Version of Record: https://www.sciencedirect.com/science/article/pii/S1566253521002293
Manuscript_73bf8e5b91ef2c7d371b12e46fa81012

# Identifying User Geolocation with Hierarchical Graph Neural Networks and Explainable Fusion

Fan Zhou[a], Tianliang Wang[a], Ting Zhong[a,*] and Goce Trajcevski[b]

[a] University of Electronic Science and Technology of China, China
[b] Iowa State University, USA

ARTICLE INFO

*Keywords*:
user geolocation
social information fusion
graph neural networks
interpretable fusion
influence function

ABSTRACT

Determining user geolocation from social media data is essential in various location-based applications – from improved transportation/supply management, through providing personalized services and targeted marketing, to better overall user experiences. Previous methods rely on the similarity of user posting content and neighboring nodes for user geolocation, which suffer the problems of: (1) position-agnostic of network representation learning, which impedes the performance of their prediction accuracy; and (2) noisy and unstable user relation fusion due to the flat graph embedding methods employed. This work presents Hierarchical Graph Neural Networks (HGNN) – a novel methodology for location-aware collaborative user-aspect data fusion and location prediction. It incorporates geographical location information of users and clustering effect of regions and can capture topological relations while preserving their relative positions. By encoding the structure and features of regions with hierarchical graph learning, HGNN can primarily alleviate the problem of noisy and unstable signal fusion. We further design a relation mechanism to bridge connections between individual users and clusters, which not only leverages the information of isolated nodes that are useless in previous methods but also captures the relations between unlabeled nodes and labeled subgraphs. Furthermore, we introduce a robust statistics method to interpret the behavior of our model by identifying the importance of data samples when predicting the locations of the users. It provides meaningful explanations on the model behaviors and outputs, overcoming the drawbacks of previous approaches that treat user geolocation as "black-box" modeling and lacking interpretability. Comprehensive evaluations on real-world Twitter datasets verify the proposed model's superior performance and its ability to interpret the user geolocation results.

## 1. Introduction

The plethora of Online Social Networks (OSN) have enabled novel interactions in daily activities – e.g., sharing notifications about events related to product descriptions and traffic jams; sharing personal experiences on Instagram and Facebook; reading news and popular topics on Twitter; building academic connections on ResearchGate, etc. These have not only changed our way of communication, reading, and social activities but also enabled a generation of an unprecedented volume of heterogeneous data, which, in turn, fosters business innovations and emerging industrial opportunities [13]. Among various applications, identifying the geographic locations of users receives lasting interest from both academia and industry and has become an essential Internet service for many industrial services, such as location-based targeted advertising, emergency location identification, political elections, substance use surveillance, local event/place recommendation and natural disaster response [20, 71, 74].

Fine-grained localization, such as various sensor-based tracking of assets and processes, have already been exploited in multiple industrial applications. However, in more extensive geographical settings, there is the issue of inaccuracy due to, e.g., cellular access restrictions, high measurement overhead, and unreliable client response times [45]. Complementary to this, the increased popularity of social media services (e.g., Twitter, Facebook, and Instagram) provide rich and timely metadata, e.g., published message contents, mention tags, and follow/followee relations. This information could be efficiently leveraged to promptly geolocate OSN users – which has recently spurred research interest in the, so calls, User Geolocation (UG) problem in OSN [15, 24, 42, 50]. For example, the CDC (centers for disease control and prevention) has been utilizing social media to help the epidemiological investigation in responding to the virus that causes COVID-19 [59].

Online user geolocation is a passive crowd-sensing problem that requires hybrid information fusion and insights from many user activities and sensing data to distill the knowledge and refine the predicted results. Early efforts [24, 54] mainly focused on mining indicative information from users' posting content relying on indicative words that can link users to their home locations, based on various natural language processing techniques (e.g., topic models and statistic models). For example, Term Frequency–Inverse Document Frequency (TF-IDF [29]) is a commonly used method to measure the distribution of location words [24]. More recent efforts fuse users interactions for collaborative sensing and boosting the geolocation accuracy – e.g., node2vec [19] is used to learn repre-

*Corresponding author
fan.zhou@uestc.edu.cn (F. Zhou); mortonwang@outlook.com (T. Wang); zhongting@uestc.edu.cn (T. Zhong); gocet25@iastate.edu (G. Trajcevski)
ORCID(s): 0000-0002-8038-8150 (F. Zhou); 0000-0003-0251-3785 (T. Wang); 0000-0002-8163-3146 (T. Zhong); 0000-0002-8839-6278 (G. Trajcevski)

sentation of users [15], combined with text representation via doc2vec [36] to predict user locations in an end-to-end manner. Recurrent Neural Networks (RNNs) with attention mechanism to model user tweet content are also used in [42], further combining the metadata such as timezone and self-declared profiles to predict user locations. A more recent work [50] employs GCNs [32] for learning network structures with graph convolution and pooling operations.

Broadly speaking, the existing state-of-the-art methods employ deep learning techniques for learning user interaction and content representation – without fully exploiting the specific constraints in the user geolocation task. When learning user interactions, graph representation methods (e.g., GCN [32], GAT [57], node2vec [19], GraphSAGE [21]) are commonly used – however, the approaches are *general, unweighted* and *location-agnostic* graph learning methods, without considering the geographical *position/location* of nodes (users). Since the graph embedding methods are not specifically tailored for user geolocation task, existing approaches ignore the strong *geolocaiton dependencies* among nodes and thus cannot capture the relative distance between any pair of nodes. In addition, existing graph-based UG methods are inherently *flat* graph learning models, which cannot capture the region-level features and thus are very sensitive to local network structure. For example, the homphily assumption, i.e., online interactions imply a higher probability of geographical proximity, is not held in many cases [14, 71].

Our main motivation is based on the observation that the methodologies in the existing literature do not exploit the benefits of joint consideration of identifying the topological structure of users along with the influence of crowds from different regions. While the former is usually noisy and unstable, the latter may provide a more robust signal for geolocating. In addition, existing models, especially those based on deep neural networks, often lack transparency and cannot interpret model behavior and localization results. Thus, their applicability in safety-critical areas is restricted. For example, when locating area with specific emergencies (for example, the spread of COVID-19), it would be more significant to explain why and how such a prediction was made instead of just presenting the predicted results [2, 26, 39].

