample should contain rich information, so labeling it will greatly benefit the training of the model. Representativeness means that the selected samples can represent a group of unlabeled samples to participate in training. Diversity means that the selected samples should be distributed throughout the sample space, rather than concentrated in one place. The combination of the three standards constitutes various sample selection algorithms in AL.

III. System Model and Overview

A. System Model

In this paper, we combine the Sparse MCS with the training of the Transformer. To obtain a high-accuracy model, we need to collect fine-grained data at a limited budget from the large-scale target sensing map to train the model. First, we randomly collect training data to obtain an initialization model. To further improve the accuracy of the model, we continue to collect data from the sensing map and train the model.

Limited by the sensing budget, after the training model is initialized, there are B sensing cycles to collect the sensing data and train the model. We divide the sensing map into s subareas, and each sensing cycle has t timeslots to be sensed. Due to the budget constraints, only b spatiotemporal positions can be selected in one sensing cycle to sense the data. Therefore, in the k-th sensing cycle, we denote the spatiotemporal positions matrix $\boldsymbol{X}^k \in \mathbb{R}^{t\times s}$ where if we select the spatiotemporal position at the j-th subareas in the i-th timeslot, $x_{ij}^k = 1$, otherwise $x_{ij}^k = 0$. Similarly, the $\hat{\boldsymbol{Y}}^k \in \mathbb{R}^{t\times s}$ denotes the ground truth at each spatiotemporal position. Then we get the spatiotemporal data series matrix $\boldsymbol{Y}^k \in \mathbb{R}^{t\times s}$ in the k-th sensing cycle:

$$\boldsymbol{Y}^k = \boldsymbol{X}^k \bullet \hat{\boldsymbol{Y}}^k \tag{1}$$

where $\bullet$ represents the element-wise product. Each row of the $\boldsymbol{Y}^k$ represents a timeslot, and each column of the $\boldsymbol{Y}^k$ represents a subarea.

After collecting the sensing data, we use the data to train the model. We consider a loss function $l(\hat{z}, f_{\boldsymbol{Y}}(y))$ where y is the input of the model and $z = f(y)$ is the output from the model f, which is trained by the spatiotemporal data series matrix $\boldsymbol{Y}$ and $\hat{z}$ is the ground truth. Then in the k-th sensing cycle, the actively selecting spatiotemporal positions to sense the training data for the model can be defined as finding the selected spatiotemporal positions matrix $\boldsymbol{X}^k$:

$$\begin{aligned} &\boldsymbol{X}^k = \arg\min E_{y\in \boldsymbol{Y}_{test}}[l(\hat{y}, f_{\boldsymbol{Y}'\cup(\boldsymbol{X}^k\bullet\hat{\boldsymbol{Y}}^k)}(y))] \\ &\text{s.t.} \sum_{i=1}^{t}\sum_{j=1}^{s} x_{ij}^k = b \end{aligned} \tag{2}$$

where $\boldsymbol{Y}_{test}$ is the test set, and $\boldsymbol{Y}' \cup (\boldsymbol{X}^k \bullet \hat{\boldsymbol{Y}}^k)$ means that we use both the historical data $\boldsymbol{Y}'$ collected in previous cycles and the new data $(\boldsymbol{X}^k \bullet \hat{\boldsymbol{Y}}^k)$ to train the model.

In this paper, we focus on f being **Transformer and its variants** and propose a heuristic algorithm that considers the influence of the representativeness and informativeness of the spatiotemporal positions to actively select more important spatiotemporal positions to sense the data and train the model.

B. Overview

We now provide an example to describe the details of data collection and model training. Assume that we have a deep learning model to be trained, and Sparse MCS is used for collecting the training data. Users are recruited to sense data used for model training. Before actively selecting the spatiotemporal positions, we first randomly selected the spatiotemporal positions to collect training data $\boldsymbol{Y}^0$ used for initialization.

When we get the initial model, according to the active sensing method, we will actively select the spatiotemporal positions in the next B sensing cycles and retrain the model. In each sensing cycle, there are t timeslots and s subareas forming $s \times t$ spatiotemporal positions. The active sensing method is to select b important spatiotemporal positions to sense data for training limited by the budget. As shown in

