MetaGeo: A General Framework for Social User Geolocation Identification With Few-Shot Learning

Fan Zhou, *Member, IEEE*, Xiuxiu Qi, Kunpeng Zhang, Goce Trajcevski, *Member, IEEE*, and Ting Zhong

***Abstract*— Identifying the geolocation of social media users is an important problem in a wide range of applications, spanning from disease outbreaks, emergency detection, local event recommendation, to fake news localization, online marketing planning, and even crime control and prevention. Researchers have attempted to propose various models by combining different sources of information, including text, social relation, and contextual data, which indeed has achieved promising results. However, existing approaches still suffer from certain constraints, such as: 1) a very few samples are available and 2) prediction models are not easy to be generalized for users from new regions—which are challenges that motivate our study. In this article, we propose a general framework for identifying user geolocation—MetaGeo, which is a meta-learning-based approach, learning the prior distribution of the geolocation task in order to quickly adapt the prediction toward users from new locations. Different from typical meta-learning settings that only learn a new concept from few-shot samples, MetaGeo improves the geolocation prediction with conventional settings by ensembling numerous mini-tasks. In addition, MetaGeo incorporates probabilistic inference to alleviate two issues inherent in training with few samples: location uncertainty and task ambiguity. To demonstrate the effectiveness of MetaGeo, we conduct extensive experimental evaluations on three real-world datasets and compare the performance with several state-of-the-art benchmark models. The results demonstrate the superiority of MetaGeo in both the settings where the predicted locations/regions are known or have not been seen during training.**

***Index Terms*— Bayesian learning, few-shot learning, geolocation, meta-learning, semisupervised learning.**

I. Introduction

With the increased popularity of social media services, such as Twitter, Facebook, Wikipedia, and Instagram, users have enabled the generation of unprecedented volumes of heterogeneous data (e.g., posted texts, shared images, and social network structures). Due to various reasons, such as privacy concerns (enforced by regulations and/or platform policies) and social media usage habits, most users are not willing to explicitly disclose their physical geolocations. Studies have shown that only a very small portion of posted information is published with explicit geotags (e.g., only 1% of the tweets are recorded with real-time locations) [1]. Identifying geolocation is a crucial component in a wide range of downstream online applications—such as emergency location identification, disease outbreak prediction, political election, local event/place recommendation, crime control, and prevention [2]. However, the users may reveal their geolocations implicitly via their published content—e.g., by mentioning the places they visited or the cities they lived—which makes the prediction of users' physical geolocations possible. For example, people from Los Angles may frequently mention "Lakers" or "Clippers" and those from Houston may often publish tweets containing the word "Rockets." As such, the problem of identifying user *geolocation* (or *geocoding*) from implicit sources has received tremendous attention from both academia and industry in the past decade [1], [3]–[7].

Existing studies have attempted to propose many machine learning models by combining various types of information, including text [8]–[12], online social relations [13]–[15], and certain contextual data, such as posting time [16], self-declared profiles [17], [18], and searching behavior [19]—all of which have enabled achieving promising results in geolocation identification for social media users. Since single-view learning does not ensure accurate geolocation, researchers have turned to unifying multiple features and paradigms. For example, multi-entry neural network (MENEXT) [5] incorporates various types of features extracted from texts, the user interactions and temporality in time zones into their learning models (e.g., doc2vec [20] and node2vec [21]). More recently, a multiview geolocation model that jointly learns from tweet content and social networks was presented in [1], based on graph neural networks [22].

Despite achieving significant progress, especially when combined with deep learning and graph learning, existing approaches still suffer from certain constraints.

(**C1**) The training data (e.g., the geotagged data) are extremely sparse, which means that some regions/locations have very few or no users associated. Though transductive learning-based models, such as GCNs [22] may, to some extent, alleviate this problem, the results are far from satisfactory because of their imbalanced characteristics [1]. Taking the `Twitter-World` data [3] for example, the number of users in different regions is extremely imbalanced and follows a heavy-tail distribution, as shown in Fig. 1(a), where more than 200 target regions have less than 80 users.

Manuscript received 6 January 2020; revised 19 October 2020, 7 April 2021, 2 August 2021, and 29 November 2021; accepted 18 February 2022. Date of publication 8 March 2022; date of current version 30 October 2023. This work was supported in part by the National Natural Science Foundation of China under Grant 62176043 and Grant 62072077 and in part by National Science Foundation Spectrum and Wireless Innovation enabled by Future Technologies (SWIFT) under Grant 2030249. *(Corresponding author: Ting Zhong.)*

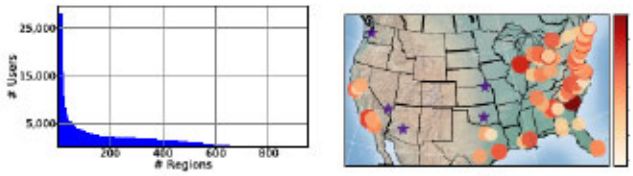
Fan Zhou, Xiuxiu Qi, and Ting Zhong are with the School of Information and Software Engineering, University of Electronic Science and Technology of China, Chengdu 610054, China (e-mail: fan.zhou@uestc.edu.cn; xiuxqihm@gmail.com; zhongting@uestc.edu.cn).

Kunpeng Zhang is with the Department of Decision, Operations and Information Technologies, University of Maryland, College Park, MD 20742 USA (e-mail: kpzhang@umd.edu).

Goce Trajcevski is with the Department of Electrical and Computer Engineering, Iowa State University, Ames, IA 50011 USA (e-mail: gocet25@iastate.edu).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TNNLS.2022.3154204.

Digital Object Identifier 10.1109/TNNLS.2022.3154204



(a) (b)

Fig. 1. (a) Distribution of users per region from the Twitter-World dataset [3] with 930 regions. The x-axis: the number of regions a user visits. (b) Visited geographical regions on GeoText [8]. The dark the color, the more users are associated with the region. The best view is in colors.

(**C2**) Existing methods are not able to handle the scenario where only a few samples are available in each region.

(**C3**) Existing methods predict the location of individual users independently in a (semi-)supervised manner, which, consequently cannot generalize the knowledge learned from the training to *new* regions which are unseen during training. Fig. 1(b) plots the geographical location of users in the GeoText dataset [8]. The *star*-labeled regions are far from those in the east/west coast (*circles*) and have fewer users visiting, which makes inferring the location of users in these regions more challenging.

Many efforts have been made to alleviate the data imbalance issue. Conventional approaches focus on designing data re-sampling strategies, over-sampling the tail classes or down-sampling the head classes [23], cost-sensitive feature representation learning [24], and class-balanced loss designs [25]. However, these methods either need to manually set a specific form of weighting function based on certain assumptions on training data or involve hyperparameters, increasing the difficulty of application and reducing performance stability. Recently, meta-learning has emerged as a promising way of addressing the class-imbalanced learning problem in the community. For example, recent works [26], [27] propose to automatically learn implicit or explicit loss-weight function parameterized by neural networks from data in a meta-learning manner, while the Jamal *et al.* [28] augment the classic class-balanced learning by explicitly estimating the differences between the class-conditioned distributions with a meta-learning approach.

Inspired by recent advances in few-shot learning and domain adaptation methods dealing with the class imbalance issue, we present a novel user geolocation identification model, called MetaGeo, relying on the meta-learning paradigm to address the constraints (**C1–C3**) in existing methods. MetaGeo consists of a text view component and a fast graph neural network to jointly capture: 1) content and network features and 2) their latent relations. By optimizing the model over the distribution of few-shot tasks, MetaGeo can quickly adapt to regions with only a few available users. It learns a well-generalized model initialization from a variety of geolocation tasks and aggregates the contextual information from source regions to predict the regions with few samples. Instead of relying on the transductive learning of graph neural networks, MetaGeo fine-tunes the task learners to improve the capability of adapting to new regions and learn transferable knowledge for regions that have not been seen during training [e.g., the star-labeled ones in Fig. 1(b)]. In summary, the main contributions of this article are fourfold.

1) We provide a novel perspective of identifying the geolocation of users on social media by leveraging the meta-learning paradigm to learn a prior distribution over tasks with a small number of users and their generated content.
2) We propose a novel framework—MetaGeo—to tackle the user geolocation prediction problem where we theoretically introduce a probabilistic graphical inference for parameter updating. In addition to faster adaptation to new regions, MetaGeo also takes task uncertainty and ambiguity into account during training.
3) MetaGeo learns from regions general knowledge encoding both tweet content and social network features and transfers the knowledge to those regions with few users, which leads to significant performance improvement for unseen regions.
4) Our extensive experiments demonstrate that MetaGeo achieves superior performance over several existing state-of-the-art geolocation prediction models on three benchmark datasets while being able to recognize users from new regions with a few samples, which is extremely difficult for the previous methods.

The following is the roadmap of the rest of this article. We provide the necessary background and introduce the formalisms used in the rest of this article in Section II. The general framework of MetaGeo and the details of implementation for adapting it to user geolocation identification are discussed in detail in Section III. Our experiments, including the setup, datasets, and evaluation against baselines, are reported in Section IV. We overview the related work and position our contributions in that context in Section V. Finally, we provide concluding remarks and outline directions for future work in Section VI.

II. Preliminaries

We now define the problem settings and objectives, followed by a concise introduction of the necessary background with respect to generating geographical regions and the building of interaction networks, and the content representation. Finally, we discuss the background in feature learning.

A. Problem Definition

The main objective of our study is to enable home location prediction of the users of social networks (in our case, Twitter users). We follow [2] and consider it as a multiclass classification problem.

More specifically, let $\mathcal{Y} = \{y_1, \ldots, y_m\}$ denote a set of m geographical regions, and $\mathcal{V} = \{v_1, \ldots, v_n\}$ denote the set of n users. The objective of the *user geolocation prediction problem* is to identify the home locations of users in $\mathcal{V}$—and we propose to achieve it by training a classification model f_θ with parameters θ on the user-generated content (tweets).

B. Data Preprocessing

We realize MetaGeo extracting textual information and forming user–user interactions network from the user-generated content. We note that although Twitter is used as the context to illustrate our study, MetaGeo can be easily generalized to other similar sources of data and is flexible to incorporate other features (e.g., tweeting-time, registration timezone, search terms, and so on).

1) Geographical Regions: As the location of a Twitter user is indicated by latitude and longitude, we need to partition locations into a set of discrete regions. Following [29], we use *k-d* tree to generate discretized regions with finer resolutions in popular areas.

2) User–User Interaction Network: The user–user interaction network is constructed from user mentioning tags. Following [12], user interaction matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ (symmetric adjacency matrix) is build based on the *collapsed* @-mention graph $\mathcal{G} = (\mathcal{V}, \mathbf{A})$ among users. Each node $v \in \mathcal{V}$ corresponds to a particular user. An edge $a_{i,j}$ is an implicit link between two users v_i and v_j created when: 1) user $v_i \in \mathcal{V}$ mentions directly user $v_j \in \mathcal{V}$ and (2) two users $v_i \in \mathcal{V}$ and $v_j \in \mathcal{V}$ co-mention a third user v_k (even v_k might not be in $\mathcal{V}$, we still add an edge to avoid the connection sparsity [12]). Furthermore, because celebrities are very frequently mentioned by others, to avoid any biased connections derived from these celebrities, we ignore mentions of celebrities when constructing $\mathbf{A}$.

C. Feature Learning

MetaGeo relies on two categories of features.

1) User Content Features: For each user v, we aggregate all the tweets to form a document and use TF-IDF [30] to measure the importance of each word in it. This, in turn, increases proportionally to the number of times it appears in the document but is offset by the frequency of the word in the corpus. TF-IDF measures the informative quantity of a word across documents—i.e., a common word would be given low weight while a rare term gets a relatively higher weight. After the word representation, the user content features matrix $\mathbf{X}$ can be immediately obtained the rows in $\mathbf{X}$ represent users and the columns are all the unique words across all documents. Note that other representation methods, such as word2vec or doc2vec can also be chosen. In this study, we follow previous work [1], [15] and use TF-IDF for fair comparisons.

2) Network Features: The network structure of user latent interactions can be captured by various network representation approaches or graph neural networks, e.g., node2vec [21] has been used for user representation [5] while GCNs [22] has been directly employed for modeling network features and predicting user geolocation [1]. Here we use simple graph convolutional (SGC) networks [31] for network feature representation, which simplifies the GCNs model by removing nonlinearity between layers and smoothing the hidden feature aggregation with linear transformations.