To address the aforementioned limitations of previous works, we propose a novel multi-view user geolocation framework, called Hierarchical Graph Neural Networks (HGNN), to fuse user-generated content and network information for collaborative user geolocation. It enhances user geolocation performance from the following aspects. First, it incorporates the relative distances of each node to other nodes (clusters) in the network, which enables the model to discriminate the nodes having similar topological structures but residing in different regions. Second, the hierarchical feature fusion method that we propose provides both coarse- and fine-grained graph representation by learning and distinguishing the crowd effects from different geographic regions. Third, our model naturally exploits unlabeled and isolated nodes for context information aggregation, which are absent in previous UG models. Fourth, the interpretability of information fusion allows us to understand the trained geolocation model's behavior and how it is affected by the information aggregated from the training samples (i.e., all in-network users and their associated features). The main contributions of this work in terms of the novelty of the proposed approach are four-fold. Specifically, we present:

- A new location-aware node relation learning model that takes the geographical location and relative distance into account when performing non-linear transformation and feature aggregation, which not only preserves network topology but also encodes node position with respect to the other nodes and/or clusters.
- A new hierarchical GNN framework that learns both region- and node-level features for robust feature aggregation and propagation, which can be combined with *any* graph learning approaches in an end-to-end manner. Compared to flat node-level embedding in existing UG approaches, we are able to alleviate the influence of noisy interactions and the impact of outlier nodes.
- A new general framework to explain the behavior of user geolocation models and the prediction results. We take the initiatives to use influence function [33] to quantify the impact of in-network users and corresponding features on the predicted outcomes.
- Extensive evaluations on three benchmark Twitter datasets. The results demonstrate that our method significantly outperforms the state-of-the-art baselines while providing explanations on both model behavior and detection results.

In the rest of this paper, Section 2 reviews the related work, followed by Section 3 that formalizes the problem and presents the necessary backgrounds. In Section 4, we give the details of the methodology, as well as the approach for explaining the user-aspect data fusion and location prediction. Experimental evaluations quantifying the benefits of our approach are performed in Section 5. We conclude this work and outline directions for future work in Section 6.

## 2. Related Work

In the body of previous works on geolocating online social networks, the models can be broadly categorized into three groups according to the type of data used to make the prediction. We now review relevant works and position our paper in the context of the existing literature.

### 2.1. Content-based Approaches

User-generated content (UGC) such as textual posts and photos may be casually attached with real-time locations facilitated by the increasing popularity of GPS-equipped devices. However, these geo-tagged tweets are extremely sparse, e.g., no more than 1% of published tweets are labeled with geographical locations [8]. A plethora of

works [1, 24, 25, 54, 61] have studied the possibility of leveraging UGC for locating users. These methods address the geolocation problem by inferring locations from the location-relevant words with various classification models. Therefore, identifying meaningful indicative words is an important step towards accurate user geolocation, where TF-IDF [29] is a widely adopted textual content representation method in the literature [24, 47, 48, 50, 52]. For example, inverse location/city frequency has been used to measure the location words in the content [24, 52]. In contrast, probabilistic models are usually used to characterize the users' location distributions w.r.t. their published UGC, which, however, requires extensive manually labeled location-related words to achieve satisfactory results.

Inspired by recent advances in applying deep learning in natural language processing, a few studies turn to model users' textual contents with various neural networks based models in order to learn the tweet representation in an end-to-end manner [15, 42, 49, 51]. Among these methods, doc2vec [36] and recurrent neural networks (RNNs) are simple yet effective choices for learning vector representation of textural contents. For example, in [15], combining TF-IDF and doc2vec representations of textual information is proposed to enhance the prediction performance. GRU [10] with attention mechanism [4] was used in [42] to model user tweet content and obtain a timeline representations. Though doc2vec and RNN-based methods can learn the language characteristics efficiently without manual location feature engineering, a recent study [22] finds that TF-IDF is consistently superior to doc2vec due to the location-indicative words captured in TF-IDF.

Our present work enables better location-awareness than the existing literature and, in particular, HGNN distinguishes the crowd effects from different geographic regions.

## 2.2. Network-based Methods

Online social relationships are also important indicators for user geolocation under the homophily assumption [3, 11, 34, 53], i.e., people prefer to interact with others in nearby areas. Backstrom et al. [3] examine the relationship between users' geographical proximity and online friendships on Facebook, and find that the likelihood of relations between any user pair drops monotonically as a function of distance. Rather than solely relying on friendships, more and more works utilize various types of connections, such as the co-mention tags and mentions between non-friends, to construct closer social interactions beyond friendships [49, 71]. In this way, similar interests among users can be retrieved from such implicit networks to improve geolocation accuracy [30, 41, 51]. Moreover, researchers also identify some noisy interaction factors that may degrade the prediction performance. For example, social influence of celebrities is a distracting factor that may confuse the prediction and thus is removed from the built user network [38, 51].

Although the existing approaches have tackled the aspect of explicitly modeling location dependency between social connected users, some challenges have not been properly addressed – namely, the sparsity of geo-tagged users and the inaccurate label propagation. More importantly, friends' locations are usually contradicting each other, which hinders the practical applicability of these works. In contrast, our HGNN learns both region-level and node-level features and aggregates them in a manner that provides better intepretability.

## 2.3. Multi-Information Fusion based Models

Recent efforts have leveraged deep graph learning methods to model user interaction networks by fusing user-generated contents and various meta-data, such as user profiles, tweeting time, and user timezone. For example, MENET [15] exploits node2vec [19] to learn user representations, combined with text representation learned by doc2vec, for predicting users' locations. Another work [50] employs GCNs [32] for learning network structures with the graph convolution and pooling operations, which has achieved state-of-the-art geolocation performance. A recent work [22] investigate several graph embedding methods and found that NetMF [46] performs better than node2vec and GraphSAGE [21] on user geolocation task, but does not show superior performance than GCN-based models [22, 50].

It is worth noting that some works make use of various meta-data (e.g., self-declared location in profile and timezone information) for improving the prediction performance. For example, user timezone, as well as UTC offset and country noun, have been used for user geolocation [15, 42, 48, 49, 79]. While such auxiliary information is a strong indicator for regularizing the locations the model predicted, a majority of users are not willing to open this privacy information, which is sometimes camouflaged or posted casually. We further note that there is another line of efforts [5, 9, 11, 37, 43] studying the Twitter message geolocation problem which tries to identify the tweeting locations rather than the Twitter user location discussed in this work.

Despite the promising results on improving geolocation performance, existing state-of-the-art methods fail to identify the importance of individual users that we addressed in this work. Arguably, while various graph embedding techniques can be utilized for network representation in user geolocation, understanding the influence of user connections is more important for interpreting the behavior of the geolocation models and therefore benefits downstream decision makings. In this spirit, we initiate the attempt to analyze theoretically and experimentally how the properties of graph structures influence the geolocation performance. This not only demystifies and interprets the predictions made by the model but outlines the underlying constraints of existing approaches, which, in turn, should be taken into consideration in modeling and predicting user geolocation.