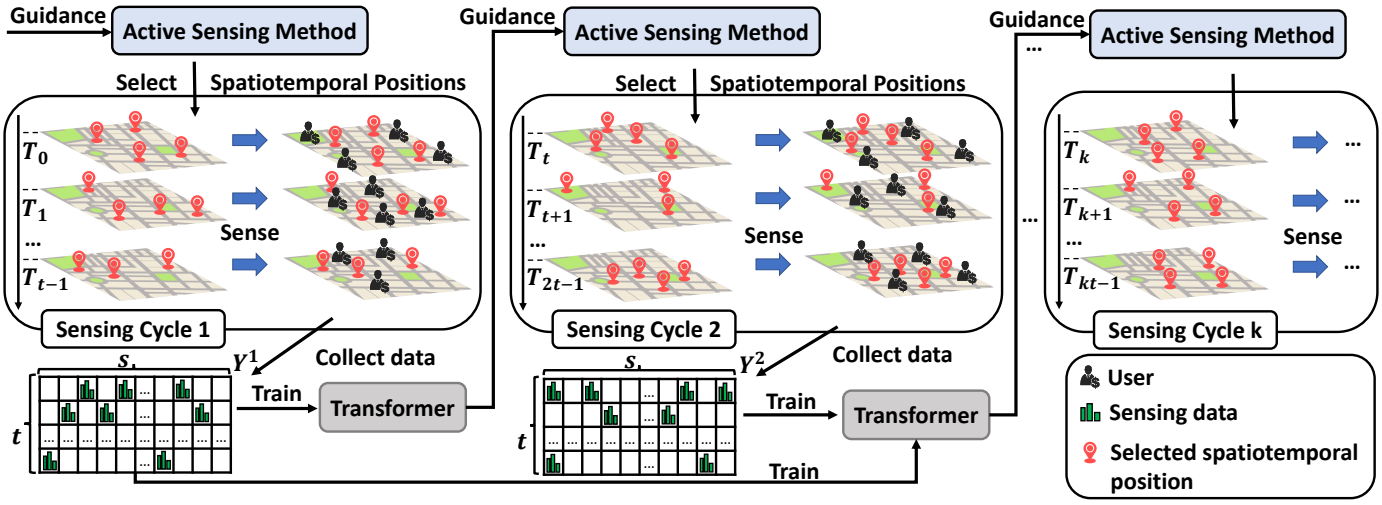


Fig. 2. The details of the sensing cycles that spatiotemporal positions are selected by the active sensing method. In each sensing cycle, we first use the active sensing method to evaluate the importance of each spatiotemporal position. Then more important spatiotemporal positions will be selected. Users will go to these spatiotemporal to sense data. Finally, all collected data will be used for training the model again. And this new model will affect the active sensing method in the next sensing cycle.

Fig.2, in the first sensing cycle, affected by the initial model, the active sensing method actively selects b spatiotemporal positions. Then sensed by recruited users, we get the spatiotemporal series matrix $\boldsymbol{Y}^1$. $\boldsymbol{Y}^0$ and $\boldsymbol{Y}^1$ are used to re-train the model. Then, in order to make the model more accurate, we start the second sensing cycle limited by the budget. In the second sensing cycle, we first utilize the training model and our proposed method to select spatiotemporal positions to be sensed at T_{t+1} to T_{2t}. Then users will go to the selected spatiotemporal positions to sense data. Using these collected data $\boldsymbol{Y}^0$, $\boldsymbol{Y}^1$, and $\boldsymbol{Y}^2$, we train the model again. Similarly, this model will affect the active sensing method to select the spatiotemporal positions in the next sensing cycle. After total B sensing cycles, we get a high-accuracy model trained by $\boldsymbol{Y}^0 \cup \boldsymbol{Y}^1 \cup ... \cup \boldsymbol{Y}^B$.

IV. METHOD

In this section, we describe the proposed active sensing method in detail. The main goal of our method is to select more important spatiotemporal positions in the sensing cycles. We evaluate the importance of the spatiotemporal positions and then choose more important positions to sense the data. The evaluation of a spatiotemporal position includes two parts: Representativeness and Informativeness.

A. Representativeness

The correlation between spatiotemporal data is one of the most typical characteristics of spatiotemporal data. Therefore, when actively sensing spatiotemporal positions, we first consider the impact of data correlations implied between them. There are many types of data correlation, such as continuity, similarity, positive correlation, negative correlation, and so on. Quantifying all these correlations can be difficult and time-consuming, especially if the data are high-dimensional or complex. Moreover, not all correlations may be relevant or important for a particular task or analysis. For example, in some cases, it may be sufficient to only focus on the strongest or most significant correlations rather than trying to quantify all correlations. Among these correlations, the similarity is always present in various sensing tasks and plays an important role. First, if the high-dimensional features of two spatiotemporal positions are similar, then their corresponding values tend to be similar as well. In addition, the similarity between samples plays an important role in AL, which is used for evaluating the representativeness of samples. In AL, when a sample has good representativeness, it can often represent a group of other samples, which is mathematically expressed as being closer to other samples in the feature space. When this sample is selected, in some aspects, other samples similar to it can also be regarded as having participated in model training. Therefore, considering the data correlations, we describe representativeness as the similarity between spatiotemporal positions. Then we should solve the problem of how to evaluate the similarity between two spatiotemporal positions.