Specifically, we learn the network feature representation of all users with D-layer graph convolutions, where the update in each layer $\mathbf{h}^d$ ($d = \{0, 1, \ldots, D-1\}$) is a simple sparse matrix multiplication

$$\mathbf{S} = \mathbf{D}^{-\frac{1}{2}} \tilde{\mathbf{A}} \mathbf{D}^{\frac{1}{2}} \tag{1}$$

$$\mathbf{h}^{d+1} = \mathbf{S}\mathbf{h}^d \tag{2}$$

where $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$ is the adjacency matrix with self-loops added, $\mathbf{D}$ is the degree matrix of $\tilde{\mathbf{A}}$, and $\mathbf{S}$ is the normalized adjacency matrix, i.e., the normalized adjacency matrix with self-connections. The input layer is the user content features, i.e., $\mathbf{h}^0 = \mathbf{X}$. Subsequently, the final preference representation of a user obtained in the last layer is used for the downstream prediction task, which can be expressed as

$$\begin{aligned} \mathbf{h}^D &= \mathbf{S}\mathbf{h}^{D-1} = \mathbf{S}^D\mathbf{h}^0 = \mathbf{S}^D\mathbf{X} \\ &= \binom{D}{0}\mathbf{h}^0 + \binom{D}{1}\mathbf{S}\mathbf{h}^0 + \cdots + \binom{D}{D}\mathbf{S}^D\mathbf{h}^0 \\ &= \binom{D}{0}\mathbf{X} + \binom{D}{1}\mathbf{S}\mathbf{X} + \cdots + \binom{D}{D}\mathbf{S}^D\mathbf{X}. \end{aligned} \tag{3}$$

The motivation behind the basic GCNs is to average the representation of locally neighboring nodes and propagate the features through the network. To make the graph learning efficient, we remove the nonlinearity between layers and only keep the last softmax layer following [31] to produce the probability distribution for each user v

$$\hat{y} = \texttt{softmax}(f_\theta(v)) = \texttt{softmax}\big(\text{MLP}(\mathbf{S}^D\mathbf{X})\big) \tag{4}$$

where f_θ is the SGC [31] model and θ denotes the parameters.

III. MetaGeo Methodology

In this section, we describe the details of our proposed user geolocation prediction framework MetaGeo. Specifically, we explain the process of task sampling, followed by the meta-training and implementation aspects. Subsequently, we present application-oriented perspectives of MetaGeo for the purpose of evaluating its generalizability. We finish the section with a discussion of the complexity and universality aspects.

A. Task Sampling

Instead of training a single model of user geolocation prediction as previous methods, we are more interested in augmenting the learning ability of prediction models via meta-training. However, meta-learning algorithms are specifically designed to identify and learn new concepts (e.g., new classes) with few-shot samples, which usually require a set of meta-training and meta-testing tasks. In conventional user location prediction settings, we *do not* need to sample tasks for meta-testing. In this regard, one natural question is: *can we follow the traditional test settings but at the same time leverage meta-learning to improve the prediction model*?

In this work, we propose to enhance the classification model by stacking numerous mini meta-learners. Formally, we split training data $\mathcal{D}^{\text{train}}$ into support sets $\mathcal{S}$ and query sets $\mathcal{Q}$ to form a set of tasks $\mathcal{T} = \{\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_M\}$ following the typical meta-training procedure [32]. Each task $\mathcal{T}_i \in \mathcal{T}$ contains a support set $\mathcal{S}_i$ and a query set $\mathcal{Q}_i$, which are used for training and testing (in each task), respectively. Specifically, we first define i as the task index and j as the region index. Then, we perform the following steps to obtain the tasks $\mathcal{T}$ for training our model.

1) Randomly sample a subset of regions $Y_{i,j} \sim \mathcal{Y}$, $|Y_{i,j}| = N$.
2) For each region $y_j \in Y_{i,j}$, sample K users to form a support set $\mathcal{S}_i$ and P users ($\notin \mathcal{S}_i$) for a query set $\mathcal{Q}_i$.
3) Repeat steps 1) and 2) for $|\mathcal{T}| = M$ times.

The above-mentioned sampling process forms a N-way K-shot learning problem. Note that K is usually small (e.g., 3 or 5) which is consistent with traditional meta-learning settings [33]–[35]. However, N is not limited to a small number as typical meta-learning studies. In Section III-D,

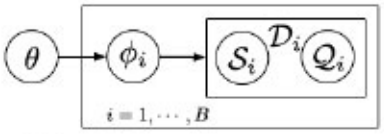


Fig. 2. Plate model for MetaGeo.

we discuss the values of N according to different applications settings.

B. Meta Training

MetaGeo trains the base model f_θ on the support set $\mathcal{S}$ and predicts the target regions of users in the query sets $\mathcal{Q}$—in the same way as the supervised learning computes the prediction loss and updates the model parameters through backpropagation. The prediction model f_θ outputs the probability that a user v is associated with the region y while incorporating the content and network-based features as in (4).

A possible way to train the base model f_θ is Model-agnostic meta-learning (MAML) [34], which explicitly learns some shared parameters θ that make the model easier to seek the right *task-specific* parameters ϕ_i when facing a novel task with limited (i.e., K) samples. However, applying MAML for training f_θ raises two challenges: 1) while MAML can directly optimize the model on the training data $\mathcal{D}^{\text{train}}$, its performance is only able to adapt to *novel* classes, rather than existing regions in training—the latter being the case of conventional geolocation setting and 2) the user locations are bounded by a region partitioned by *k-d* tree, which is uncertain and very sensitive to their parameters, especially for those from dense population areas, e.g., the east coast in the US [cf. Fig. 1(b)]. This may result in unstable task training and unsatisfactory prediction results with only a few-shot samples in each task.

To adapt our model to the conventional testing while considering the task uncertainty, we introduce a probabilistic graph model for maximum posterior inference toward estimating and alleviating the task ambiguity problem in training MetaGeo. The graphical model of MetaGeo is shown in Fig. 2, where $\mathcal{D}_i$ is independent of θ given ϕ_i (task-specific parameter), and $\mathcal{S}_i$ is independent of $\mathcal{Q}_i$. Inspired by recent advances in probabilistic meta-learning [36]–[39], we use the amortized variational inference [40] to estimate the posterior distribution of network parameters. Specifically, we consider the following optimization problem in a mini-batch of B tasks with support and query set split:

$$\arg\min \mathop{\mathbb{E}}_{\theta\sim q_\psi}\left\{-\sum_{i=1}^{B}\mathop{\mathbb{E}}_{\phi_i\sim q_\theta(\phi_i|\mathcal{S}_i)}\log p(\mathcal{Q}_i|\phi_i) + \mathrm{KL}(q_\theta(\phi_i|\mathcal{S}_i)\|p(\phi_i|\mathcal{S}_i,\theta))\right\} + \mathrm{KL}\big(q_\psi(\theta)\|p(\theta)\big). \tag{5}$$

The maximum *a posteriori* estimate of ϕ_i is obtained by a fixed number of optimizations with support data, and ψ is the variational parameter of the approximate posterior over θ. Equation(5) needs to approximate $p(\phi_i|\mathcal{S}_i,\theta)$ with proposal $q_\theta(\phi_i|\mathcal{S}_i)$, which can be obtained by updating the parameters with support set (see Appendix for derivation). We note that following [39], here, we also use the support set for amortized

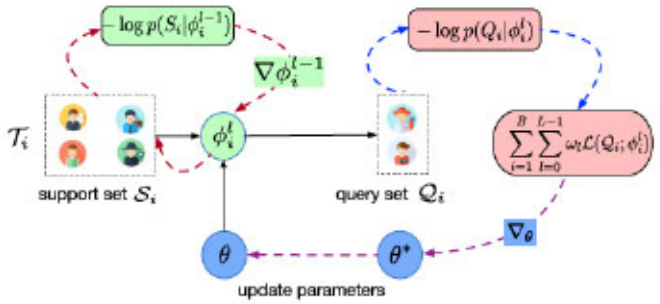


Fig. 3. Parameter updating in MetaGeo.

variational inference. However, in MetaGeo, the parameter update is different from [39]. We further clarify the details of implementation in Section III-C.

Fig. 3 shows the parameters updating procedure of MetaGeo, which is performed via a mini-batch gradient descent with random initialization of variational parameters. It consists of inner updates and outer updates, described next.

1) Inner Updates With $\mathcal{S}_i$: In the inner update with support set data $\mathcal{S}_i$, the inference is performed by running a gradient descent on the loss $\mathcal{L}(\mathcal{S}_i;\phi_i^{l-1})$ for a fixed number (L) of iterations starting from $\phi_i^0=\theta$

$$\begin{aligned}\mathcal{L}\big(\mathcal{S}_i;\phi_i^{l-1}\big) &= -\log p\big(\mathcal{S}_i|\phi_i^{l-1}\big)\\ \phi_i^l &= \phi_i^{l-1}-\alpha\nabla_{\phi_i^{l-1}}\mathcal{L}\big(\mathcal{S}_i;\phi_i^{l-1}\big)\\ &= \phi_i^{l-1}-\alpha\frac{\partial\mathcal{L}\big(\mathcal{S}_i;\phi_i^{l-1}\big)}{\partial\phi_i^{l-1}}\end{aligned} \tag{6}$$

where α is the task-based learning rate. In our work, since we treat the user geolocation prediction as a supervised classification problem, we use cross-entropy error as the loss function. In the case of supervised classification with inputs $x_j^{\mathcal{S}_i}$, corresponding labels $y_j^{\mathcal{S}_i}$ and a classifier $f_{\phi_i^{l-1}}$, we denote the negative log-likelihood of the data under the classifier as

$$\begin{aligned}\mathcal{L}\big(\mathcal{S}_i;\phi_i^{l-1}\big) &= -\log p\big(\mathcal{S}_i|\phi_i^{l-1}\big)\\ &= -\sum_{\mathbf{x}_j,\mathbf{y}_j\sim\mathcal{S}_i}\log p\Big(y_j^{\mathcal{S}_i}|x_j^{\mathcal{S}_i},\phi_i^{l-1}\Big).\end{aligned} \tag{7}$$

For discrete classification tasks with a cross entropy loss, we have

$$\begin{aligned}\mathcal{L}\big(\mathcal{S}_i;\phi_i^{l-1}\big) = \sum_{\mathbf{x}_j,\mathbf{y}_j\sim\mathcal{S}_i} & y_j^{\mathcal{S}_i}\log f_{\phi_i^{l-1}}\Big(x_j^{\mathcal{S}_i}\Big)\\ &+\Big(1-y_j^{\mathcal{S}_i}\Big)\log\Big(1-f_{\phi_i^{l-1}}\Big(x_j^{\mathcal{S}_i}\Big)\Big).\end{aligned} \tag{8}$$

2) Outer Updates With $\mathcal{Q}_i$: After each inner update, the model parameters θ in the *overall* meta-objective $\mathcal{L}(\mathcal{Q}_i;\phi_i^l)$ are updated using gradient descent

$$\theta=\theta-\beta\nabla_\theta\sum_{i=1}^{B}\sum_{l=1}^{L}\omega_l\mathcal{L}\big(\mathcal{Q}_i;\phi_i^l\big) \tag{9}$$

$$\mathcal{L}\big(\mathcal{Q}_i;\phi_i^l\big)=\arg\min -\log p\left(\mathcal{Q}_i\ |\underbrace{\phi_i^{l-1}-\alpha\nabla_{\phi_i^{l-1}}\mathcal{L}\big(\mathcal{S}_i;\phi_i^{l-1}\big)}_{\phi_i^l}\right) \tag{10}$$

Algorithm 1 MetaGeo Training

Input: training data $\mathcal{D}^{\text{train}} = (\mathcal{S}, \mathcal{Q})$; task-based learning rate α; meta-learning rate β; the number of inner loop updates L;
Output: optimal model parameters θ^*.