## 2.4. Graph Neural Networks

Graph neural networks are effective methods models for analyzing and learning from data on graphs, and have been

successfully applied to a variety of domains including image processing [44], social networks [76], transportation systems [77], etc. Existing GNN models vary from each others on message passing mechanisms, while most of them rely on flat information aggregations [32]. There are several hierarchical GNN frameworks that gradually coarsen the original graph with pooling operation for graph classification [67, 70] and image recognition [44, 73]. The main difference with our work is how HGNN model defines the graph hierarchy for clusters and exploits the geographic information. Directly applying GraphPool [67] or HGP-SL [70] for UG task is problematic since both of them fail to consider the relative location of nodes w.r.t. other nodes/clusters and cannot cluster the unlabeled nodes. Another related work is PGNN [68], recently proposed to learn the relative position of nodes. However, it does not leverage nodes' geographic information that is critical for UG. More importantly, all these methods are suitable for *fully connected* graph learning, while our HGNN model is capable of incorporating unlabeled and isolated nodes and thus is more suitable for UG task.

Despite the promising performance gains on many graph tasks, most GNNs are still black-box models without human-understandable model behaviors and explanations. Although GAT [57] can learn the importance of edges and thus, to some extent, explain the node aggregation behaviors via attention mechanism, it is limited to specific architectures and fails to provide single-instance explanations. To adaptively adjust the influence of each node, a learnt exploitation of information from neighborhoods of differing locality and selective combining of different aggregations was proposed in [65]. Though their method can automatically discover the importance of each node in a GNN, it is not specifically designed for explaining model predictions. GNNExplainer [66] was proposed to explain the predictions of model-agnostic GNNs. It interprets the GNN models by maximizing the mutual information between a subgraph (or a subset of node features) and the predictions for the original graph. Another work [6] uses image interpretation methods, such as sensitivity analysis, guided backpropagation, and layer-wise relevance propagation (LRP), to explain the node-level predictions. GraphLIME [28] is a local interpretable method that captures the nonlinear dependency between features and predictions. It then considers the perturbation near a node and uses a linear explanation model to find features as explanations for GNNs. X-GNN [69] proposes to find the graph patterns that maximize a particular prediction through graph generation, which is formulated as a reinforcement learning problem and trained with a policy gradient method. GNN-LRP [55] is a theoretically founded XAI method for interpreting GNN predictions, which is derived from the higher-order Taylor expansions based on LRP. A recent work [27] systematically reviews existing explainable GNN methods, and proposes to enable information fusion for multi-modal causability using interpretable GNNs.

What separates our work from the existing GNN-based approaches is that we propose a learning model which incorporates the geolocations and distances and we provide a greater extent of explainability.

## 3. Preliminaries

Without loss of generality, we use as a running illustrative scenario the domain of Twitter, and explore the problem of geolocation detection based on *tweets* – i.e., short texts with no more than 140 characters. Some auxiliary information is typically associated with a tweet, describing specific semantic aspects – e.g., "@" means *mention* another Twitter accounts, and words starting with "#" are *hashtags*, which indicate mentioning a topic. In Table 1, we summarize the frequently used notations in this paper.

### 3.1. User Content

For each user, we combine his/her messages as linguistic content, including both tweet messages by himself and retweets forwarding other users' postings. Following previous works [42, 50], we filter out photos and symbols for each user. We denote content embedding vector of a user $v$ as $\mathbf{x}_v$, and the vectors for all users as $\mathbf{X}$.

### 3.2. Mention Graph

In addition to text, we construct a mentioning graph to represent social relationships among users by extracting mention (@-somebody) information from the content. The mention graph is defined as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is a set of all users (nodes) where a node $v \in \mathcal{V}$ is associated with a feature (tweet content) vector $\mathbf{x}_v \in \mathbf{X}$. An edge exists between two nodes $v_i$ and $v_j$ either if they are social friends, or one has re-tweeted an item from the other. Note that the graph need not be fully connected, i.e., some *isolated* nodes might exist.

### 3.3. Problem Definition

We focus on predicting the "home" location of the user [71], i.e., the location that a user most probably resides in. Since geolocations are typically described by longitude and latitude, we formulate this problem as a classification task by dividing the surface of earth into closed and *non-overlapping clusters* using $k$-d tree [7] following and improving upon the related works [48, 50]. Thus, each user is tagged with one (and only one) label indicating the cluster he/she belongs to. We denote labels (clusters) as $\mathbf{Y} \in \mathbb{R}^{n \times c}$, where $n$ is the number of all users, and $c$ is the number of clusters. The geolocation problem can now be more formally phrased as follows:

**Definition 1.** ***User Geolocation from Social Media Content:*** *Given the tweet contents* $\mathbf{X}$ *and the mention graph* $\mathcal{G}$*, as well as partially labeled users, we identify the geographical locations of unlabeled users through information fusion of user generated contents and social relationships.*

**Table 1**
Frequently used notations.

| Notation | Description |
|---|---|
| $\mathcal{G}$ / $\mathcal{E}$ | mention network with a set of edges. |
| $v$ / $\mathcal{V}$ | a user (a node in $\mathcal{G}$) / all users (nodes of $\mathcal{G}$). |
| $\mathcal{N}(v)$ | the neighborhoods of node $v$ in $\mathcal{G}$. |
| $\mathbf{x}_v$ / $\mathbf{X}$ | content representation of user $v$ / all users. |
| $\mathbf{X}'$ | the updated representations of all users. |
| $\mathbf{Y}$ / $\mathbf{Y}'$ | truth label / predicted label of all users. |
| $k$ / $K$ | $k$-th layer of model / the total layers. |
| $\theta^{(k)}$ | the parameters at $k$-th layer. |
| $n$ / $c$ | number of all users / clusters. |
| $d$ | the dimension of initial $\mathbf{x}_v$. |
| $\ell$ | loss of model training. |
| $s_j$ / $S$ | $j$-th cluster / the set of all clusters. |
| $\mathbf{s}_j^{(k)}$ / $\mathbf{S}^{(k)}$ | features of $j$-th / all cluster at $k$-th layer. |
| $s_j^v$ | reachable node set in $s_j$ of user $v$. |
| $\mathcal{G}_s$ | the cluster graph. |
| $\mathbf{A}_s$ | the weighted adjacency matrix of $\mathcal{G}_s$. |
| $\mathbf{D}_s$ | the degree matrix of $\mathbf{A}_s$. |
| $e_i = (\mathbf{x}_i, \mathbf{y}_i)$ | attribute vector and its label of $i$-th sample. |
| $\theta^*$ / $\theta^*_{-e}$ | the optimal parameters with / without $e$. |
| $m$ | the number of all training samples. |
| $\gamma$ | small upweighted value on $e$. |
| $\mathbf{H}$ | the Hessian matrix. |
| $\psi$ | the influence value. |
| $e_{\text{test}}$ | a testing sample. |

### 3.4. Graph Neural Networks (GNNs)

GNNs is a powerful tool for graph representation learning, which has received increasing attention over the past years [63, 78]. A GNN model consists of a stack of neural network layers, where each layer aggregates neighborhood information around each node and then passes the aggregated information to the next layer. Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and node features $\mathbf{x}_v \in \mathbf{X}$ of the node $v$, the GNN model try to calculate the node vectors in an iterative manner. Specifically, in the $k$-th ($k > 0$) layer, the model updates the nodes as following:

$$\mathbf{x}_v^{(k)} = f_m^{\theta_2}\left(\mathbf{x}_v^{(k-1)}, f_a^{\theta_1}\left(\left\{\mathbf{x}_u^{(k-1)} \middle| u \in \mathcal{N}(v)\right\}\right)\right), \quad (1)$$

where $\mathcal{N}(\cdot)$ denotes the neighbor nodes, and parameters $\theta_1$ and $\theta_2$ are trained with the model. The aggregation function $f_a^{\theta_1}$ updates the features from neighbors using one of the operations such as *Mean* and *Pooling* [21]. $f_m^{\theta_2}$ is to merge node's representations from previous $k-1$ step. The two functions can be implemented with any arbitrary differentiable, permutation-invariant functions such as deep neural networks. The learned node embeddings can be used for downstream tasks including link prediction [32], node classification [57], user geolocation [50], location/trip recommendation [75], information cascade prediction [76], traffic forecasting [77], etc.