To solve this problem, we first need to obtain the high-dimensional feature vectors of each spatiotemporal position. Generally, we can get it through the encoder of the Transformer. However, for spatiotemporal positions, the performance of traditional Transformer embedding is limited. Therefore, we have abandoned the traditional structure of position embedding. Instead, as shown in Fig.3, we expand the embedding layer with spatiotemporal position embedding and spatiotemporal feature embedding to help the Transformer better extract the information of spatiotemporal positions. Moreover, when we use the encoder to get high-dimensional features of spatiotemporal positions, the value embedding will be discarded. Then for the spatiotemporal position (t, s), we have the embedding $E_{pos}(t, s)$:

$$E_{pos}(t,s) = T_{pos}(t) + T_{feat}(t) + S_{pos}(s) + S_{feat}(s) \quad (3)$$

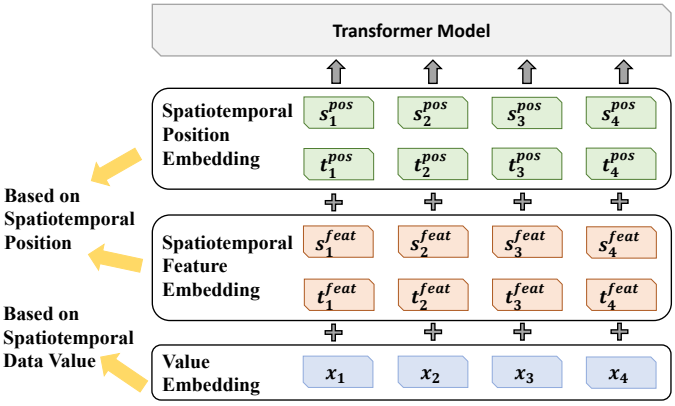


Fig. 3. The embedding layer of the Transformer, which is used for spatiotemporal data. When we use the encoder to get high-dimensional features of spatiotemporal positions, the value embedding will be discarded, and we will only use the information of the spatiotemporal position including the position embedding for the physical spatiotemporal position and the feature embedding for additional feature information on the spatiotemporal position.

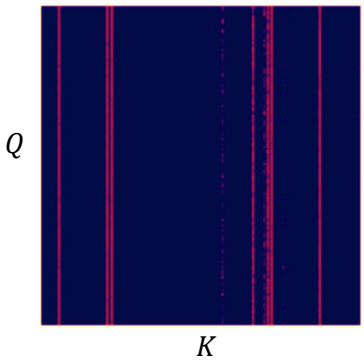


Fig. 4. The heat map of the attention scores, which is the sum of the attention scores from all heads at the first layer of the encoder. The more the area tends to black, the smaller the value of the area is. The more the area tends to red, the bigger the value of the area is.

where $T_{pos}(t)$ and $S_{pos}(t)$ are the position embedding for the physical spatiotemporal position, $T_{pos}(t)$ and $T_{pos}(t)$ are the feature embedding for additional feature information like vacations, seasons, and so on. To get the feature of the spatiotemporal position, we enter the embedding result of position to the encoder of the Transformer. For the spatiotemporal position (t, s), we have its feature $\boldsymbol{F}_{t,s}$:

$$\boldsymbol{F}_{t,s} = Encoder(E_{pos}(t, s)) \tag{4}$$

where $Encoder(\cdot)$ represents the encoder of Transformer. The details of the encoder will be introduced in the next subsection.

When we get the features of spatiotemporal positions, we measure the similarity of each other. Here, in order to have the same scale as the subsequent informativeness, we use the dot product that is also applied in the self-attention mechanism to measure the similarity of the features. Assume that N is a set that includes all spatiotemporal positions, and $\boldsymbol{F}_i$ and $\boldsymbol{F}_j$ are the d-dimension feature vectors of the spatiotemporal position $\boldsymbol{F}_{t_i,s_i}$ and $\boldsymbol{F}_{t_j,s_j}$. According to the dot product results of spatiotemporal position i and all other spatiotemporal positions, we get the representative score $Rep(i)$:

$$Rep(i) = \sum_{i \neq j, j \in N} \frac{\boldsymbol{F}_i \boldsymbol{F}_j^T}{\sqrt{d}} \tag{5}$$

where the product of $\boldsymbol{F}(i)$ and $\boldsymbol{F}(j)$ describes the similarity of i and j. By summing $\boldsymbol{F}(i)$ and the product of features at all other positions, we obtain the representativeness of the spatiotemporal position i.

B. Informativeness

In AL, the informativeness describes the importance of the sample to the model, which is often closely related to the model itself. Therefore, in order to better evaluate the informativeness of the spatiotemporal positions, we focus on the self-attention mechanism, which is a key component of the Transformer. For spatiotemporal data, in order to help the model extract more spatiotemporal information, we propose a spatiotemporal attention based on the traditional self-attention mechanism.