Initialize $\theta = \{\mu_\theta, \sigma_\theta^2\}$ at random;
$p(\theta) = \mathcal{N}(\mu_\theta; 0, I)\,\Gamma(\tau_\theta; a, b)$;
while not done **do**
  Sample a mini-batch of tasks $\mathcal{T}_i \sim \mathcal{T}$, $i = [1, \ldots, B]$;
  foreach $\mathcal{T}_i$ **do**
    $\mu_{\phi_i}^0 = \mu_\theta$; $\sigma_{\phi_i}^{2(0)} = \sigma_\theta^2$;
    /* Training on $\mathcal{S}_i$ */
    for $l = 1, \ldots, L$ **do**
      $\phi_i^l = \{\mu_{\phi_i}^l, \sigma_{\phi_i}^{2(l)}\}$;
      Compute mean and variance:
      $\mu_{\phi_i}^l = \mu_{\phi_i}^{l-1} - \alpha \nabla_{\mu_{\phi_i}^{l-1}} \mathcal{L}(\mathcal{S}_i; \phi_i^{l-1})$;
      $\sigma_{\phi_i}^{2(l)} = \sigma_{\phi_i}^{2(l-1)} - \alpha \nabla_{\sigma_{\phi_i}^{2(l-1)}} \mathcal{L}(\mathcal{S}_i; \phi_i^{l-1})$;
      Evaluate weighted query loss $\omega_l \mathcal{L}(\mathcal{Q}_i; \phi_i^l)$ on query set $\mathcal{Q}_i$;
    end
  end
  Compute the total loss: $\sum_{i=1}^{B} \sum_{l=1}^{L} \omega_l \mathcal{L}(\mathcal{Q}_i; \phi_i^l)$;
  Update θ by: $\theta = \theta - \beta \nabla_\theta \sum_{i=1}^{B} \sum_{l=1}^{L} \omega_l \mathcal{L}(\mathcal{Q}_i; \phi_i^l;)$.
end

Algorithm 2 Evaluating MetaGeo

Input: testing data $\mathcal{D}^{\text{test}} = (\mathcal{S}^{\text{test}}, \mathcal{Q}^{\text{test}})$; model parameters θ^*.

while not done **do**
  foreach $\mathcal{T}_i$ **do**
    /* Update ϕ_i with $\mathcal{S}^{\text{test}}$. */
    $\mu_{\phi_i}^0 = \mu_{\theta^*}$; $\sigma_{\phi_i}^{2(0)} = \sigma_{\theta^*}^2$;
    for $l = 1, \ldots, L$ **do**
      $\phi_i^l = \{\mu_{\phi_i}^l, \sigma_{\phi_i}^{2(l)}\}$;
      $\mu_{\phi_i}^l = \mu_{\phi_i}^{l-1} - \alpha \nabla_{\mu_{\phi_i}^{l-1}} \mathcal{L}(\mathcal{S}_i^{\text{test}}; \phi_i^{l-1})$;
      $\sigma_{\phi_i}^{2(l)} = \sigma_{\phi_i}^{2(l-1)} - \alpha \nabla_{\sigma_{\phi_i}^{2(l-1)}} \mathcal{L}(\mathcal{S}_i^{\text{test}}; \phi_i^{l-1})$;
    end
    Update model parameters: $\theta = \phi_i^L$;
    /* New region user prediction. */
    foreach *user* $v^* \in \mathcal{Q}_i^{test}$ **do**
      predict the region $\hat{y}^* = f_\theta(v^*)$;
    end
  end
end

where β is the meta-learning rate, and ω_l denotes the importance weight of the query set loss at lth step used for improving the gradient stability of training MetaGeo suggested by [41].

C. Implementation Details

Ravi and Beatson [39] use the local reparameterization trick for the fully connected layers and the Flipout technique for the convolution layers to generate (almost) completely independent weight samples. When implementing (5), we instead calculate the Kullback–Leibler (KL)-divergence terms analytically and approximate the expectations by averaging over a number of samples from the approximate posterior.

Specifically, we assume that the prior $p(\theta) = \mathcal{N}(\mu_\theta, \sigma_\theta^2)$ follows a Gaussian with mean μ_θ and variance σ_θ^2 for each parameter of the model f_θ. Then the posterior $p(\phi_i|\mathcal{S}_i, \theta)$ is also a Gaussian, i.e., $p(\phi_i|\mathcal{S}_i, \theta) = \mathcal{N}(\phi_i; \mu_\theta, \sigma_\theta^2)$. When approximating $p(\phi_i|\mathcal{S}_i, \theta)$, the proposal $q_\theta(\phi_i|\mathcal{S}_i)$ is computed with L iterations in the inner loop with initialization $\phi_i^0 = \theta = \{\mu_\theta, \sigma_\theta^2\}$. Then, for each iteration $l = 1, \ldots, L$, we can update $\phi_i^l = \{\mu_{\phi_i}^l, \sigma_{\phi_i}^{2(l)}\}$ as

$$\begin{aligned} \mu_{\phi_i}^l &= \mu_{\phi_i}^{l-1} - \alpha \nabla_{\mu_{\phi_i}^{l-1}} \mathcal{L}\left(\mathcal{S}_i; \phi_i^{l-1}\right) \\ \sigma_{\phi_i}^{2(l)} &= \sigma_{\phi_i}^{2(l-1)} - \alpha \nabla_{\sigma_{\phi_i}^{2(l-1)}} \mathcal{L}\left(\mathcal{S}_i; \phi_i^{l-1}\right). \end{aligned} \tag{11}$$

The rest of the problem is how to initialize the parameters of the prior $p(\theta)$ from which ϕ_i is sampled at the beginning of each iteration. According to Bayes rule

$$\begin{aligned} p(\phi_i|\mathcal{S}_i, \theta) &\propto p(\mathcal{S}_i, \theta|\phi_i) p(\phi_i) \\ p\left(\mu_{\phi_i}^l, \tau_{\phi_i}^l|\mathcal{S}_i, \theta\right) &\propto p\left(\mathcal{S}_i, \theta|\mu_{\phi_i}^l, \tau_{\phi_i}^l\right) p\left(\mu_{\phi_i}^l, \tau_{\phi_i}^l\right) \end{aligned} \tag{12}$$

where $\tau_{\phi_i}^l = 1/\sigma_{\phi_i}^{2(l)}$ is the precision. After L iterations with support data $\mathcal{S}_i$, the proposal $q_\theta(\phi_i|\mathcal{S}_i)$ is updated to $q_\theta(\phi_i^L)$, which can be written as

$$q_\theta\left(\phi_i^L\right) = q_\theta\left(\mu_{\phi_i}^L, \tau_{\phi_i}^L\right) = q_\theta\left(\mu_{\phi_i}^L\right) q_\theta\left(\tau_{\phi_i}^L\right) \tag{13}$$

where we assume a mean field variational approximation to true posterior $p(\phi_i|\mathcal{S}_i, \theta)$. The optimal factors $q_\theta(\mu_{\phi_i}^L)$ and $q_\theta(\tau_{\phi_i}^L)$ have analytical solutions according to general variational inference [42], which, combined with the outer updates, can be then used to initialize $p(\theta)$ at the next iteration.

As for the prior $p(\phi_i^0)$ that initialized with $p(\theta)$, we have

$$p\left(\phi_i^0\right) = p(\theta) = \mathcal{N}(\mu_\theta; 0, I)\,\Gamma(\tau_\theta; a, b) \tag{14}$$

which is derived according to the Gaussian-Gamma conjugate prior distribution. In (14), a and b are the alpha and beta parameters for the Gamma distribution.

Gaussian Variational Posterior: In previous endeavors [40], [43], KL-divergence is obtained by analysis and calculation, while the expectation values are approximated by taking the average of multiple samples from the approximate posterior. Here, we describe the Gaussian variational posterior, shifting parameters by a mean μ and scaling by a standard deviation σ. We parametrize the standard deviation pointwise as $\sigma = \log(1 + \exp(\rho))$. Then, for function f_θ

$$\begin{aligned} f_\theta(\mu, \sigma) = \log q((\mu + \sigma \circ \epsilon)|\theta) \\ - \log p(\mu + \sigma \circ \epsilon) p(\mathcal{T}|(\mu + \sigma \circ \epsilon)) \end{aligned} \tag{15}$$

where $\circ$ is pointwise multiplication, $\epsilon \sim \mathcal{N}(0, I)$. Then, for each optimization, we calculate the gradient with respect to the mean

$$\Delta_\mu = \frac{\partial f_\theta(\mu, \sigma)}{\partial(\mu + \sigma \circ \epsilon)} + \frac{\partial f_\theta(\mu, \sigma)}{\partial \mu}. \tag{16}$$

For the standard deviation parameter we calculate the gradient as follows:

$$\Delta_\rho = \frac{\partial f_\theta(\mu,\sigma)}{\partial(\mu+\sigma\circ\epsilon)}\frac{\epsilon}{1+\exp(-\rho)} + \frac{\partial f_\theta(\mu,\sigma)}{\partial\rho}. \quad (17)$$

To learn both the mean and the standard deviation, we simply calculate the usual gradients found by backpropagation and then scale and shift them as above. In a few-shot learning setting, we can easily generate multiple weight samples by replicating data in each task. In the case of defining Gaussian distribution, we use the learned prior knowledge and Adam [44] to optimize the model.

D. Application Details in MetaGeo

The pseudocodes for training and evaluating MetaGeo are, respectively, outlined in Algorithms 1 and 2. We evaluate the generalizability of the model in two scenarios. In Section III-D.1, we compare our model with conventional user geolocation methods. In Section III-D.2, we evaluate MetaGeo's performance on identifying the user from unseen regions. The specific experimental setups are described in Sections IV-C and IV-D, respectively.

1) Application-1 (User Geolocation Prediction): In the case of conventional user geolocation prediction, i.e., the regions that the testing users belong to have been seen during training, f_θ is a $|\mathcal{Y}|$-classifier model, where $\mathcal{Y}$ is the label set for location prediction. Therefore, we classify each testing user $v^\star$ into the most likely region by the trained model f_θ through the softmax function: $\text{softmax}(f_\theta(v^\star))$, which outputs the probability $p(\hat{y}^\star)$ of the region that $v^\star$ is classified to.

Note that here, we modify the meta-learner to be a $|\mathcal{Y}|$-classifier, rather than a N-classifier in typical few-shot learning ($|\mathcal{Y}| \gg N$).

2) Application-2 (Generalization to New Regions): When predicting the user regions that are *unknown* during meta-training, MetaGeo turns to address a typical meta-learning problem. In other words, it becomes an N-classifier model, where N is the number of regions (labels) in each task, rather than $\mathcal{Y}$ in Application-1. Namely, we classify each user $v^\star$ into one of the N classes using the softmax function.

Note that in this application the model would still update the parameters (inner updates) with support data in $\mathcal{D}^{\text{test}}$ when predicting users from new regions (cf. Algorithm 2). We also note that in the implementation, MetaGeo does not directly parameterize the variance parameters but the standard deviation instead. This trick is suggested by [39] and [43], which allows us to parameterize the standard deviation and guarantees that σ is always nonnegative.

E. Discussion

1) Computational Complexity: By operating in the weight space, MetaGeo substantially reduces sample complexity. For clarity, we compare our method with the GCN4Geo model. Similar to [38] and [45], we consider splitting the model architecture into the features extractor layers and the classifier. For the GCN4Geo model, the network features extractor is a GCN with D layers and the classifier is an MLP with softmax output. In MetaGeo, we share the feature extractor across all the users, whereas each user has its own classifier. Therefore, the computational complexity of GCN4Geo is $\mathcal{O}(|\theta_{\text{GCN}}| + |\theta_{\text{MLP}}|)$. MetaGeo's features extractor also combines the text- and network features by D-layer linear feature aggregation—however, we remove the nonlinearities between layers in graph learning, which could reduce the computation complexity to $O(1)$. Note that MetaGeo's classifier is a single-layer fully connected network with softmax output. Therefore, the computational complexity of the MetaGeo network is $\mathcal{O}(|\theta_{\text{MLP}}|)$.

For the parameter updating procedure of MetaGeo in a mini-batch (B) task, we get the initial network parameters with Gaussian distribution by variational inference. During inference, the complexity of MetaGeo can be divided into two parts. The first part is the support set (S has $N \times K$ samples) complexity caused by the inner updates. The second part is query set (Q has $N \times P$ samples) complexity due to the outer updates. The space complexity of MetaGeo is $\mathcal{O}(BN(K|\theta_{\text{MLP}}|+P|\phi_{\text{MLP}}|))$, where ϕ_{MLP} has the same dimension as θ_{MLP} to represent the parameters updated by gradient descent. In contrast, the space complexity of the same number of samples in GCN4Geo is $\mathcal{O}(BN(K+P)(|\theta_{\text{GCN}}|+|\theta_{\text{MLP}}|))$.