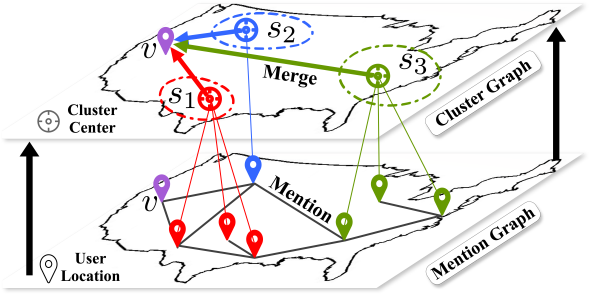



**Figure 1:** Illustration of the basic idea of HGNN. (1) Aggregating nodes features to form cluster attributes; and (2) Updating node features by merging cluster attributes. Colors represent node or cluster connections across the maps; dotted circles indicate clusters.

## 4. HGNN: Structure and Methodology

The HGNN aims to learn both cluster-level and node-level influences of crowds and individual users, respectively. Figure 1 briefly illustrates our model setup. On the cluster-level, it aggregates the features of users in the same cluster so as to form the regional influence. The learned regional attributes embed location information and content features, which are propagated by a GNN on the cluster graph. On the node-level, nearby regions for each user based on the hops-distance (i.e., the topology of connectivity) are extracted. Combined with the regional influence, these are fed into another GNN to learn the node representation. In this way, the relative position of each node to the clusters can be captured for boosting the UG performance.

### 4.1. Cluster-level GNN

Let $s_j$ be the $j$-th cluster, whose location is the geographical centroid of all the nodes within $j$-th cluster. The set of all clusters is denoted by $S = \{s_1, \ldots, s_c\}$, which is also considered as the vertices set of cluster graph. For each cluster $s_j$, we utilize an aggregation function $f_a$ to learn its attributes $\mathbf{s}_j^{(k-1)}$ by fusing the content features of users in this cluster:

$$\mathbf{s}_j^{(k-1)} = f_a\left(\left\{\mathbf{x}_u^{(k-1)}, u \in s_j\right\}\right), \quad (2)$$

where $\mathbf{x}_u^{(k-1)}$ represents the content features of user $u$ in $(k-1)$-th layer. There are many choices for function $f_a$, such as *MEAN*, *MIN*, *MAX*, *SUM* and *LSTM* [21]. Here we choose *SUM* as the aggregation function due to its expressiveness and injectivity property [64].

Now we can construct a cluster graph $\mathcal{G}_s = (S, \mathbf{A}_s)$, where $\mathbf{A}_s \in \mathbb{R}^{c \times c}$ is a weighted adjacency matrix representing the geographic distance between any pair of clusters. Specifically, each element $A_s^{ij} = 1 - \text{Haversine}(s_i, s_j)/\texttt{Max}$, where `Max` is the maximum distance calculated by Haversine$(\cdot)$ distance [56]. Here we use Haversine distance because it can determine the great-circle distance between any two points (cluster centers) on a sphere, given their longitudes and latitudes. In practice, we first calculate the distance by Haversine$(,\cdot,)$ between every

pair of clusters, since each cluster has its own cluster center with deterministic coordinates. Then, the longest distance is denoted as `Max` and used to normalize the rest distances.

We use a message passing function to learn the coarsened cluster graph by propagating the aggregated features $\mathbf{s}_j^{(k-1)} \in \mathbf{S}^{(k-1)}$ of clusters into the next layer. To make our method general, we adopt a vanilla GCN [32] as the message passing function:

$$\mathbf{S}^{(k)} = \sigma\left(\mathbf{D}_s^{-\frac{1}{2}}\mathbf{A}_s\mathbf{D}_s^{-\frac{1}{2}}\mathbf{S}^{(k-1)}\theta_1^{(k)}\right), \tag{3}$$

where $\mathbf{D}_s$ is the degree matrix of $\mathbf{A}_s$, $\theta_1^{(k)}$ is a layer-specific trainable weight matrix, and $\sigma(\cdot)$ denotes an activation function – which is $\mathrm{ReLU}(\cdot) = \max(0, \cdot)$ in our implementation. Now, we can update attributes of all regions $\mathbf{S}^{(k)}$ by aggregating the information from neighboring clusters, which will be scaled by the topological distance of each user in the next step. In addition, we have following result assuming that feature aggregation is invariant under node permutations in each cluster.

**Proposition 1.** *The cluster-level GNN learning in HGNN is permutation invariant for as long as the aggregation function $f_a(\cdot)$ is permutation invariant.*

*Proof.* Given an arbitrary set $s_j = \{u_1, \ldots, u_\alpha\}$ and a bijective function $\pi(\cdot)$ whose domain and codomain are both $s_j$, i.e., $\pi(s_j) = s_j$. If $f_a(\cdot)$ is node permutation invariant, the cluster-level feature aggregation (Eq. (2)) would not change if the inputs are reordered, i.e., $f_a(\{\mathbf{x}_{u_1}, \ldots, \mathbf{x}_{u_\alpha}\}) \equiv f_a(\{\mathbf{x}_{\pi(u_1)}, \ldots, \mathbf{x}_{\pi(u_\alpha)}\})$, where $\mathbf{x}_{u_i}$ and $\mathbf{x}_{\pi(u_i)}$ are the features of nodes $u_i \in s_j$ and $\pi(u_i) \in \pi(s_j)$, respectively. □

We note that one can readily replace the GCN used here with other more sophisticated GNN models such as GraphSAGE [21], GAT [57], SGC [62].

## 4.2. Node-level GNN

Above we have learned the regional attributes for each cluster, which could capture high level features of different regions and their relations. Now we are interested in learning the location-aware node representation to preserve the relative position of each node. Here we take the geographical information of each cluster into account to capture the topological distance of each node to its nearby clusters. That is, the clusters with geographic information will act as anchor nodes/clusters that would be used for positioning all other nodes, including the *unlabeled* nodes and topologically *isolated* nodes.