1) Spatiotemporal Attention: Self-attention is a particular implementation of the attention mechanism, which is used in Transformer to extract dependencies between input data. For spatiotemporal data, to better extract information between them, combined with the spatiotemporal embedding proposed in the previous subsection, we used a spatiotemporal attention mechanism. When the spatiotemporal series $\boldsymbol{X}$ passes through the embedding layer as shown in Fig.3, we get $\boldsymbol{X}_{feed} \in \mathbb{R}^{T \times S \times d}$:

$$\boldsymbol{X}_{feed} = Value(\boldsymbol{X}) + E_{pos}(\boldsymbol{X}^t, \boldsymbol{X}^s) \tag{6}$$

In the multi-head attention layer, for the i-th head, $\boldsymbol{X}_{feed}$ is mapped into three vectors: the query vector $\boldsymbol{Q}_i$, the key vector $\boldsymbol{K}_i$, and the value vector $\boldsymbol{V}_i$ through three learnable weight matrices $\boldsymbol{W}_i^q$, $\boldsymbol{W}_i^k$, $\boldsymbol{W}_i^v$:

$$\boldsymbol{Q}_i, \boldsymbol{K}_i, \boldsymbol{V}_i = \boldsymbol{X}_{feed}(\boldsymbol{W}_i^q, \boldsymbol{W}_i^k, \boldsymbol{W}_i^v) \tag{7}$$

After that, we use the formula of Scaled Dot-Product Attention to calculate the attention scores $\boldsymbol{S}_i$:

$$\boldsymbol{S}_i = softmax(\frac{\boldsymbol{Q}_i \boldsymbol{K}_i^T}{\sqrt{d_k}}) \tag{8}$$

where the multiplication of $\boldsymbol{Q}_i$ and $\boldsymbol{K}_i$ is to obtain the similarity between spatiotemporal data and then is divided by $\sqrt{d_k}$ to prevent the result of the softmax function from being unbalanced caused by the too large product. Then we use the score $\boldsymbol{S}_i$ to enhance the representation of the spatiotemporal data:

$$Attention_i(\boldsymbol{Q}_i, \boldsymbol{K}_i, \boldsymbol{V}_i) = \boldsymbol{S}_i \boldsymbol{V}_i \tag{9}$$

After all heads are computed, we concatenate the results into a matrix and multiply it with a learnable weight matrix W.

Algorithm 1: Actively Collection for Model Training

Input: $B, b, \boldsymbol{Y}^0, \boldsymbol{X}$
Output: f
initialize: Initial model f is trained by $\boldsymbol{Y}^0$, $k = 1$;
$\boldsymbol{Y}_{train} = \boldsymbol{Y}^0$;
while $k \leq B$ **do**
  $\boldsymbol{X}^k$ = $\boldsymbol{X}[k]$;
  for $\boldsymbol{X}_i \in \boldsymbol{X}^k$ **do**
    $\boldsymbol{S}_i$ =**Importance Evaluation**$(\boldsymbol{X}_i, \boldsymbol{X}^k, f)$
  end
  for $n = 1$ ***to*** b **do**
    $i = \arg\max \boldsymbol{S}$;
    $\boldsymbol{X}_i^k = 1$, $\boldsymbol{S}_i = 0$;
  end
  sense data $\boldsymbol{Y}^k$ **with** $\boldsymbol{X}^k$;
  $\boldsymbol{Y}_{train} = \boldsymbol{Y}_{train} \cup \boldsymbol{Y}^k$;
  train f by $\boldsymbol{Y}_{train}$;
  $k = k + 1$
end

After n encoder layers, we get high-dimensional feature vector $Feature(X)$ from the encoder:

$$Feature(\boldsymbol{X}) = Concat(h_1, h_2, ..., h_k)\boldsymbol{W} \tag{10}$$

$$h_i = Attention_i(\boldsymbol{Q}_i, \boldsymbol{K}_i, \boldsymbol{V}_i) \tag{11}$$

where we ignore the previous encoder layers for simplification.

2) Informativeness Evaluation: To evaluate the informativeness of the spatiotemporal positions, we are focused on the self-attention mechanism, which plays an important role in Transformer. As mentioned before, when we get the attention score of the data, it will be used to enhance the representation of the spatiotemporal data. If the result of multiplying the vector $\boldsymbol{q}_i$ of the data i and the vector $\boldsymbol{k}_j$ of the data j is a big value after the softmax() function, the data j will have a high weight in the enhanced representation of i, which means great information for the model to represent i. So when we get the true value of j, the representation of i will be more accurate.

Meanwhile, it is found in Informer [8] that the attention score presents a long-tail distribution, and it applies to spatiotemporal data. As shown in Fig.4, for a real-world data set about PM2.5, we get the heat map of the attention scores, which is the sum of the attention scores from all heads at the first layer of the encoder. The few bright verticals indicate that the multiplications of query vector $\boldsymbol{q}$ of many spatiotemporal data and key vector $\boldsymbol{k}$ of a few spatiotemporal data have high scores, which means the corresponding spatiotemporal data to the key vector $\boldsymbol{k}$ is very helpful for the model to represent the high-dimensional feature. Therefore, the attention score reflects the informativeness of spatiotemporal data in the model. Based on the informativeness of the spatiotemporal data, we evaluate the informativeness of the spatiotemporal positions. Because before sensing we don't know the true value of spatiotemporal data, we don't use the value embedding.