2) Universality of MetaGeo: We now address the last question: whether MetaGeo is a generic framework for identifying user geolocation. More specifically, is it applicable both for traditional user geolocation and identifying users in unseen regions studied in this article? To answer this question, we investigate the way of parameter learning in MetaGeo. Suppose we predict the location $\hat{y}^\star$ for a test user $v^\star$: $\hat{y}^\star = f_\theta(v^\star)$, where parameters θ are learned from MetaGeo. Since there are K users during training in each task, parameters are updated by

$$\phi = \theta - \alpha\nabla_\theta \frac{1}{K}\sum_{j=1}^{K}\mathcal{L}\big(y_j, f_\theta(v_j)\big) \quad (18)$$

where y_j is the ground truth for v_j, and $\mathcal{L}$ can be any loss function (e.g., cross entropy). For a neural network f_θ that can approximate the underlying function, its postupdate function is

$$f_\phi(v^\star) \approx \text{h}\left(-\alpha\frac{1}{K}\sum_{j=1}^{K}\sum_{i=1}^{N}\text{vec}_i\big((v_j, y_j), v_i^\star\big); \theta_{\text{h}}\right) \quad (19)$$

where $\text{h}(\cdot;\theta_{\text{h}})$ is a neural network with hidden layers, and vector vec is approximated by $\text{vec}((v_j, y_j)_{j=1}^K, v^\star) \approx [0,\ldots,\mathbf{y}_j,\ldots,0]^\intercal$ for discrete labels $\mathbf{y}_j$. The summation over vec amounts to and completely describes the frequency counts of the triplets $((v_j, y_j)_{j=1}^K, v^\star)$ and can also decode the labels $\mathbf{y}_j$. Since neural network h is a universal function approximator and its representation is redundant in $v^\star$, it contains sufficient information to decode the test input $v^\star$ and set of users $(v_j, y_j)_{j=1}^K$. That is, function f_ϕ is a universal function approximator w.r.t. $((v_j, y_j)_{j=1}^K, v^\star)$—thus, it does not depend on the specific forms of loss function. It has been theoretically proven in [46] that cross entropy loss and mean-squared error allow for the universality of gradient-based meta-learning.

IV. Evaluation

We now present the details of our experimental evaluation of MetaGeo. Specifically, we first describe the datasets and metrics. Subsequently, we follow with

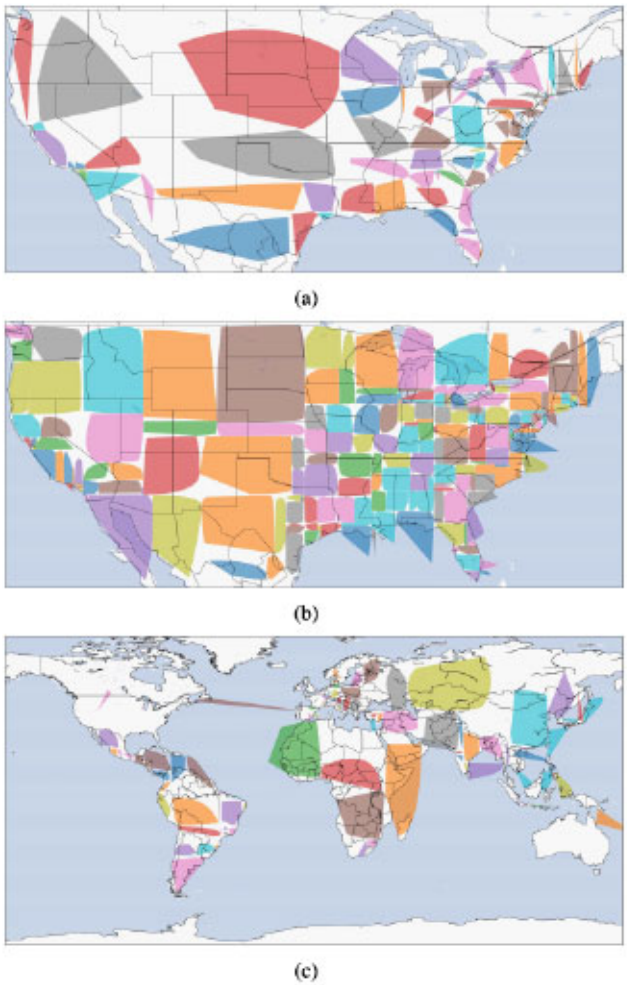


Fig. 4. Regions obtained via *k-d* tree. (a) GeoText. (b) Twitter-US. (c) Twitter-World.

Sections III-D.1 and III-D.2, and in each of them, we present the state-of-the-art baselines we compared and the outcomes of the evaluation.

A. Datasets

We evaluate the methods using three benchmark Twitter datasets, i.e., GeoText [8], Twitter-US [29], and Twitter-World [3], which have been widely used for evaluating user geolocation prediction models. GeoText and Twitter-US consist of users across the main continental of the U.S., while Twitter-World covers a set of worldwide users. The label (region) of each user is determined by the geographical coordinates of the training data using *k-d* tree [3] which results in 129, 256, and 930 regions for GeoText, Twitter-US, and Twitter-World, respectively. We note that a visualization of the *k-d* tree-based regions for the datasets is presented in Fig. 4, which provides an illustration of the distribution of regions obtained via the *k-d* tree, for three datasets used in this work. We evaluate the generalizability of the model in two scenarios. In Section IV-C, we compare our model with traditional user geolocation methods. In Section IV-D, we replace our meta-learning method with several state-of-the-art approaches and evaluate their performance in identifying the user from unseen regions. The specific experimental setups are described in Sections IV-C and IV-D, respectively. Table I shows the statistics of the datasets.

TABLE I

DATA DESCRIPTION. $|\mathcal{V}|$: THE NUMBER OF USERS; m: THE NUMBER OF REGIONS (LABELS); $\mathcal{E}$: THE NUMBER OF EDGES; N: THE NUMBER OF WAYS (REGIONS) IN EACH TASK; K/P: THE NUMBER OF USERS IN SUPPORT SET AND QUERY SET FOR EACH TASK; $|\mathcal{T}|$: THE NUMBER OF SAMPLED TASKS

	GeoText	Twitter-US	Twitter-World
$\lvert\mathcal{V}\rvert$	9,475	449,200	1,386,766
$\lvert\mathcal{V}\rvert_{\text{train}}$	5,685	429,200	1,366,766
$\lvert\mathcal{V}\rvert_{\text{val}}$	1,895	10,000	10,000
$\lvert\mathcal{V}\rvert_{\text{test}}$	1,895	10,000	10,000
$\mathcal{Y}$	129	256	930
$\mathcal{E}$	77,155	5,297,215	1,001,181
N	10	40	40
K/P	5/30	5/75	5/15
$\lvert\mathcal{T}\rvert$	12,800	16,000	25,600

B. Metrics

We measure the distance between the predicted and the true geolocation and evaluate the models with three commonly used metrics in geolocation prediction, i.e., *ACC@161*, *Median*, and *Mean* value of error distances. *ACC@161* is an accuracy metric that considers predictions having errors within 161 km (or 100 miles) as correct predictions, while *Median* and *Mean* denote the median and mean value of error distances in predictions, respectively. As a classification metric, the higher value of *ACC@161* indicates a good prediction. On the contrary, lower values of *Median* and *Mean* distance errors indicate better performance.

C. Application-1: User Geolocation Prediction

In the first group of experiments, we compare the overall performance of methods on user geolocation prediction.

1) Baselines: We compare MetaGeo with several state-of-the-art approaches, including in the following.

1) *HierLR* [4] is a text-based geolocation model, which adopts a grid representation of locations and resorts to hierarchical classification using logistic regression (LR).
2) *MADCEL* [15] combines the text and network information and uses LR for location prediction.
3) *MLP4Geo* [12] is a text-based model which uses dialectal terms to improve the prediction performance. A simple MLP network is used to predict locations.
4) *MENET* [5] is a MENEXT architecture unifying various features of tweets. To have a fair comparison, we only use text and network information in MENET, i.e., we do not use the metadata, such as timestamps, that have been exploited in [5].
5) *GeoAtt* [6] models the textual context with an attention-based RNN. We remove the location descriptions in GeoAtt for a fair comparison.
6) *DCCA* [1] is a multiview geolocation model using twitter text and network information and measures the canonical correlation for location prediction.
7) *BiLSTM-C* [47] is a text-view geolocation model which treats user-generated content and their associated

TABLE II
PERFORMANCE COMPARISON OF METHODS ON THREE DATASETS

	Acc@161	Median	Mean	Acc@161	Median	Mean	Acc@161	Median	Mean
	GeoText			Twitter-US			Twitter-World		
HierLR	42%	426	856	48%	191	686	31%	509	1,669
MADCEL	58%	60	586	54%	116	705	45%	279	2,525
MLP4Geo	38%	389	844	54%	120	554	34%	415	1,456
MENET	55%	125	643	56%	93	526	52%	126	1,290
GeoAtt	57%	81	612	54%	91	545	50%	214	1,263
DCCA	56%	79	627	58%	90	516	21%	913	2,095
BiLSTM-C	45%	363	796	49%	137	689	41%	423	1,543
Attn	52%	236	657	52%	105	602	45%	286	2,456
GCN4Geo	60%	45	546	62%	71	485	54%	108	1,130
SGC4Geo	61%	45	543	62%	71	487	54%	108	1,133
MetaGeo	**62%**	**42**	**533**	**63%**	**70**	**479**	**55%**	**105**	**1,118**

locations as sequences and employs bidirectional long short-term memory (LSTM) and convolution operations to infer locations.

8) *Attn* [47] is an attentive memory network for the localization of social media messages. It consists of an attentive message encoder, which selectively focuses on location-indicative terms to derive a discriminative message representation.
9) *GCN4Geo* [1] is a GCNs-based model that utilizes both text and network context for geolocation prediction, where layerwise gates are employed for controlling the neighborhood smoothing to alleviate the noisy propagation in GCNs.
10) *SGC4Geo* is a simplified graph convolutional network [31] that reduces the excess complexity of GCNs by repeatedly removing the nonlinearities between GCN layers and collapsing the resulting function into a single-linear transformation. It locates the home position of users given users' posts and social connections.

2) Experimental Settings: Following previous works [1], [6], [10], [15], the datasets are partitioned into the train, validation, and test sets. The parameters in all baselines follow the settings in the original papers. For MetaGeo, we sample the meta-training data from the original training set using the task sampling algorithm described in Section III-A. The validation and test sets are the same as the conventional settings of previous geolocation approaches [1]. The values of parameters N, K, P and $|\mathcal{T}|$ in meta-training phase are as shown in Table I, unless otherwise specified. In addition, the batch size B in all deep learning methods is set to 32, 64, and 128 for GeoText, Twitter-US, and Twitter-World, respectively. Finally, we use 4, 3, and 3 layer linear feature aggregation for interaction network structure learning.

3) Performance on User Geolocation Prediction: Table II shows the overall performance of all methods across three datasets, from which we have the following observations.

1) MetaGeo consistently outperforms the baselines on all metrics, which proves the effectiveness of tackling the user geolocation problem with the proposed meta-learning framework. This improvement lies in training MetaGeo in an ensemble learning manner even with few-shot samples within each task, which improves the classification accuracy via learning the cross-task generality. This result also suggests another important application of few-shot learning except identifying new concepts that meta-learning and transfer learning did. That is, we can stack a number of minor and simple neural networks to improve the classification performance in a conventional testing way.
2) The performance of deep learning-based models, including MENET, GeoAtt, DCCA, and GCN4Geo, are very similar if both text and network features are used. Surprisingly, they are also very close to the models using simple classification methods, e.g., the LR in MADCEL. This result implicates that meaningful features are more important than complicated models in the user geolocation prediction task. This observation can be further proven in previous work [5], [6] that incorporates more strong indicators, such as timezone of users and descriptions in the location field—however, improving MetaGeo with more features is beyond the scope of this work and is left for future work.
3) Relying only on tweet content [4], [12] is not enough for accurate geolocation prediction. In other words, user interaction network plays a crucial role in predicting locations. For example, GCN4Geo directly leverages GCNs for modeling user interactions and usually performs best among baselines. However, one drawback of GCN4Geo is the computational complexity due to the considerably nonlinear transformations in GCNs. In contrast, our MetaGeo removes the nonlinearities between GCNs layers and therefore requires significantly less overhead in extracting the network structure. Though stacking numerous tasks incurs extra overhead, it is trivial compared to learning textual and network features, due to the very few samples within each task.