### *4.2.1. Location-aware Representation*

We utilize the topological distance to represent the relative position of a node w.r.t. all clusters. Formally, we define the concept of reachable nodes set as:

**Definition 2 (Reachable nodes set).** *Consider a node $v \in \mathcal{V}$ and a cluster $s_j$, where $v$ is not a node from $s_j$ ($v \notin s_j$). For a node $u \in s_j$, if there exists a path between $v$ and $u$ on $\mathcal{G}$, we call $u$ a reachable node for $v$, and the existence of such $u$ makes the cluster $s_j$ reachable from $v$. The set of all the nodes in $s_j$ reachable from $v$ is denoted $s_j^v$ ($s_j^v \subseteq s_j$).*

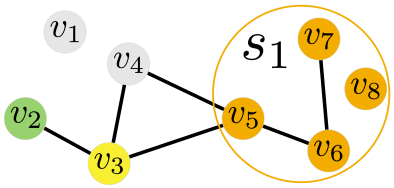



**Figure 2:** An illustration of calculating topological distance between a node and a cluster, where nodes with the same label are in the same color, and the gray nodes ($v_1, v_4$) are unlabeled.

We use a function $\mathrm{hops}(\cdot, \cdot)$ to stand the topological distance between a node and a cluster and use it to measure the impact of the cluster to the node. The intuition behind of this choice is that the shorter distance from the user to the cluster, the greater impact of this cluster on the user. The rationale is that the crowd effect from a region is more stable than the influence of individual user which is usually noisy and volatile. Note that we do not employ the geographical distance between users because there are many unlabeled users whose geographical locations are unknown (or masked for testing).

Specifically, for a node $v$, we first select the eligible nodes from set $s_j$ to form the reachable nodes set $s_j^v$, and then define the function $\mathrm{hops}(\cdot, \cdot)$ as:

$$\mathrm{hops}(v, s_j) = \begin{cases} 0 & , v \in s_j, \\ \mathbb{E}_{\forall u \in s_j^v}(\{\mathrm{hop}(v, u)\}) & , v \notin s_j \& s_j^v \neq \emptyset, \\ \infty & , v \notin s_j \& s_j^v = \emptyset, \end{cases} \tag{4}$$

where $\mathrm{hop}(v, u)$ indicates the shortest topological path from $v$ to $u$ on mention graph $\mathcal{G}$. $\mathbb{E}(\cdot)$ represents an averaging operation and the mean value can effectively avoid interference with partial extremes and outliers, which makes it more stable to obtain the topological structure information.

Figure 2 shows an example summarizing all cases in calculating $\mathrm{hops}(\cdot, \cdot)$. **Case 1:** For a node within a cluster ($v \in s_1$), $\mathrm{hops}(v, s_1) = 0$ – e.g., nodes $v_5, v_6, v_7$, and $v_8$. **Case 2:** For nodes $v \notin s_1 \& s_1^v \neq \emptyset$, we compute $\mathbb{E}(\{\mathrm{hop}(v, u)\}), \forall u \in s_1^v$. e.g., for $v_2$, $s_1^{v_2} = \{v_5, v_6, v_7\}$ and $\mathrm{hops}(v_2, s_1) = \mathbb{E}(\{\mathrm{hop}(v_2, v_5), \mathrm{hop}(v_2, v_6), \mathrm{hop}(v_2, v_7)\}) = \mathbb{E}(\{2, 3, 4\}) = 3$. **Case 3:** $v \notin s_1$ & $s_1^v = \emptyset$, e.g., unlabeled and isolated node $v_1$, $\mathrm{hops}(v_1, s_1) = \infty$. Note that the unlabeled nodes and isolated nodes have been incorporated into the model for message aggregation. Since the unlabeled nodes have no geographic information and therefore do not belong to any cluster, they can be considered as either the second (e.g., node $v_4$) or the third case (e.g., node $v_1$) when calculating their value of $\mathrm{hops}(\cdot, \cdot)$.

#### 4.2.2. Feature Merging

Above we have elaborated the measuring the impact of regions according to the topological distance from a node to the clusters – i.e., the shorter distance, the larger influence of the clusters. We normalize the impact of clusters by scaling the value of hops$(\cdot,\cdot)$, i.e., we convert hops$(\cdot,\cdot)$ into $(0,1]$ via $\frac{1}{\text{hops}(\cdot,\cdot)+1}$ to make the nearby regions more influential. Finally, we can update the features of each user by merging the regional attributes and propagating the message in the HGNN as:

$$\begin{aligned}\mathbf{x}_v^{(k)} &= f_m^{\theta_2^{(k)}}\left(\mathbf{x}_v^{(k-1)}, \mathbf{s}_j^{(k)}\right) \\ &= \sigma\left(\left[\mathbf{x}_v^{(k-1)} \middle\| \frac{\mathbf{s}_j^{(k)}}{\text{hops}(v, s_j)+1}\right]\theta_2^{(k)}\right), \forall s_j \in S,\end{aligned} \tag{5}$$

where $[\cdot\|\cdot]$ is a column-wise concatenation combining the node features with regional attributes to form a matrix. The learnable parameters $\theta_2^{(k)}$ are therefore expected to capture both regional attributes from different clusters and features of individual nodes, which will be trained with the graph neural networks. As Eq. (5) shows, the cluster attributes $\mathbf{s}_j$ are crucial for updating the user features $\mathbf{x}_v$. Thus, in order to efficiently extract the attributes, function $f_a(\cdot)$ should be an universal approximator which can be achieved by tuning the dimension ($d$) of the content features. We have:

**Proposition 2.** *The function $f_a(\cdot)$ is an universal approximator of a cluster if $d \geq \frac{M}{2^{k-1}}$, where $M$ is the maximum size of all clusters.*

*Proof.* The dimensions of $\mathbf{x}_v$ at layers $k-1$ and $k$ of the HGNN are $2^{k-1}d$ and $2^k d$, respectively. The dimension of clusters $\mathbf{s}_j$ at $k$-th layer equals to the dimension of $\mathbf{x}_v$ at $k-1$-st layer, i.e., $2^{k-1}d$. According to the universal approximation theory on sets, universal function representation for set inputs can only be achieved with a latent dimension at least the size of the maximum number of input elements $M$ [58]. Therefore, let $2^{k-1}d \geq M$, we get $d \geq \frac{M}{2^{k-1}}$. □

### 4.3. User Geolocation

We note that the user content representations $\mathbf{X}$ ($\mathbf{X} \in \mathbb{R}^{n\times d}$) are considered as node features in our HGNN. In addition to two widely used content representation techniques – TF-IDF [29] and doc2vec [36], we also leverage Bert [12] to embed the user content. We will discuss the effects of the three methods in Sec. 5.