Algorithm 2: Importance Evaluation

Input: $\boldsymbol{X}_i, \boldsymbol{X}^k, f$
Output: s
initialize: $inf = 0$, $rep = 0$, $s = 0$;
$\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{F}, scale, l = f(\boldsymbol{X}^k)$;
for $\boldsymbol{X}_j \in \boldsymbol{X}^k$ **do**
  if $i \neq j$ **then**
    $inf = inf + \boldsymbol{Q}_j \cdot \boldsymbol{K}_i^T / scale / l$;
    $rep = rep + \boldsymbol{F}_i \cdot \boldsymbol{F}_j^T / scale$;
  end
end
$s = inf + rep$;

We input the spatiotemporal positions to the encoder of the Transformer trained in the previous sensing cycle. When the spatiotemporal position i reach the attention layer, we have its informativeness $Inf(i)$:

$$Inf(i) = \sum_{l} \sum_{i \neq j, j \in N} \frac{\boldsymbol{q}_j \boldsymbol{k}_i^T}{\sqrt{d_k} * l} \tag{12}$$

where N is the set of all spatiotemporal positions, and l is the number of encoder layers. The reason for dividing by l is to have the same scale as $Rep(i)$.

C. Actively Select

Now we get the evaluation of the importance of the spatiotemporal positions, that is:

$$S(i) = Rep(i) + Inf(i) \tag{13}$$

At first, we randomly collect the spatiotemporal data to initialize the model. Then, we use the encoder of the Transformer trained in the previous sensing cycle to evaluate the importance of the spatiotemporal positions by (13). After getting the importance scores, we actively select the spatiotemporal positions with the top b importance scores and recruit users to sense training data. Then we retrain the model with the collected data. Algorithm 1 and 2 show the process in detail where B is the number of sensing cycles, b is the sensing budget for one sensing cycle, $\boldsymbol{Y}_0$ is the initial data set to train the initial model f, and $\boldsymbol{X}$ is a list that contains all positions for all sensing cycles. Finally, we get a high-accuracy model f.

D. Incremental Training

As the sensing cycle increases, more and more training data is collected and involved in model training. If the model is retrained by using all collected data in each training stage, a huge training cost will be incurred. For this, we propose an incremental training method that uses the parameters of the previous training stage to warm start the new training stage. Inspired by [25], [26], we first set the learning rate to a big value to help the model jump out of the local optimum and then gradually decrease the value like [25] to enter another local optimum.

TABLE I
STATISTICS OF THREE EVALUATION DATA SETS

Task Type	***Urban Environment***		***Urban traffic***
Data Sets	Sensor-Scope	U-air	Traffic Volume Viewer
City	Lausanne(CHE)	BJ(CHN)	NSW(AUS)
Data	Humidity	PM2.5	Traffic flow
Subareas	57 subareas	36 subareas	30 checkpoints
Cycle	$0.5h$	$1h$	$1d$
Duration	$7d$	$11d$	$1y$
Mean	84.52	79.11	19095.73
Std. Dev.	6.32	81.21	26750.79
Unit	%	$\mu g/m3$	n

V. EXPERIMENTS

A. Data Set

To evaluate the performance of the active sensing method we proposed, we determine the learning model as the Transformer-based data inference model in Sparse MCS and conduct experiments on three real-world data sets. These data sets include the Sensor-Scope and U-Air about the urban environment sensing and the Taxi-Speed about the urban traffic sensing. We show the main information in Table I with more details as follows:

- The **Sensor-Scope**[1] is an environmental data set, including temperature, humidity, and other variables. This data set collects data regularly through a lot of static sensors deployed on the EPFL campus. We use humidity data to evaluate our proposed method.
- The **U-Air**[2] is an air quality data set, which includes $PM2.5$, SO_2, and other variables. It collects data from monitor stations deployed in Beijing, China.
- The **Traffic Volume Viewer**[3] is a set of traffic flow information, which is collected by sensors deployed at the traffic collection station in New South Wales, Australia. It monitors traffic flow, such as the number or type of vehicles at more than 60 stations.

It is worth noting that some of these data sets are collected from static sensors, but in fact, we can use mobile devices or social media to collect these data dynamically. In addition, these data sets are typical urban MCS tasks. Therefore, it is effective and reasonable to use these data sets for evaluation.

B. Experimental Settings

1) Data Inference Model: To evaluate the performance of our proposed method, we determine the learning model as the Transformer-based data inference model in Sparse MCS. In order to better extract the dependency between spatiotemporal data, we change the embedding layer in the traditional Transformer to a new embedding layer, as shown in Fig.3. Meanwhile, there are other settings about the Transformer model: the model dimension d_{model} is 512; the number of attention head is 8; the number of encoder layers is 3. RMSE is used as the loss function to measure each method.

2) Data Set Settings: we split all data sets at ratio 7:2:1 into training sets, validation sets, and test sets by the time, then we normalize the data into range [-1, 1] and feed the normalized data into the model.