4) Sensitivity Analysis: There exist many parameters involved in MetaGeo that might affect its performance. In this section, we conduct a sensitivity analysis to understand the impact of these parameters on model effectiveness. We use the dataset GeoText as illustrative example. Similar phenomena are also observed on the other two datasets.

a) Effect of $|\mathcal{T}|$: Fig. 5(a) shows the influence of the number of tasks on the performance of MetaGeo, where we vary the value of $|\mathcal{T}|$ from 1600 to 25 600. We can see that the best performance is achieved when there are around 13 000 tasks. Intuitively, we need to sample enough tasks to stabilize the model, but we also observe overfitting of

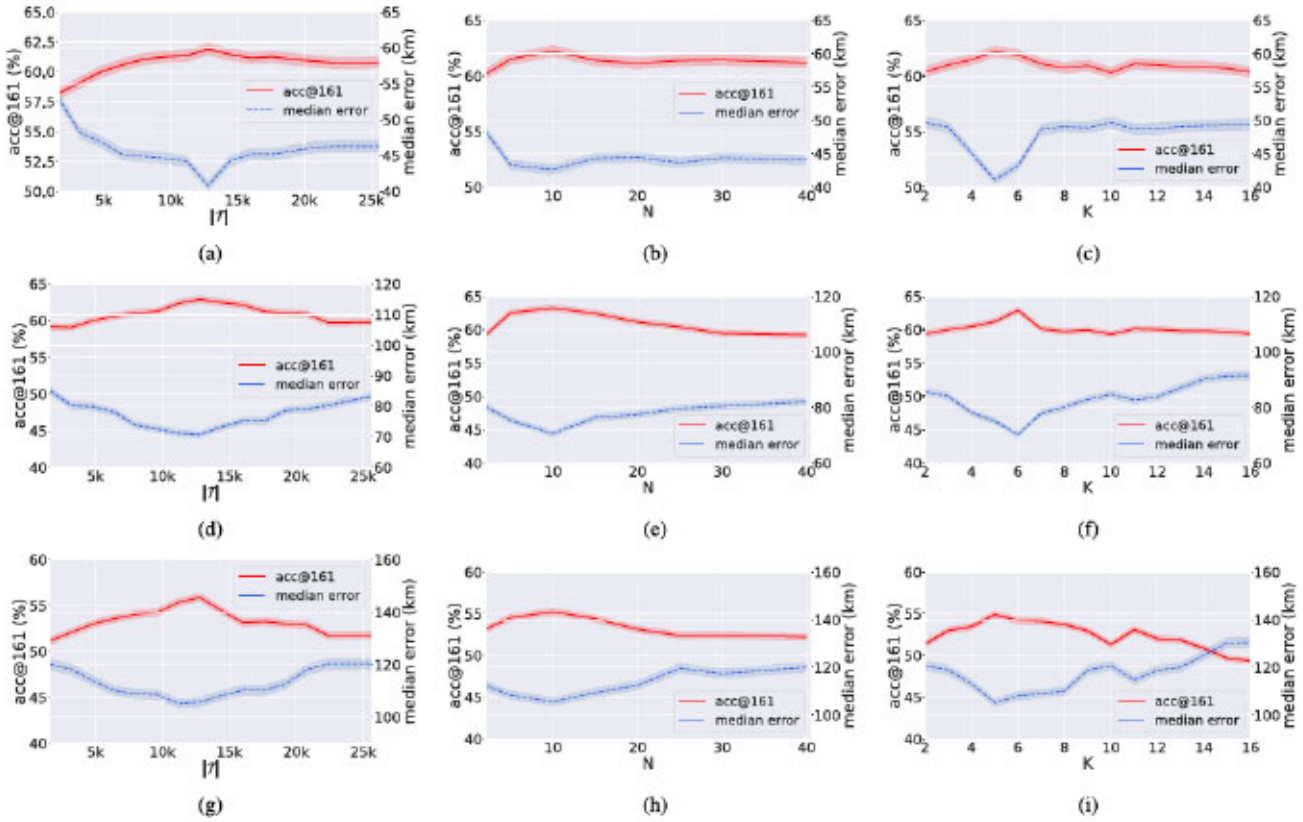


Fig. 5. Impact of parameters on performance. GeoText: (a) effect of $|\mathcal{T}|$; (b) effect of N; and (c) effect of K. Twitter-US: (d) effect of $|\mathcal{T}|$; (e) effect of N; and (f) effect of K. Twitter-World: (g) effect of $|\mathcal{T}|$; (h) effect of N; and (i) effect of K.

the model when the tasks are over-sampled, e.g., beyond 15 000 tasks. The main explanation is that many tasks might be similar or even the same when $|\mathcal{T}|$ is too large, which may skew the model performance via focusing more on these tasks. A straightforward method of alleviating this is to remove repeated tasks, which is effective based on our study (details are omitted since this is not the focus of this article).

b) Effect of N: N determines the number of classes (target regions) in each task. We investigate its impact by varying the value from 2 to 40. Intuitively, the performance of MetaGeo should be improved with the value of N increased. However, we empirically observe that a small number of N (e.g., 10) is enough to achieve good performance [see Fig. 5(b)]. This is reasonable since the tasks are sufficiently sampled, few classes are enough to train the MetaGeo, i.e., the performance of MetaGeo is more related to the number of tasks. In addition, fewer classes in each region require less computation overhead which is more desirable from the efficiency perspective.

c) Effect of K: Fig. 5(c) illustrates the impact of K, where we can observe that the performance of MetaGeo becomes stable if K is larger than 5. This is intuitive: while increasing the samples in tasks is a straightforward method to improve the learning ability of base models (e.g., when $K < 5$), this effect would be neutralized if there are sufficient tasks.

D. Application-2: Generalization to New Regions

We now turn to the results of identifying locations of users in new regions by comparing MetaGeo to few-shot learning baselines. This is one of our main objectives as we expect the system can be generalized to new areas that are *unseen* in the training set.

1) Baselines: To further demonstrate the superiority of MetaGeo, we add several meta-learning-based benchmark models:

1) *Matching Network* [48] learns a classifier defining a probability distribution over output labels given a test sample, and uses a kernel to weight the samples in the support set.
2) *Prototypical Network* [49] is a metric learning approach that encodes each input data into a low-dimensional vector. Then the prototype feature vector—i.e., the mean vector of the embedded support samples in each class—is used to predict the labels using k-NN.
3) *Meta-learner* [50] implements an LSTM as the meta-learner to update parameters using a small support set so that the learner can adapt to new tasks quickly.
4) *MAML* [34] is an optimization-based method, which learns an initialization for a base model such that after a few gradient updates w.r.t. samples in support set, the base-model can achieve strong generalization performance on new classes given only a few samples.
5) *BMAML* [38] is a Bayesian MAML model that combines gradient-based meta-learning with nonparametric variational inference. It applies approximate Bayesian inference to task-specific parameters.
6) *ABML* [39] extends the work of [36] that considered hierarchical variational inference for meta-learning by

learning the posteriors over both meta- and task-specific parameters with variational inference.

2) Experimental Settings: We randomly select $|\mathcal{C}^{\text{train}}|$ regions and the corresponding users for meta-training ($\mathcal{D}^{\text{train}}$), and $|\mathcal{C}^{\text{test}}|$ regions for meta-testing ($\mathcal{D}^{\text{test}}$). Note that the regions in $\mathcal{D}^{\text{test}}$ are unseen during training, i.e., $\mathcal{C}^{\text{train}} \cap \mathcal{C}^{\text{test}} = \varnothing$. We repeat this process ten times for each dataset and report the average performance. The meta-learning parameters of all methods, such as K, P and $|\mathcal{T}|$ in meta-training and meta-testing phase are shown in Table I, except that N is set to 5. The batch size B in all the methods is set to 32, 64, and 128 for `GeoText`, `Twitter-US` and `Twitter-World`, respectively.

In both Matching Network and Prototypical Network, we use a CNN containing four layers as a backbone network, each of which comprised 64-filters of 3×3 convolution, a Relu nonlinearity function, and max pooling. A fully connected layer followed by a softmax nonlinearity is used to define the baseline classifier. Here we add dropout to each convolutional block in both matching network and prototypical network to prevent overfitting. For Meta-learner, we use a 2-layer LSTM as a meta-learner, where the first layer is a normal LSTM and the second layer is the modified LSTM meta-learner. The gradients and losses are preprocessed and fed into the first layer LSTM, and the regular gradient coordinates are also used by the second layer LSTM to implement the state update rule. In meta-learner, we use Adam to train LSTM. For the hyperparameters of the Adam optimizer, we set the learning rate to 0.001 and use gradient clipping with a value of 0.25. Finally, MAML uses the same backbone networks as MetaGeo, however, it does not need to estimate the task uncertainty.

3) Performance Comparison on New Region Users: Table III reports the results for the settings of new region user geolocation prediction. As shown, our model achieves around 71%, 66%, and 57% Acc@161 on three datasets, which are significantly better than GCN4Geo—a semisupervised graph neural networks model. Note that we omit other user geolocation baselines because they are supervised methods and less competitive in recognizing users from new regions. Here, the models are performing five-class classification in each meta-testing task, which means that GCN4Geo only performs slightly better than a random guess. Although GCNs has been effective in user geolocation prediction with a few labeled data due to its ability of in-network label propagation, it is inapplicable to the scenario where users are from the regions that have not been seen during training. In contrast, our MetaGeo can accurately locate users from new regions with limited exposure to the few-shot samples in the new task.

Among few-shot learning baselines, matching network and meta-learner do not show comparable performance. These metric learning-based methods use deep neural networks to map the input space into the embedding space. The main idea is to classify samples based on the learned distance function, i.e., users belonging to the same regions should be close in the embedding space. However, this assumption may not be true in the user geolocation task since we only have their textual features and interaction networks. In other words, users that are geographically close (or in the same region) might have different posting behavior and social interactions. We note that the prototypical network exhibits the best performance among baselines, though it still follows the line of metric learning. The reason is that the prototypical network learns a prototype for each class which could, to some extent, compensate for the discrepancy of user representation in each region.

TABLE III
PERFORMANCE COMPARISON FOR NEW REGION USER GEOLOCATION PREDICTION. THE REPORTED NUMBERS ARE THE AVERAGE RESULTS ON TEN SAMPLED DATASETS

	Acc@161	Median	Mean
`GeoText`($\|\mathcal{C}^{\text{train}}\|/\|\mathcal{C}^{\text{test}}\|$ = 95/34)			
GCN4Geo	30±10%	859±264	1,041±219
Matching Net	47±7%	226±84	697±151
Meta-learner	52±7%	187±44	647±134
MAML	59±9%	68±24	467±121
Prototypical Net	61±5%	54±17	421±107
BMAML	67±5 %	49±18	406±110
ABML	63±4 %	55±19	422±101
MetaGeo	**71±3%**	**42±13**	**364±96**
`Twitter-US`($\|\mathcal{C}^{\text{train}}\|/\|\mathcal{C}^{\text{test}}\|$ = 189/67)			
GCN4Geo	22±2%	1,255±114	1,379±44
Matching Net	49±7%	174±45	679±164
Meta-learner	55±9%	94±34	523±110
MAML	58±11%	82±26	519±147
Prototypical Net	60±6%	76±21	451±126
BMAML	64±4 %	69±19	447±104
ABML	62±3 %	73±23	449±109
MetaGeo	**66±4%**	**64±17**	**442±93**
`Twitter-World`($\|\mathcal{C}^{\text{train}}\|/\|\mathcal{C}^{\text{test}}\|$ = 687/243)			
GCN4Geo	19±3%	1,453±201	2,021±121
Matching Net	37±5%	354±102	1,321±117
Meta-learner	46±4%	291±95	1,538±173
MAML	51±6%	203±22	1,278±153
Prototypical Net	53±5%	114±23	1,120±115
BMAML	55±5 %	93±24	1,099±110
ABML	54±5 %	107±21	1,103±116
MetaGeo	**57±3%**	**89±15**	**1,043±94**

Our model outperforms MAML because we consider the task uncertainty and stabilize the inner parameter updates with stochastic inference, besides learning the parameter initialization. Compared with BMAML and ABML which also leverage Bayesian inference for meta-learning, MetaGeo generates more accurate geolocation results. This benefit comes from the different parameter update strategies in our model. For example, ABML minimizes the loss of the support and query sets of a task jointly. It is equivalent to maximize $\mathbb{E}[\log p(\mathcal{S}, \mathcal{Q})]$, which does not explicitly encourage the model to maximize the posterior $\mathbb{E}[\log p(\mathcal{Q}|\mathcal{S})]$ that optimized in our method—the latter can well generate the model performance on the support data to the query set for the present work.