Given the content features $\mathbf{X}$ and the mention graph $\mathcal{G}$, the output of HGNN can be represented as follows:

$$\mathbf{X}' = f_{\text{HGNN}}^{K}(\mathcal{G}, \mathbf{X}), \tag{6}$$

where $K$ is the last layer of HGNN. The updated representations $\mathbf{X}'$ ($\mathbf{X}' \in \mathbb{R}^{n\times 2^K d}$) of all the users combine multi-view features, including content features, network information of mention graph $\mathcal{G}$, as well as location information of the cluster graph $\mathcal{G}_s$. In order to map the new representations into the corresponding labels, we use a multi-layer perceptron (MLP) with softmax function as output for user geolocation:

$$\mathbf{Y}' = \text{softmax}\left(\text{MLP}(\mathbf{X}')\right), \tag{7}$$

where $\mathbf{Y}'$ ($\mathbf{Y}' \in \mathbb{R}^{n\times c}$) are the predicted results, and $y'_{ij} \in \mathbf{Y}'$ denotes the probability of $i$-th user belongs to $j$-th cluster. During training, we use cross entropy between predictions $\mathbf{Y}'$ and ground truth $\mathbf{Y}$ as loss function:

$$\ell = -\sum_{i=1}^{n}\sum_{j=1}^{c} y_{ij}\log y'_{ij}. \tag{8}$$

Parameters $\theta_1^{(k)}$ and $\theta_2^{(k)}$ ($k = \{1,\ldots,K\}$), as well as the parameters of MLP, are trained together using Adam [31] optimizer. The pseudo-codes for training HGNN is outlined in Algorithm 1.

### 4.4. Discussion

The presented HGNN with two-level GNNs can learn hierarchical structures of user mention graph – the cluster-level GNN aims at learning regional attributes that are beneficial for anchoring unlabeled nodes at the node-level GNN. Now we discuss the design choices and the model complexity.

#### 4.4.1. Why cluster-level aggregation?

The different impacts of individual user or locally connected users are not enough to locate users, because the homophily assumption (from sociological theories of connections forming) is often violated [42]. In addition, considering only the neighborhood aggregation [71] yields noise and affects stability. In HGNN, the geographic locations of clusters can be explicitly calculated, which can be used to enhance the regional influence and combine with the tweeting behavior of users in the same cluster (cf. Eq. (3)). The learned regional influence represents the crowd features which are more representative than individual or topologically neighboring user features. We note that there may exists a number of isolated users (both labeled and unlabeled) after building the mention graph $\mathcal{G}$ which, in the previous solutions, were useless and cannot be accurately located. In contrast, these node features can be leveraged in our HGNN.

#### 4.4.2. Alternative Choices

Though we choose the regions obtained by $k$-d tree [7] as the clusters – a natural choice in the context of UG task – one can alternatively use other clustering methods (e.g., $k$-means and DBSCAN [17]) to form the clusters in HGNN, i.e., the way of forming clusters in HGNN is independent of the way of region split. Notably, HGNN would degenerate to general GNN if we treat each node as an individual cluster ($S = \{v_i\}$), in which case the cluster-level GNN is exactly the same as the node-level GNN. Furthermore, other GNN models [21, 57] can be easily adapted to HGNN in an

**Algorithm 1:** Training Algorithm of HGNN.

**Input:** Tweet content $\mathbf{X}$, Mention network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, User label $\mathbf{Y}$.
**Output:** Trained HGNN.

/* Data Preprocessing */
Compute the shortest topological distance between any pair of nodes on $\mathcal{G}$;
Split all nodes into multiple clusters $S = \{s_1, \dots, s_c\}$ according to their labels;
Construct the weighted complete graph $\mathcal{G}_s$ based on all clusters and their locations;
**foreach** *node* $v \in \mathcal{V}$ **do**
  **for** *cluster id* $j = 1$ *to* $c$ **do**
    Calculate the reachable nodes set $s_j^v$ based on $s_j$ and $\mathcal{G}$;
    Compute the topological distance between $v$ and $s_j$ via Eq. (4);
  **end**
**end**
**for** *layer* $k = 1$ *to* $K$ **do**
  /* Cluster-level GNN */
  **for** *cluster id* $j = 1$ *to* $c$ **do**
    Learn $j$-th cluster's attributes $\mathbf{s}_j^{(k-1)}$ by aggregating nodes' features via Eq. (2);
    Perform convolution on $\mathcal{G}_s$ and form the regional attributes $\mathbf{S}^{(k)}$ via Eq. (3);
  **end**
  /* Node-level GNN */
  **foreach** *node* $v \in \mathcal{V}$ **do**
    Get the new representation $\mathbf{x}_v^{(k)}$ by merging all scaled regional attributes via Eq. (5);
  **end**
**end**
/* Geolocation Predictor */
Get the new representation of all user $\mathbf{X}'$;
Compute predictions $\mathbf{Y}'$ with $\mathbf{X}'$ via Eq. (7);
Calculate loss between $\mathbf{Y}$ and $\mathbf{Y}'$ via Eq. (8);
Update model parameters with Adam.

end-to-end manner. Therefore, HGNN can be considered as a general GNN framework independent of specific message passing while enabling hierarchical and location-aware graph learning.

#### *4.4.3. Complexity*

In HGNN, the number of clusters equals to the number of labels $c$ which is a fixed number in a given dataset. For each node, there are at most $n$ features of clusters to merge, resulting in $\mathcal{O}(nc)$ complexity. Therefore, HGNN has the same computational complexity as the specific GNN model that is applied in each level. An additional overhead is to calculate shortest path between any pairs of nodes in $\mathcal{G}$, which only needs to be performed once since the shortest paths are fixed for a static mention graph.

### 4.5. Interpretability

Most of the graph neural network-based methods including those user geolocation approaches [15, 42, 50] model the process of learning and predicting user geolocations as "black-box" and therefore are limited in terms of explaining the geolocation results. However, it is important for downstream decision making to understand how the model learns the data and why the prediction are generated. Whether influence functions can be applied to GNN-based models remained unclear. We take a step towards bridging this gap by tracing the geolocation results from the GNN-based models back to the important nodes in the mention network.

The empirical influence function [23] is a measure of the dependence of the model on one of the samples, which has been widely used for inferring data samples in computer vision [33], graph-structural learning [72], etc. In particular, we utilize the influence function to estimate the importance of each training user $e$ on a particular user geolocation result $e_{\text{test}}$ during testing. We achieve this goal by tracing the geolocation results output by specified model and back to the nodes in the mention graph that are important for predicting, both positively and negatively.