C. Model Evaluation

To comprehensively evaluate the active sensing method we proposed, we first evaluate the performance of the model. Therefore, we have compared several typical data inference methods:

- **BGMC-st** [27] is a matrix completion-based algorithm, which uses low-rank attributes and spatiotemporal constraints to infer data.
- **KNN-S** and **KNN-T** are variants of KNN algorithm. With the help of spatiotemporal relationship, it uses the weighted average of k data sensed from the nearest subregion as the inferred value to fill in the missing value.
- **DMF** [15] is the Matrix factorization based on the deep learning multi-layer network, which can better use the nonlinear characteristics of the matrix through multi-layer neural networks compared to the traditional method.

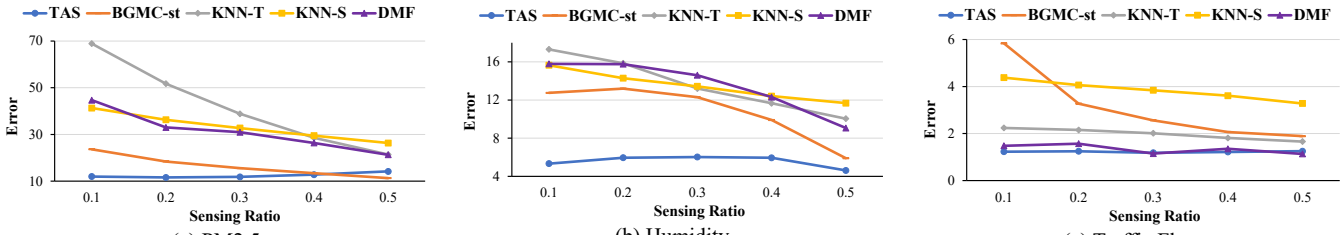
To evaluate the data inference model, we test the effect of the sensing ratio on model performance. It is because that budget has always been one of the important factors in Sparse MCS, and a smaller budget often means a smaller perception ratio for the entire perception map. Therefore, the performance of the model under different sensing ratios is the key indicator to evaluate the model. We set five perception ratios from 10% to 50% and actively sense sparse data from the complete sensing map according to the sensing ratio. Then we use the sensing data to train the model. The experiment results are shown in Fig.5.

The experimental results show that for all data sets, compared with other inference methods, TAS (**T**ransformer based on **A**ctive **S**ensing method) has obviously better performance in general, especially on the Humidity data set. On the PM2.5 data set, when the sensing ratio is relatively large, BGMC-st and TAS have comparable performance, but as the sensing ratio decreases, the performance gap between the two gradually widens, and the TAS effect is better. On the Traffic Flow data set, there is a large gap in the performance of each method. For convenience, we standardized the error results in all experiments in this paper. The performance of DMF and TAS are very good, but DMF is not as effective as TAS in other data sets. Therefore, the experimental results undoubtedly show Transformer's powerful feature extraction capability and indirectly prove the necessity of our paper that designing an active sensing method for the Transformer model.

[1]http://sensorscope.epfl.ch/network code

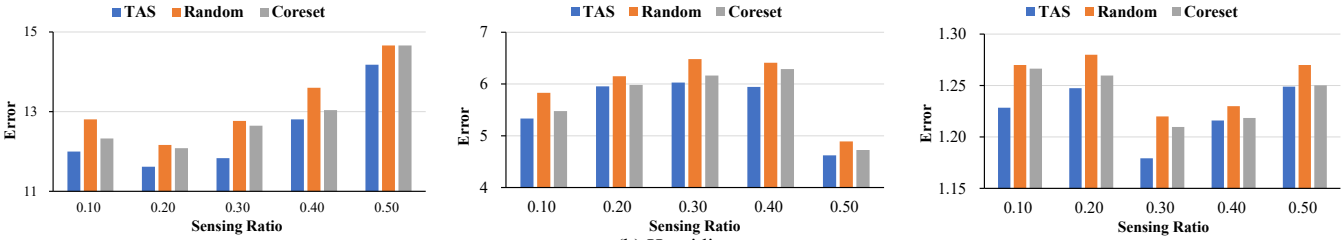
[2]https://www.microsoft.com/en-us/research/project/urban-computing/

[3]https://www.rms.nsw.gov.au/about/corporate-publications/statistics/traffic-volumes/aadt-map/



(a) PM2.5 (b) Humidity (c) Traffic Flow

Fig. 5. Inference accuracy under different sensing ratios with different inference methods.



(a) PM2.5 (b) Humidity (c) Traffic Flow

Fig. 6. Inference accuracy under different sensing ratios with different sensing methods.

D. Method Evaluation

In the previous section, we use the proposed active sensing method to collect data and train the model and then compare it with other typical data inference methods to prove the performance of the model. In this section, we will evaluate the performance of the proposed active sensing method. For the data inference model based on Transformer, we have the following two methods to sense the training data:

- **Random**: Random collecting is the most common sensing method for Sparse MCS. It randomly selects subareas in the entire sensing map.
- **Coreset** [24]: Coreset is one of the typical sample selection methods in AL. It equates the error to the coverage radius of the feature space through mathematical derivation. Therefore, the strategy of this method is to select a batch of samples (data) so that this batch of samples (data) can cover the entire feature space with the minimum coverage radius.