4) Error Analysis: We conducted another experiment to understand the prediction error of MetaGeo. Specifically, we first sample the test data $\mathcal{C}^{\text{test}}$ from the dense population areas (e.g., the east coast in the U.S., [cf. Fig. 1(b)] while keeping other experimental settings unchanged. The results are shown in Fig. 6, where we can see that MetaGeo achieves very high accuracy if users are from the dense population areas, even if the regions are unknown during training. On the contrary, Fig. 7 shows the performance of models on geolocating users from remote areas, where we can see that all

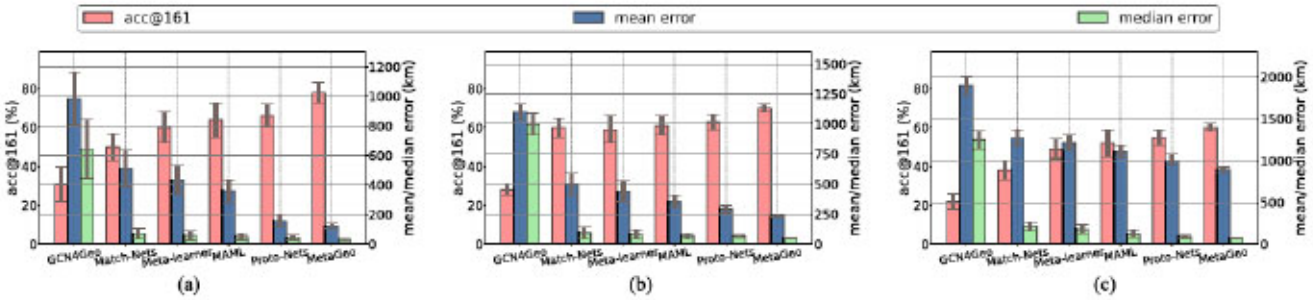


Fig. 6. Performance on dense population areas. $R = |\mathcal{C}^{\text{train}}|/|\mathcal{C}^{\text{test}}|$. (a) GeoText ($R = 101/28$). (b) Twitter-US ($R = 199/57$). (c) Twitter-World ($R = 716/214$).

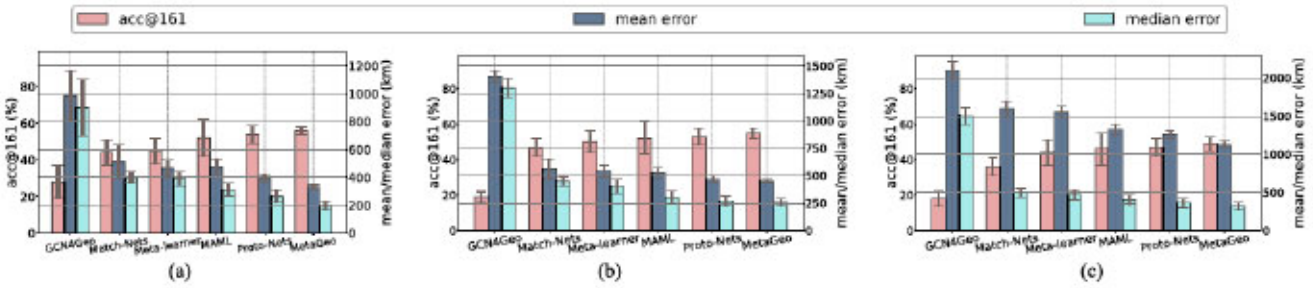


Fig. 7. Performance on remote population areas. $R = |\mathcal{C}^{\text{train}}|/|\mathcal{C}^{\text{test}}|$. (a) GeoText ($R = 114/15$). (b) Twitter-US ($R = 225/31$). (c) Twitter-World ($R = 818/112$).

methods decline significantly, although our MetaGeo performs relatively well due to its ability to handle task uncertainty. The reason behind this phenomenon is obvious, i.e., there are a few users in these remote areas who are far from each other, which makes it difficult for the model to identify the users, especially with few-shot samples. These results indicate that our model can be more effective when locating users from geographically close areas, and also hint at an open problem for future work, i.e., how to generalize the knowledge from knowledge-intensive areas to locate users from remote regions.

V. Related Work

We now present an overview of the related literature and position our work in that context.

A. Twitter User Geolocation

Previous works can be broadly categorized as content-based methods, user interaction network-based approaches, and multiview fusion models. Earlier efforts [8]–[10], [29], [58]–[60] focused on leveraging user-generated content for geolocation prediction, motivated by the fact that user location can be casually revealed by certain location-indicative words in his/her posting content. These methods study the location-related terms with the probabilistic model by characterizing the conditional distribution of user location given their published content [2]. This, however, requires extensive manually labeled location-related words to achieve satisfactory results.

Another important factor is the online social relation with the homophily assumption [13]–[15], [61], [62], i.e., people prefer to interact with others who have higher geolocation similarity. Rather than solely relying on friendship, researchers found that various types of connections, such as user co-mention tags, interactions between nonfriends, and social influence, are also strong indicators for location prediction [12]. Due to the involvement of location dependency between socially connected users, some issues still need to be carefully investigated, e.g., the location of connected users of a focal user is often unknown, or their locations might be contradicting with each other.

Recent efforts [1], [6], [12], [47] have developed many deep learning-based methods to tackle the user location prediction problem. For example, bag-of-words representation of user-generated content and fully connected networks have been implemented for predicting geolocations in [12], while RNN and attention mechanisms have been used for text content modeling and indicative location capturing [47]. Recently, [1] employed GCNs for jointly learning user-generated content and network structures. However, these methods fail to consider the scenario with few samples and cannot be applied to infer users belonging to new regions—i.e., regions which have not been encountered during training.

A detailed comparison of existing user geolocation methods is summarized in Table IV. We note that the data management community has also studied the co-located community detection (CCD) problem in the context of social networks [63], [64]. However, CCD deals with "snapshot" data (i.e., instantaneous data) and does not predict users' geolocations.

B. Few-Shot and Meta-Learning

Few-shot learning [65] plays a key role in stimulating human intelligence due to its ability to bridge the gap between machine learning and human recognition, i.e., humans can quickly acquire new concepts with little supervision information, whereas machine learning (especially deep learning) models usually require a certain amount of labeled data. Meta-learning or *learning-to-learn* [32] aims at addressing the new

TABLE IV
SUMMARY OF STUDIES ON USER GEOLOCATION PREDICTION (SORTED BY PUBLICATION YEARS)

Work	DataSource	Ground Truth	Granularity	Model	Features	Few-Shot
ACL'10 [8]	Twitter	The earliest geo-tagged city	State	Geographical Variation with Cascading Topic	Content	×
CIKM'10 [51]	Twitter	Most frequent geo-tagged city	City	Local Filtering with Neighborhood Smoothing	Content	×
WWW'10 [13]	Facebook	Coordinates	Coordinates	Maximum Likelihood	Geography Social Friendship	×
ACL'12 [29]	Twitter Wiki	The earliest geo-tagged coordinates	Grid	Document Similarity	Content	×
COLING'12 [3]	Twitter	Most frequent geo-tagged city	City	Finding Location Indicative Words	Content	×
VLDB'12 [52]	Twitter	Registered locations	City	Multiple Location Profiling	Content, Network	×
AAAI'13 [53]	Twitter Foursquare	Location profile, Median geo-tagged coordinates	Coordinates	Spatial Label Propagation	Network	×
CIKM'13 [54]	Twitter	Median geo-tagged coordinates	Coordinates	Decision Tree, Maximum Likelihood Estimator	Network	×
VLDB'14 [14]	Twitter Gowalla	Most frequent check-in, location profile	Coordinates	Social friendship-based	Network	×
ACL'15 [15]	Twitter	The earliest geo-tagged coordinates, coordinates of the most frequent geo-tagged city	Coordinates	Logistic Regression	Content, Network	×
ACL'17 [12]	Twitter	The earliest geo-tagged coordinates, coordinates of the most frequent geo-tagged city	Grid	Multilayer Perceptron	Content	×
ACL'17 [6]	Twitter	The earliest geo-tagged coordinates, majority vote of the closest city center	City	Neural Networks, Attention Mechanism	Content, Network, Context,Timestamp	×
CoRR'17 [5]	Twitter	The earliest geo-tagged coordinates	Grid	Neural Networks	Content, Network Timestamp	×
EMNLP'17 [55]	Twitter	The earliest geo-tagged coordinates	Coordinates	Mixture Density Networks	Context	×
ACL'18 [1]	Twitter	The earliest geo-tagged coordinates	Grid	Graph Convolutional Networks	Content, Network	×
EMNLP'18 [56]	Twitter Yago3	The earliest geo-tagged coordinates	Coordinates	Knowledge-Based Model	Content	×
TKDE'18 [47]	Twitter	Single city	City	Bayes-based Model	Content, Timestamp	×
TKDD'18 [7]	Twitter	The earliest geo-tagged coordinates	Grid	Gaussian Mixture Based Model	Content, Network	×
SDM'19 [57]	Twitter	Coordinate of POI center	Coordinates	Attention Memory Model	Content, Points-of-Interest	×
MetaGeo (Ours)	Twitter	The earliest geo-tagged coordinates	Coordinate	Meta-learning, Amortized Bayesian Learning	Content, Network	√

task adaption problem by learning the prior knowledge to tackle a *new* few-shot classification task. This has been widely used for benefiting various application domains, including computer vision [33], [35], [66], natural language processing [67], [68], healthcare [69], graph data learning [70], [71], concept drift adaptation [72], and spatiotemporal data mining [73]. In particular, meta-learning has shown dominant performance in image classification, where low-level patterns and features are transferable across tasks [34], [50], [74]. In this spirit, MetaGeo is among the first works that learn to utilize distributional user-generated content and interaction network structure in the context of cross-region knowledge transfer.

Existing meta-learning methods can be broadly classified into three groups: metric learning [48], [49], [75]–[77], model-based methods [33], [35] and optimization-based methods [34], [50]. The main idea of metric learning is to learn the representation of samples (in the support set) and predict the labels of samples (in the query set) using the designed distance kernel functions. For example, Matching Networks [48] learn a classifier by defining a probability distribution over output labels given a test sample, where attention kernel is used for weighting the samples in the support set. Prototypical Networks encode each input data into a low-dimensional vector and the prototype feature vector—i.e., the mean vector of the embedded support samples in each class—is used to predict the labels, which is similar to the k-NN method. Similarly, Relation Networks [77] predict the relation score between any pair of samples using a CNN classifier. Metric learning-based approaches strongly rely on the i.i.d. assumption of the data/task distributions, which, unfortunately, does not always hold in realistic scenarios.

Model-based meta-learning methods use specifically designed neural networks (e.g., RNNs) to update model's states (e.g., the internal state of RNNs), which are then utilized to make predictions. MANN [33] uses external memory storage to cache the knowledge from previous tasks in order to facilitate the learning process of new tasks. The attention mechanism is utilized for important information retrieval, which decides how to assign attention weights to information in the memory. MetaNet [35] consists of a base learner and a meta learner—the former performs in the input task space whereas the latter operates in a task-agnostic metaspace. The meta-learner learns meta-level knowledge across tasks and rapidly parameterizes both itself and the base learner to recognize new concepts of new tasks. Despite their promising performance on new task adaptation, model-based methods require specific neural architecture and/or external memory. In addition, the optimal strategy of designing a meta-learner for *arbitrary* tasks may not always be obvious [78].

Optimization-based approaches frame meta-learning as a bilevel optimization procedure, where the *inner* steps try to adapt a given task, and the *outer* meta-learner generalizes the adaptation ability of models. The goal of the meta-learner is therefore to find a single set of model parameters with a few steps of gradient descent using the support set. MAML [34] is one of the typical optimization-based methods that are independent of the underlying specific models and has achieved superior performance compared with the other popular meta-learning approaches. It also inspired numerous extensions in the recent years [36]–[39], [41], [79]. Significantly, instead of directly applying MAML, we modify the training process and present a few-shot learning paradigm to adapt MetaGeo to the typical geolocation setting.

We borrow the idea of stochastic inference and probabilistic meta-learning [36]–[39] to alleviate the task uncertainty issues arising from coarse-level user geolocation prediction. However, the parameter update in MetaGeo is different from previous methods. For example, these methods usually use the local reparameterization trick for the fully connected layer and flip the convolution layer to generate (nearly) completely independent weight samples. In contrast, we obtain KL-divergence by analysis and calculation and approximate the expectation values with the average of multiple samples from the approximate posterior.