Formally, assume we have $m$ training samples $e_1, \dots, e_m$, which are the nodes (users) in the mention graph $\mathcal{G}$. Here each $e_i = (\mathbf{x}_i, \mathbf{y}_i)$ consists of the attribute vector $\mathbf{x}_i$ and the corresponding label $\mathbf{y}_i$. According to the robust statistics [23, 33], removing one data $e$ from the training set leads to a change of the optimal model parameters from $\theta^*$ to $\theta^*_{-e}$, where $\theta^*$ and $\theta^*_{-e}$ are optimal parameter sets before and after removing the data point $e$, respectively. The posterior one $\theta^*_{-e}$ is estimated as:

$$\theta^*_{-e} \overset{\text{def}}{=} \arg\min_{\theta\in\Theta} \sum_{e_i \neq e} \ell(e_i, \theta), \tag{9}$$

where $\ell(e_i, \theta)$ denotes the loss of a particular node $e_i$. Koh et al. [33] propose to efficiently approximate the influence of removing a data $e$ by computing the change of parameters, which is similar to upweight $e$ with a small value $\gamma$. Then, the updated parameters can be estimated as:

$$\theta^*_{\gamma,e} \overset{\text{def}}{=} \arg\min_{\theta\in\Theta} \frac{1}{m} \sum_{i=1}^{m} \ell(e_i, \theta) + \gamma \ell(e, \theta), \tag{10}$$

where the effect of upweighting $e$ on $\theta^*$ is inferred as:

$$\psi_{\text{up},\theta^*}(e) \overset{\text{def}}{=} \left.\frac{\partial \theta^*_{\gamma,e}}{\partial \gamma}\right|_{\gamma=0} = -\mathbf{H}^{-1}_{\theta^*} \bigtriangledown_\theta \ell(e, \theta^*), \tag{11}$$

where $\mathbf{H}_{\theta^*}$ is the Hessian matrix. This formula implies that the effect of removing $e$ equals to upweight it by $\gamma = -\frac{1}{m}$. In other words, it allows us to approximate the model change as $\theta^*_{-e} - \theta^* \approx -\frac{1}{m}\psi_{\text{up},\theta^*}(e)$ without retraining the model. Its influence a testing node $e_{\text{test}}$ is thereby computed:

$$\psi_{\text{up,loss}}(e, e_{\text{test}}) \overset{\text{def}}{=} \left.\frac{\partial \ell(e_{\text{test}}, \theta^*_{\gamma,e})}{\partial \gamma}\right|_{\gamma=0}$$

**Table 2**
Statistics of datasets.

| Dataset | $\lvert\mathcal{V}_{train}\rvert$ / $\lvert\mathcal{V}_{val}\rvert$ / $\lvert\mathcal{V}_{test}\rvert$ | $\lvert\mathcal{E}\rvert$ | $c$ |
|---|---|---|---|
| GeoText | 5,685 / 1,895 / 1,895 | 77,155 | 129 |
| Twitter-US | 429,200 / 10,000 / 10,000 | 18,498,702 | 256 |
| Twitter-World | 1,366,766 / 10,000 / 10,000 | 1,001,181 | 930 |

$$= \bigtriangledown_\theta \ell(e_{\text{test}}, \theta^*)^\intercal \frac{\partial \theta^*_{\gamma,e}}{\partial \gamma}\Big|_{\gamma=0}$$
$$= -\bigtriangledown_\theta \ell(e_{\text{test}}, \theta^*)^\intercal \mathbf{H}^{-1}_{\theta^*} \bigtriangledown_\theta \ell(e, \theta^*). \quad (12)$$

In real implementation, one can use implicit Hessian-vector products to accelerate the approximation of $\pounds_{\text{test}} \overset{\text{def}}{=} \mathbf{H}^{-1}_{\theta^*} \bigtriangledown_\theta \ell(e_{\text{test}}, \theta^*)$. It means that Eq. (12) can be reformulated as $\psi_{\text{up,loss}}(e, e_{\text{test}}) = -\pounds_{\text{test}} \bigtriangledown_\theta \ell(e, \theta^*)$. Since the matrix $\mathbf{H}_{\theta^*}$ is assumed to be positive semi-definite, we have:

$$\pounds_{\text{test}} \equiv \arg\min_\eta \left\{ \frac{1}{2} \eta^\intercal \mathbf{H}_{\theta^*} \eta - \bigtriangledown_\theta \ell(e_{\text{test}}, \theta^*)^\intercal \eta \right\}, \quad (13)$$

where the exact solution $\eta$ is achieved with conjugate gradients that only needs to estimate$\mathbf{H}_{\theta^*}\eta$. That is, it becomes unnecessary to calculate $\mathbf{H}^{-1}_{\theta^*}$ that is computational expensive. We refer the details of influence function and its applications to [23, 33, 72]. In next section, we will present the empirical results of explaining the model behavior and information fusion results.

## 5. Experiments

In this section, we present the results of the evaluation of our proposed methods on real-world datasets.

We first describe the experimental settings, including datasets, baselines, evaluation protocols and parameter settings. Next, we follow with performance comparison and model interpretability. Specifically, our extensive experiments aimed to answer the following research questions:

- (**RQ1** How does HGNN perform on user geolocation compared with the state-of-the-art baselines?
- (**RQ2** What is the effect of each component in the proposed model?
- (**RQ3** Can HGNN explain the model behavior on information fusion and location prediction?

### 5.1. Experimental Settings

#### *5.1.1. Datasets*

We conduct all the experiments on three real-world Twitter datasets which have been widely used for evaluating user geolocation models:

- **GeoText** [16] is a Twitter dataset consisting of 9.5K users from 49 states in the U.S., which is originally compiled by the authors in [16].
- **Twitter-US** [54] is a larger dataset consisting of 449K users from the U.S. This dataset is also referred to as UT-Geo2011 in some papers [15, 54].
- **Twitter-World** [24] is a much larger dataset released by the authors of [24] and had been rebuilt by the authors of [50]. This dataset consists 1.3M users from different countries in the world, and the primary locations of users are mapped to the geographic center of the city from where the majority of their tweets are posted.

The statistics of the datasets are presented in Table 2. The distributions of clusters after dividing the training users are shown in Figure 3, where each cluster is outlined using minimum convex polygon algorithm [18].

#### *5.1.2. Baselines*

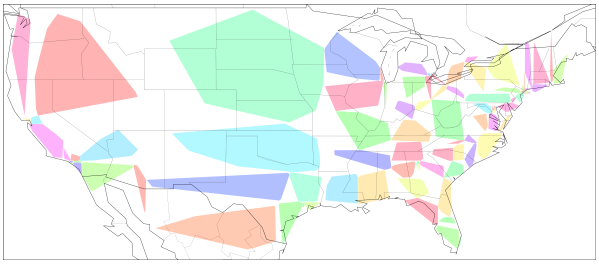
We compare HGNN with two categories of baselines. For fair comparison, we remove the metadata (e.g., timezone information and geographical description) in this line of work.

The first category consists of *text-based* methods, including:

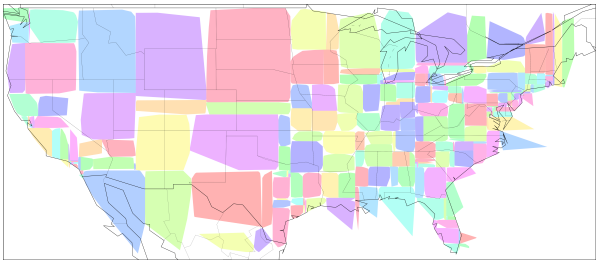
- *HierLR* [60] employs logistic regression (LR) for user classification.
- *MLP4Geo* [49] utilizes dialectal terms to improve the prediction performance and a MLP network to predict the locations.
- *DocSim* [54] uses the similarity (KL divergence) of the tweet contents for user geolocation.
- *MixNet* [47] applies mixture density network for embedding coordinates and MLP for classification.