Some of the active sensing methods mentioned in the previous sections can only handle specific sensing tasks, and some are not applicable to the Transformer model. Therefore, we don't choose these methods for comparison. In addition, Coreset is proposed for CNN. To generalize it to the Transformer model, we change the distance calculation method but don't change its essence, so it will not affect its performance. Fig.6 shows the experiment results.

As the results show, TAS has more significant advantages than other sensing methods. As the basis of Sparse MCS, Random has the worst effect compared with the other two methods. This is because it neither focuses on the amount of information in the data itself nor does it consider the correlation and distribution of data. Coreset is better than Random but not as good as TAS. This is because it only considers the correlation and distribution of data when selecting sensing positions, which makes the selected points more evenly distributed in the feature space. However, it does not consider the impact of the data itself on the model. In addition, when we get the feature of spatiotemporal positions, the time complexity of Coreset is about the third power of TAS.

TABLE II
ERROR INCREASE COMPARED TO TAS ON THREE DATA SET

Sensing Ratio	0.10	0.20	0.30	0.40	0.50
TAS-Inf	5.12%	1.34%	4.04%	4.28%	3.68%
TAS-Rep	3.44%	1.36%	2.68%	2.91%	2.74%
Random	9.33%	3.29%	7.49%	7.82%	5.84%
TAS-Inf	5.41%	3.64%	7.24%	6.50%	2.21%
TAS-Rep	1.38%	1.47%	2.12%	2.41%	1.94%
Random	6.72%	4.70%	7.90%	6.21%	3.42%
TAS-Inf	4.93%	1.03%	5.29%	1.51%	2.12%
TAS-Rep	2.25%	1.03%	2.98%	0.91%	1.50%
Random	6.89%	5.30%	7.03%	2.32%	3.39%

E. Ablation Study

In our paper, the importance of spatiotemporal position i has two parts: Informativeness $Inf(i)$ and Representativeness $Rep(i)$. In this section, we use an ablation study to verify the impact of each part on the overall method. We have:

- **TAS-Inf**: When evaluating the importance of data, only the informativeness $Inf()$ is considered, and the representativeness $Rep()$ is ignored.
- **TAS-Rep**: When evaluating the importance of data, only the representativeness $Rep()$ is considered, and the informativeness $Inf()$ is ignored.

The results of the ablation study are shown in Table II. The data sets corresponding to the results in Table II are PM2.5, Humidity, and Traffic Flow from top to bottom. When we change the TAS to TAS-Inf and TAS-Rep, the error of the model increases but is less than Random. This means that both $Inf()$ and $Rep()$ play a role in the selection method, but not as much as their combination. Moreover, the results show that TAS-Rep performs better than TAS-Inf. This may be because the long-tailed distribution of attention scores makes only a few spatiotemporal positions particularly important. Therefore, the selection of TAS-Inf after selecting these important data is casual. TAS-Rep considered the correlation between data, which is more comprehensive, so its selection is more uniform. TAS considers these two aspects, which makes its performance the best.

VI. Conclusion

In this paper, we propose an active sensing method in Sparse MCS to collect data for training the Transformer-like models. Inspired by the sample selection criteria in AL, we consider the representativeness and informativeness of the spatiotemporal positions. For the former, we focus on the similarity from the data correlation and use the similarity of spatiotemporal position features to evaluate the representativeness of spatiotemporal positions. For the latter, to better extract the dependency between data, we changed the traditional self-attention to the spatiotemporal attention for spatiotemporal data, and then we used the characteristics of the attention scores in Transformer to quantify the informativeness. Finally, we use three data sets in the real world to evaluate the performance of our proposed method.

Acknowledgment

This work is supported in part by National Key R&D Program of China under Grant Nos. 2021ZD0112501 and 2021ZD0112502, and National Natural Science Foundation of China under Grant Nos. 62272193, 62102161 and 61972450, and CCF-Baidu Open Fund (No.2021PP15002000).

References

[1] Raghu K Ganti, Fan Ye, and Hui Lei. Mobile crowdsensing: current state and future challenges. *IEEE communications Magazine*, 49(11):32–39, 2011.

[2] Huadong Ma, Dong Zhao, and Peiyan Yuan. Opportunities in mobile crowd sensing. *IEEE Communications Magazine*, 52(8):29–35, 2014.

[3] Daqing Zhang, Leye Wang, Haoyi Xiong, and Bin Guo. 4w1h in mobile crowd sensing. *IEEE Communications Magazine*, 52(8):42–48, 2014.

[4] Jiafu Wan, Jianqi Liu, Zehui Shao, Athanasios V Vasilakos, Muhammad Imran, and Keliang Zhou. Mobile crowd sensing for traffic prediction in internet of vehicles. *Sensors*, 16(1):88, 2016.