VI. Conclusion

We presented a new perspective on the user geolocation prediction by casting the problem in the realm of meta-learning. We devised a few-shot learning protocol for training a number of sampled geolocation predictors to optimize the model performance on prediction adaptation, enabling Bayesian posterior inference to ease the geolocation task ambiguity issue and alleviate the location uncertainty. The empirical results show that our method not only achieves the state-of-the-art geolocation prediction performance in the conventional settings but is also able to identify users from unseen regions. Our future work includes investigating the effect of the proposed model on other location-based prediction problems, such as human mobility prediction and event location prediction. In addition, we will tackle the explainability of the model behavior (i.e., theoretical explanations of the ensembled meta-learners) and investigate the application of the proposed model on other graph learning tasks such as link prediction and graph classification.

Appendix
Evidence Lower Bound

The lower bound of the data with support set and query set split can be derived as

$$
\begin{aligned}
&\log\left(\prod_{i=1}^{B} p(\mathcal{Q}_i, \mathcal{S}_i)\right) \\
&= \log\left[\int p(\theta)\left(\prod_{i=1}^{B}\int p(\mathcal{Q}_i|\phi_i)p(\mathcal{S}_i|\phi_i)p(\phi_i|\theta)\right.\right. \\
&\qquad \left.\left.\times \frac{q_\psi(\theta)q_{\lambda_i}(\phi_i)}{q_\psi(\theta)q_{\lambda_i}(\phi_i)}d\phi_i\right)d\theta\right] \\
&= \log\left[\int \frac{p(\theta)}{q_\psi(\theta)}q_\psi(\theta)\left(\prod_{i=1}^{B}\int p(\mathcal{Q}_i|\phi_i)p(\mathcal{S}_i|\phi_i)\right.\right. \\
&\qquad \left.\left.\times \frac{p(\phi_i|\theta)}{q_{\lambda_i}(\phi_i)}q_{\lambda_i}(\phi_i)d\phi_i\right)d\theta\right] \\
&= \log\left[\mathop{\mathbb{E}}_{\theta\sim q_\psi}\left(\frac{p(\theta)}{q_\psi(\theta)}\prod_{i=1}^{B}\int p(\mathcal{Q}_i|\phi_i)p(\mathcal{S}_i|\phi_i)\right.\right. \\
&\qquad \left.\left.\times \frac{p(\phi_i|\theta)}{q_{\lambda_i}(\phi_i)}q_{\lambda_i}(\phi_i)d\phi_i\right)\right] \\
&\geq \mathop{\mathbb{E}}_{\theta\sim q_\psi}\left[\log\left(\frac{p(\theta)}{q_\psi(\theta)}\prod_{i=1}^{B}\int p(\mathcal{Q}_i|\phi_i)p(\mathcal{S}_i|\phi_i)\right.\right. \\
&\qquad \left.\left.\times \frac{p(\phi_i|\theta)}{q_{\lambda_i}(\phi_i)}q_{\lambda_i}(\phi_i)d\phi_i\right)\right] \\
&= \mathop{\mathbb{E}}_{\theta\sim q_\psi}\left[\log\left(\prod_{i=1}^{B}\int p(\mathcal{Q}_i|\phi_i)p(\mathcal{S}_i|\phi_i)\frac{p(\phi_i|\theta)}{q_{\lambda_i}(\phi_i)}q_{\lambda_i}(\phi_i)d\phi_i\right)\right] \\
&\quad - \mathop{\mathbb{E}}_{\theta\sim q_\psi}\log\frac{q_\psi(\theta)}{p(\theta)} \\
&= \mathop{\mathbb{E}}_{\theta\sim q_\psi}\left[\sum_{i=1}^{B}\log\left(\mathop{\mathbb{E}}_{\phi_i\sim q_{\lambda_i}} p(\mathcal{Q}_i|\phi_i)p(\mathcal{S}_i|\phi_i)\frac{p(\phi_i|\theta)}{q_{\lambda_i}(\phi_i)}\right)\right] \\
&\quad - \mathrm{KL}\big(q_\psi(\theta)\|p(\theta)\big) \\
&\geq \mathop{\mathbb{E}}_{\theta\sim q_\psi}\left[\sum_{i=1}^{B}\mathop{\mathbb{E}}_{\phi_i\sim q_{\lambda_i}}\left[\log\left(p(\mathcal{Q}_i|\phi_i)p(\mathcal{S}_i|\phi_i)\frac{p(\phi_i|\theta)}{q_{\lambda_i}(\phi_i)}\right)\right]\right] \\
&\quad - \mathrm{KL}\big(q_\psi(\theta)\|p(\theta)\big) \\
&= \mathop{\mathbb{E}}_{\theta\sim q_\psi}\left[\sum_{i=1}^{B}\left(\mathop{\mathbb{E}}_{\phi_i\sim q_{\lambda_i}}\log(p(\mathcal{Q}_i|\phi_i)p(\mathcal{S}_i|\phi_i))\right.\right. \\
&\qquad \left.\left.- \mathrm{KL}\big(q_{\lambda_i}(\phi_i)\|p(\phi_i|\theta)\big)\right)\right] \\
&\quad - \mathrm{KL}\big(q_\psi(\theta)\|p(\theta)\big) \qquad (20)
\end{aligned}
$$

where KL denotes the Kullback–Leibler divergence. Maximizing (20) can then be translated into the following minimization problem:

$$
\begin{aligned}
\arg\min \mathop{\mathbb{E}}_{\theta\sim q_\psi}\left\{-\sum_{i=1}^{B}\left(\mathop{\mathbb{E}}_{\phi_i\sim q_{\lambda_i}}\log(p(\mathcal{Q}_i|\phi_i)p(\mathcal{S}_i|\phi_i))\right.\right. \\
\left.\left.+ \mathrm{KL}\big(q_{\lambda_i}(\phi_i)\|p(\phi_i|\theta)\big)\right)\right\} + \mathrm{KL}\big(q_\psi(\theta)\|p(\theta)\big) \qquad (21)
\end{aligned}
$$

where we approximate the posterior $p(\phi_i|\theta)$ with a fixed number (L) of iterations using support data $\mathcal{S}_i$. This is achieved by gradient descent with samples in $\mathcal{S}_i$, which corresponds exactly to *maximum a posteriori* (MAP) inference under a Gaussian prior $p(\phi_i|\theta)$—however, the exact form of $p(\phi_i|\theta)$ is intractable. We alternatively interpret this MAP approximation as inferring a posterior of ϕ_i in the form of $p(\phi_i|\mathcal{S}_i, \theta)$, which is obtained via gradient descent on $\mathcal{S}_i$ starting from θ. Therefore, we reformulate the above-mentioned

objective as

$$\arg\min \mathop{\mathbb{E}}_{\theta\sim q_\psi}\left\{-\sum_{i=1}^{B}\mathop{\mathbb{E}}_{\phi_i\sim q_\theta(\phi_i|\mathcal{S}_i)}\log p(\mathcal{Q}_i|\phi_i) + \mathrm{KL}(q_\theta(\phi_i|\mathcal{S}_i)\|p(\phi_i|\mathcal{S}_i,\theta))\right\} + \mathrm{KL}\big(q_\psi(\theta)\|p(\theta)\big). \quad (22)$$

Recall that the maximum *a posteriori* estimate of ϕ_i corresponds to the global mode of the posterior $p(\phi_i|\mathcal{S}_i,\theta)$

$$p(\phi_i|\mathcal{S}_i,\theta)=\frac{p(\mathcal{S}_i|\phi_i,\theta)p(\phi_i|\theta)}{p(\mathcal{S}_i|\theta)}\propto p(\mathcal{S}_i|\phi_i)p(\phi_i|\theta) \quad (23)$$

where the global parameters θ are independent of $\mathcal{S}_i$.

REFERENCES

[1] A. Rahimi, T. Cohn, and T. Baldwin, "Semi-supervised user geolocation via graph convolutional networks," in *Proc. 56th Annu. Meeting Assoc. Comput. Linguistics*, 2018, pp. 2009–2019.

[2] X. Zheng, J. Han, and A. Sun, "A survey of location prediction on Twitter," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 9, pp. 1652–1671, Sep. 2018.

[3] B. Han, P. Cook, and T. Baldwin, "Geolocation prediction in social media data by finding location indicative words," in *Proc. COLING*, 2012, pp. 1045–1062.

[4] B. Wing and J. Baldridge, "Hierarchical discriminative classification for text-based geolocation," in *Proc. EMNLP*, 2014, pp. 336–348.

[5] T. H. Do, D. M. Nguyen, E. Tsiligianni, B. Cornelis, and N. Deligiannis, "Twitter user geolocation using deep multiview learning," in *Proc. ICASSP*, 2018, pp. 6304–6308.

[6] Y. Miura, M. Taniguchi, T. Taniguchi, and T. Ohkuma, "Unifying text, metadata, and user network representations with a neural network for geolocation prediction," in *Proc. ACL*, 2017, pp. 1260–1272.

[7] J. Bakerman, K. Pazdernik, A. G. Wilson, G. Fairchild, and R. Bahran, "Twitter geolocation: A hybrid approach," *ACM Trans. Knowl. Discovery Data*, vol. 12, no. 3, pp. 1–17, 2018.

[8] J. Eisenstein, B. O'Connor, N. A. Smith, and E. P. Xing, "A latent variable model for geographic lexical variation," in *Proc. EMNLP*, 2010, pp. 1277–1287.

[9] A. Ahmed, L. Hong, and A. J. Smola, "Hierarchical geographical modeling of user locations from social media posts," in *Proc. WWW*, 2013, pp. 25–36.

[10] B. Han, P. Cook, and T. Baldwin, "Text-based Twitter user geolocation prediction," in *Proc. JAIR*, vol. 49, pp. 451–500, 2014.

[11] M. Hulden, M. Silfverberg, and J. Francom, "Kernel density estimation for text-based geolocation," in *Proc. AAAI*, 2015, pp. 145–150.

[12] A. Rahimi, T. Cohn, and T. Baldwin, "A neural model for user geolocation and lexical dialectology," in *Proc. ACL*, 2017, pp. 209–216.

[13] L. Backstrom, E. Sun, and C. Marlow, "Find me if you can: Improving geographical prediction with social and spatial proximity," in *Proc. WWW*, 2010, pp. 61–70.

[14] L. Kong, Z. Liu, and Y. Huang, "SPOT: Locating social media users based on social network context," *Proc. VLDB Endowment*, vol. 7, no. 13, pp. 1681–1684, 2014.

[15] A. Rahimi, T. Cohn, and T. Baldwin, "Twitter user geolocation using a unified text and network prediction model," in *Proc. ACL*, 2015, pp. 630–636.

[16] M. Dredze, M. Osborne, and P. Kambadur, "Geolocation for Twitter: Timing matters," in *Proc. NAACL-HLT*, 2016, pp. 1064–1069.

[17] B. Han, P. Cook, and T. Baldwin, "A stacking-based approach to Twitter user geolocation prediction," in *Proc. ACL*, 2013, pp. 7–12.

[18] W.-H. Chong and E.-P. Lim, "Exploiting user and venue characteristics for fine-grained tweet geolocation," *ACM Trans. Inf. Syst.*, vol. 36, no. 3, p. 26, 2018.

[19] P. Zola, P. Cortez, and M. Carpita, "Twitter user geolocation using web country noun searches," *Decis. Support Syst.*, vol. 120, pp. 50–59, Dec. 2019.

[20] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," in *Proc. ICML*, vol. 2014, pp. 1188–1196.

[21] A. Grover and J. Leskovec, "Node2vec: Scalable feature learning for networks," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2016, pp. 855–864.

[22] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. ICLR*, 2017, pp. 1–14.

[23] M. Buda, A. Maki, and M. A. Mazurowski, "A systematic study of the class imbalance problem in convolutional neural networks," *Neural Netw.*, vol. 106, pp. 249–259, Oct. 2017.

[24] S. H. Khan, M. Hayat, M. Bennamoun, F. A. Sohel, and R. Togneri, "Cost-sensitive learning of deep feature representations from imbalanced data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 8, pp. 3573–3587, Feb. 2017.

[25] K. Cao, C. Wei, A. Gaidon, N. Arechiga, and T. Ma, "Learning imbalanced datasets with label-distribution-aware margin loss," in *Proc. NeurIPS*, 2019, pp. 1567–1578.

[26] M. Ren, W. Zeng, B. Yang, and R. Urtasun, "Learning to reweight examples for robust deep learning," in *Proc. ICML*, 2018, pp. 4334–4343.

[27] J. Shu *et al.*, "Meta-weight-net: Learning an explicit mapping for sample weighting," in *Proc. NeurIPS*, 2019, pp. 1919–1930.