[5] Leye Wang, Daqing Zhang, Yasha Wang, Chao Chen, Xiao Han, and Abdallah M'hamed. Sparse mobile crowdsensing: challenges and opportunities. *IEEE Communications Magazine*, 54(7):161–167, 2016.

[6] Shiting Zhao, Guozi Qi, Tengjiao He, Jinpeng Chen, Zhiquan Liu, and Kaimin Wei. A survey of sparse mobile crowdsensing: Developments and opportunities. *IEEE Open Journal of the Computer Society*, 2022.

[7] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[8] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 11106–11115, 2021.

[9] Leye Wang, Wenbin Liu, Daqing Zhang, Yasha Wang, En Wang, and Yongjian Yang. Cell selection with deep reinforcement learning in sparse mobile crowdsensing. In *2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS)*, pages 1543–1546, 2018.

[10] Kun Xie, Xiaocan Li, Xin Wang, Gaogang Xie, Jigang Wen, and Dafang Zhang. Active sparse mobile crowd sensing based on matrix completion. In *Proceedings of the 2019 International Conference on Management of Data*, SIGMOD '19, page 195–210, New York, NY, USA, 2019. Association for Computing Machinery.

[11] Leye Wang, Daqing Zhang, Dingqi Yang, Animesh Pathak, Chao Chen, Xiao Han, Haoyi Xiong, and Yasha Wang. Space-ta: Cost-effective task allocation exploiting intradata and interdata correlations in sparse crowdsensing. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 9(2):1–28, 2017.

[12] Xiaoyu Zhu, Yueyi Luo, Anfeng Liu, Wenjuan Tang, and Md. Zakirul Alam Bhuiyan. A deep learning-based mobile crowdsensing scheme by predicting vehicle mobility. *IEEE Transactions on Intelligent Transportation Systems*, 22(7):4648–4659, 2021.

[13] Rajib Kumar Rana, Chun Tung Chou, Salil S Kanhere, Nirupama Bulusu, and Wen Hu. Ear-phone: an end-to-end participatory urban noise mapping system. In *Proceedings of the 9th ACM/IEEE international conference on information processing in sensor networks*, pages 105–116, 2010.

[14] Anqi Liu, Changle Li, Wenwei Yue, and Xun Zhou. Real-time traffic prediction: A novel imputation optimization algorithm with missing data. In *2018 IEEE Global Communications Conference (GLOBECOM)*, pages 1–7, 2018.

[15] En Wang, Mijia Zhang, Xiaochun Cheng, Yongjian Yang, Wenbin Liu, Huaizhi Yu, Liang Wang, and Jian Zhang. Deep learning-enabled sparse industrial crowdsensing and prediction. *IEEE Transactions on Industrial Informatics*, 17(9):6170–6181, 2020.

[16] Tong Liu, Yanmin Zhu, Yuanyuan Yang, and Fan Ye. Alc2 : When active learning meets compressive crowdsensing for urban air pollution monitoring. *IEEE Internet of Things Journal*, 6(6):9427–9438, 2019.

[17] Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in Neural Information Processing Systems*, 34:22419–22430, 2021.

[18] Shizhan Liu, Hang Yu, Cong Liao, Jianguo Li, Weiyao Lin, Alex X Liu, and Schahram Dustdar. Pyraformer: Low-complexity pyramidal attention for long-range time series modeling and forecasting. In *International conference on learning representations*, 2021.

[19] Burr Settles. *Active learning literature survey*. University of Wisconsin-Madison Department of Computer Sciences, 2009.

[20] Anita Krishnakumar. Active learning literature survey. *Tech. rep., Technical reports, University of California, Santa Cruz.*, 42, 2007.

[21] Neil Houlsby, Ferenc Huszár, Zoubin Ghahramani, and Máté Lengyel. Bayesian active learning for classification and preference learning. *arXiv preprint arXiv:1112.5745*, 2011.

[22] Naoki Abe. Query learning strategies using boosting and bagging. *Proc. of 15ˆ¡ th¿ Int. Cmf. on Machine Learning (ICML98)*, pages 1–9, 1998.

[23] Dongrui Wu. Pool-based sequential active learning for regression. *IEEE transactions on neural networks and learning systems*, 30(5):1348–1359, 2018.

[24] Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. *arXiv preprint arXiv:1708.00489*, 2017.

[25] Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations*, 2017.

[26] Wentao Zhang, Yu Shen, Yang Li, Lei Chen, Zhi Yang, and Bin Cui. Alg: Fast and accurate active learning framework for graph convolutional networks. In *Proceedings of the 2021 International Conference on Management of Data*, SIGMOD '21, page 2366–2374, New York, NY, USA, 2021. Association for Computing Machinery.

[27] Wenbin Liu, Yongjian Yang, En Wang, and Jie Wu. Fine-grained urban prediction via sparse mobile crowdsensing. In *2020 IEEE 17th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*, pages 265–273. IEEE, 2020.