[28] M. A. Jamal, M. Brown, M.-H. Yang, L. Wang, and B. Gong, "Rethinking class-balanced methods for long-tailed visual recognition from a domain adaptation perspective," in *Proc. CVPR*, 2020, pp. 7610–7619.

[29] S. Roller, M. Speriosu, S. Rallapalli, B. Wing, and J. Baldridge, "Supervised text-based geolocation using language models on an adaptive grid," in *Proc. EMNLP*, 2012, pp. 1500–1510.

[30] A. Rajaraman and J. D. Ullman, *Mining Massive Datasets*. Cambridge, U.K.: Cambridge Univ. Press, 2011.

[31] F. Wu, T. Zhang, A. H. D. Souza, Jr., C. Fifty, T. Yu, and K. Q. Weinberger, "Simplifying graph convolutional networks," in *Proc. ICML*, 2019, pp. 6861–6871.

[32] S. Thrun and L. Pratt, *Learning to Learn*. Norwell, MA, USA: Kluwer, 1998.

[33] A. Santoro, S. Bartunov, M. Botvinick, D. Wierstra, and T. P. Lillicrap, "Meta-learning with memory-augmented neural networks," in *Proc. ICML*, 2016, pp. 1842–1850.

[34] F. Chelsea, A. Pieter, and L. Sergey, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proc. ICML*, 2017, pp. 1126–1135.

[35] T. Munkhdalai and H. Yu, "Meta networks," in *Proc. ICML*, 2017, pp. 2554–2563.

[36] R. Amit and R. Meir, "Meta-learning by adjusting priors based on extended PAC-Bayes theory," in *Proc. ICML*, 2018, pp. 205–214.

[37] E. Grant, C. Finn, S. Levine, T. Darrell, and T. Griffiths, "Recasting gradient-based meta-learning as hierarchical Bayes," in *Proc. ICLR*, 2018, pp. 1–13.

[38] J. Yoon, T. Kim, O. Dia, S. Kim, Y. Bengio, and S. Ahn, "Bayesian model-agnostic meta-learning," in *Proc. NeurIPS*, 2018, pp. 7343–7353.

[39] S. Ravi and A. Beatson, "Amortized Bayesian meta-learning," in *Proc. ICLR*, 2019, pp. 1–14.

[40] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," in *Proc. ICLR*, 2014, pp. 1–14.

[41] A. Antoniou, H. Edwards, and A. Storkey, "How to train your MAML," in *Proc. ICLR*, 2019, pp. 1–11.

[42] C. M. Bishop and N. M. Nasrabadi, *Pattern Recognition and Machine Learning*, vol. 4, no. 4. New York, NY, USA: Springer, 2006.

[43] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, "Weight uncertainty in neural networks," in *Proc. ICML*, 2015, pp. 1613–1622.

[44] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. ICLR*, 2015, pp. 1–15.

[45] M. Bauer, M. Rojas-Carulla, J. B. Swiatkowski, B. Schölkopf, and R. E. Turner, "Discriminative k-shot learning using probabilistic models," *CoRR*, vol. abs/1706.00326, pp. 1–12, Dec. 2017.

[46] C. Finn and S. Levine, "Meta-learning and universality: Deep representations and gradient descent can approximate any learning algorithm," in *Proc. ICLR*, 2018, pp. 1–20.

[47] P. Li, H. Lu, N. Kanhabua, S. Zhao, and G. Pan, "Location inference for non-geotagged tweets in user timelines," *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 6, pp. 1150–1165, Jun. 2019.

[48] O. Vinyals *et al.*, "Matching networks for one shot learning," in *Proc. NeurIPS*, 2016, pp. 3630–3638.

[49] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," in *Proc. NeurIPS*, 2017, pp. 4077–4087.

[50] S. Ravi and H. Larochelle, "Optimization as a model for few-shot learning," in *Proc. ICLR*, 2017, pp. 1–11.

[51] Z. Cheng, J. Caverlee, and K. Lee, "You are where you tweet: A content-based approach to geo-locating Twitter users," in *Proc. CIKM*, 2010, pp. 759–768.
[52] R. Li, S. Wang, and K. C. Chang, "Multiple location profiling for users and relationships from social network and content," *Proc. VLDB Endowment*, vol. 5, no. 11, pp. 1603–1614, 2012.
[53] D. Jurgens, "That's what friends are for: Inferring location in online social media platforms based on social relationships," in *Proc. ICWSM*, 2013, pp. 273–283.
[54] J. McGee, J. Caverlee, and Z. Cheng, "Location prediction in social media based on tie strength," in *Proc. CIKM*, 2013, pp. 459–468.
[55] A. Rahimi, T. Baldwin, and T. Cohn, "Continuous representation of location for geolocation and lexical dialectology using mixture density networks," in *Proc. EMNLP*, 2017, pp. 167–176.
[56] T. Miyazaki, A. Rahimi, T. Cohn, and T. Baldwin, "Twitter geolocation using knowledge-based methods," in *Proc. EMNLP*, 2018, pp. 7–16.
[57] S. Li, C. Zhang, D. Lei, J. Li, and J. Han, "GeoAttn: Localization of social media messages via attentional memory network," in *Proc. SDM*, 2019, pp. 64–72.
[58] E. Amitay, N. Har'El, R. Sivan, and A. Soffer, "Web-a-where: Geotagging content," in *Proc. SIGIR*, 2004, pp. 273–280.
[59] B. P. Wing and J. Baldridge, "Simple supervised document geolocation with geodesic grids," in *Proc. ACL*, 2011, pp. 955–964.
[60] W.-H. Chong and E.-P. Lim, "Tweet geolocation: Leveraging location, user and peer signals," in *Proc. CIKM*, 2017, pp. 1279–1288.
[61] M. Ebrahimi, E. ShafieiBavani, R. Wong, and F. Chen, "A unified neural network model for geolocating Twitter users," in *Proc. CoNLL*, 2018, pp. 42–53.
[62] W.-H. Chong and E.-P. Lim, "Fine-grained geolocation of tweets in temporal proximity," *Trans. Inf. Syst.*, vol. 37, no. 2, pp. 1–33, 2019.
[63] L. Chen, C. Liu, R. Zhou, J. Li, X. Yang, and B. Wang, "Maximum co-located community search in large scale social networks," *Proc. VLDB Endowment*, vol. 11, no. 10, pp. 1233–1246, 2018.
[64] Y. Chen, J. Xu, and M. Xu, "Finding community structure in spatially constrained complex networks," *Int. J. Geograph. Inf. Sci.*, vol. 29, no. 6, pp. 889–911, 2015.
[65] V. Garcia and J. Bruna, "Few-shot learning with graph neural networks," 2017, *arXiv:1711.04043*.
[66] Z. Wang, B. Du, and Y. Guo, "Domain adaptation with neural embedding matching," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 7, pp. 2387–2397, Jul. 2019.
[67] J. Gu, Y. Wang, Y. Chen, V. O. Li, and K. Cho, "Meta-learning for low-resource neural machine translation," in *Proc. EMNLP*, 2018, pp. 3622–3631.
[68] G. Cai, Y. Wang, L. He, and M. Zhou, "Unsupervised domain adaptation with adversarial residual transform networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 8, pp. 3073–3086, Sep. 2019.
[69] X. S. Zhang, F. Tang, H. H. Dodge, J. Zhou, and F. Wang, "Metapred: Meta-learning for clinical risk prediction with limited patient electronic health records," in *Proc. KDD*, 2019, pp. 2487–2495.
[70] F. Zhou, C. Cao, K. Zhang, G. Trajcevski, T. Zhong, and J. Geng, "Meta-GNN: On few-shot node classification in graph meta-learning," in *Proc. CIKM*, 2019, pp. 2357–2360.
[71] H. Yao *et al.*, "Graph few-shot learning via knowledge transfer," in *Proc. AAAI*, 2020, pp. 6656–6663.
[72] Y. Sun, K. Tang, Z. Zhu, and X. Yao, "Concept drift adaptation by exploiting historical knowledge," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 10, pp. 4822–4832, Oct. 2018.
[73] H. Yao, Y. Liu, Y. Wei, X. Tang, and Z. Li, "Learning from multiple cities: A meta-learning approach for spatial-temporal prediction," in *Proc. WWW*, 2019, pp. 2181–2191.
[74] W.-Y. Chen, Y.-C. Liu, Z. Kira, Y.-C. F. Wang, and J.-B. Huang, "A closer look at few-shot classification," in *Proc. ICLR*, 2019, pp. 1–17.
[75] G. Koch, R. Zemel, and R. Salakhutdinov, "Siamese neural networks for one-shot image recognition," in *Proc. ICML Learn. Workshop*, 2015, pp. 1–30.
[76] L. Bertinetto, J. F. Henriques, J. Valmadre, P. Torr, and A. Vedaldi, "Learning feed-forward one-shot learners," in *Proc. NeurIPS*, 2016, pp. 523–531.
[77] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. Torr, and T. M. Hospedales, "Learning to compare: Relation network for few-shot learning," in *Proc. CVPR*, 2018, pp. 1199–1208.
[78] R. Vuorio, S.-H. Sun, H. Hu, and J. J. Lim, "Multimodal model-agnostic meta-learning via task-aware modulation," in *Proc. NeurIPS*, 2019, pp. 1–12.
[79] C. Finn, K. Xu, and S. Levine, "Probabilistic model-agnostic meta-learning," in *Proc. NeurIPS*, 2018, pp. 9537–9548.

Fan Zhou (Member, IEEE) received the B.S. degree in computer science from Sichuan University, Chengdu, China, in 2003, and the M.S. and Ph.D. degrees from the University of Electronic Science and Technology of China, Chengdu, in 2006 and 2012, respectively.

He is currently an Associate Professor with the School of Information and Software Engineering, University of Electronic Science and Technology of China. His research interests include machine learning, neural networks, spatiotemporal data management, graph learning, social network data mining, and knowledge discovery.

Xiuxiu Qi received the M.S. degree from the University of Electronic Science and Technology of China, Chengdu, China, in 2021.

She is currently an Engineer with ByteDance, Beijing, China. Her research interests include few-shot learning, social network analysis, knowledge discovery, and data mining.

Kunpeng Zhang received the Ph.D. degree in computer science from Northwestern University, Evanston, IL, USA, in 2013.

He is currently an Assistant Professor with the Department of Information Systems, Smith School of Business, University of Maryland, College Park, MD, USA. He is a Researcher in the area of large-scale data analysis, with particular focuses on social data mining, image understanding via machine learning, social network analysis, and natural language processing. He has authored or coauthored papers in the area of social media, artificial intelligence, network analysis, and information systems on top conferences and journals.

Dr. Zhang serves as a program committee member for many conferences and an associate editor for journals.

Goce Trajcevski (Member, IEEE) received the B.Sc. degree from the Saints Cyril and Methodius University, Skopje, North Macedonia, in 1989, and the M.S. and Ph.D. degrees from the University of Illinois at Chicago, Chicago, IL, USA, in 1995 and 2002, respectively.

He is currently an Associate Professor with the Department of Electrical and Computer Engineering, Iowa State University, Ames, IA, USA. He has coauthored a book chapter, three encyclopedia chapters, and 140 publications in refereed conferences and journals. His research has been funded by the National Science Foundation (NSF), The Office of Naval Research (ONR), BEA Systems, Inc., and Northrop Grumman Corporation, Falls Church, VA, USA. His research interests include spatiotemporal data management, uncertainty, and reactive behavior management in different application settings, and incorporating multiple contexts.

Dr. Trajcevski has served in various roles in organizing committees in numerous conferences and workshops. He was the General Co-Chair of the IEEE International Conference on Data Engineering (ICDE) 2014 and ACM Special Interest Group on SPATIAL information (SIGSPATIAL) 2019, the PC Co-Chair of the European Conference on Advances in Databases and Information Systems (ADBIS) 2018 and ACM SIGSPATIAL 2016 and 2017. He is an Associate Editor of the *ACM Transactions on Spatial Algorithms and Systems* (TSAS) and the *Geoinformatica* journals.

Ting Zhong received the B.S. degree in computer application and the M.S. degree in computer software and theory from Beijing Normal University, Beijing, China, in 1999 and 2002, respectively, and the Ph.D. degree in information and communication engineering from the University of Electronic Science and Technology of China, Chengdu, China, in 2009.

Dr. Zhong has beene an Associate Professor with the School of Information and Software Engineering, University of Electronic Science and Technology of China, since 2010. Her research interests include machine learning, social network analysis, and mobile computing.