The second category considers user mention graph for *multi-view* information fusion and user geolocation, including:

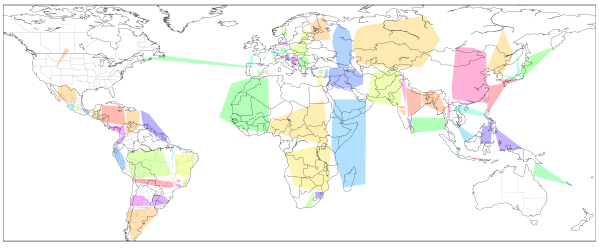
- *MADCEL* [48] combines the text and network information and uses LR for prediction.
- *MENET* [15] concatenates the features from textual information (*TF-IDF* and *doc2vec*) and embeds user network with *node2vec*.
- *GeoAtt* [42] models the textual context with RNN and attention mechanism.
- *DCCA* [50] is a multiview geolocation model using Twitter text and network information and measures the canonical correlations among users for location prediction.



(a) GeoText.



(b) Twitter-US.



(c) Twitter-World.

**Figure 3:** Clusters distribution of training users in three datasets, produced by $k$-d tree and minimum convex polygon algorithm. The number of polygons equals to the number of clusters (labels) for corresponding dataset. Due to the limited map resolution, many areas (polygons) are too small to see clearly in Twitter-World.

- *GCN4Geo* [50] is a GCN-based multi-view UG model.
- *KB-emb* [43] is UG method based on entity linking and the embedding of knowledge-base.
- *GaussMix* [5] utilizes a series of Gaussian mixture models to exploit both text and network features according to geographic information.

#### 5.1.3. Evaluation protocols

We evaluate all approaches using following three metrics following previous works: (1) *Mean* is the averaged errors between the predicted cluster centers and the ground-truth geolocations. (2) *Median* reports the median value of all predicted results (3) *Acc@100* measures the accuracy of the classification, i.e., if the distance between the predicted center and ground-truth is within 100 miles, the result will be considered as a correct prediction. We respectfully note that, while this may be perceived as a rather coarse value, it is widely used in the literature (e.g., [42, 50]) and we used it for comparison with the baselines.

#### 5.1.4. Parameter settings

We apply two-layer convolutions on both cluster- and node-level aggregations in HGNN, i.e., $K = 2$. We adopt the Adam [31] optimizer to train the model with learning rate in the range of $[0.001, 0.1]$ and weight decay in the set of $\{5e-9, 5e-8, 5e-7\}$. We also add dropout to the hidden units at each layer of MLP to stabilize the training of our model. Besides, the dimension of cluster attributes is equal to the dimension of user features at the same layer. When proceeding to the next layer, the dimension of user features will be doubled by extending the cluster attributes. Specifically, the dimension of initial user features $\mathbf{x}_v$ (i.e., content features) is $d$. As for the learnable weight matrices, the size of $\theta_1^{(k)}$ (used for cluster-level convolution) is $(2^{k-1}d, 2^{k-1}d)$, and the size of $\theta_2^{(k)}$ (used for node-level learning) is $(2^k d, 2^k d)$. Consequently, the size of parameters for MLP output is $(2^K d, c)$. We performed a grid search for the parameter $d$ and set $d$ to 128, 256 and 256 for the GeoText, Twitter-US, and Twitter-World, respectively. In addition, early stopping strategy is adopted when training the HGNN if the validation loss does not decrease for 30 consecutive epochs.

### 5.2. Performance Comparison (RQ1)

The overall performance of all methods across three datasets is presented in Table 3, from which we make the following observations.

First, the HGNN model consistently outperforms the baselines on all metrics, demonstrating the effect of addressing the user geolocation problem with the proposed location-aware hierarchical GNNs. In particular, the performance gains of HGNN over the best baseline method in terms of Mean, Median, and Acc@100 metrics are 6.78%, 17.86%, and 5.00% on GeoText, 12.62%, 15.91%, and 4.84% on Twitter-US, and 9.47%, 7.41%, and 3.70% on Twitter-World, respectively.

Next, we observe that solely relying on tweet content (e.g., HierLR, MLP4Geo, DocSim, and MixNet) is not enough for accurate user geolocation, which usually exhibits too high prediction bias. This is intuitive since neither indicative words nor topic-based language models can filter out noisy signals with only user tweeting contentand, in a sense, also verifies that users often write tweets and engage in retweeting articles in a very casual manner [71], which makes word-centric and location-centric methods inaccurate.

Third, the performance of multi-view information fusion models, including MENET, GeoAtt, DCCA, KB-emb, GaussMix, and GCN4Geo, are very similar if both text and network features are used for user geolocation. In many online social networks, mention and following are two essential user interactions, which have been widely used for modeling close relationships or similar attributes. As shown in the results, this kind of information is a strong indicator for user geolocation, which should be carefully incorporated into the model. Nevertheless, previous models rely on one- or two-hop friendship or mentioning to infer the geolocation

**Table 3**
Performance comparison.

| Method | GeoText | | | Twitter-US | | | Twitter-World | | |
|---|---|---|---|---|---|---|---|---|---|
| | *Mean* | *Median* | *Acc@100* | *Mean* | *Median* | *Acc@100* | *Mean* | *Median* | *Acc@100* |
| HierLR | 518 | 250 | 41 | 437 | 106 | 49 | 1065 | 304 | 32 |
| MLP4Geo | 524 | 242 | 38 | 344 | 75 | 54 | 904 | 258 | 34 |
| DocSim | 557 | 268 | 35 | 534 | 288 | 34 | – | – | – |
| MixNet | 537 | 256 | 39 | 407 | 134 | 42 | – | – | – |
| MADCEL | 361 | 35 | 59 | 329 | 48 | 60 | 872 | 69 | 53 |
| MENET | 400 | 78 | 55 | 327 | 58 | 52 | 802 | 78 | 52 |
| GeoAtt | 380 | 50 | 57 | 339 | 57 | 54 | 785 | 133 | 50 |
| DCCA | 390 | 49 | 56 | 321 | 56 | 58 | 1302 | 567 | 21 |
| GCN4Geo | 339 | 28 | 60 | 301 | 44 | 62 | 702 | 67 | 54 |
| KB-emb | 485 | 211 | 43 | 373 | 122 | 45 | – | – | – |
| GausMix | 368 | 43 | 56 | 336 | 55 | 56 | 844 | 124 | 51 |
| **HGNN** | **316** | **23** | **63** | **263** | **37** | **65** | **636** | **62** | **56** |

**Table 4**
Comparison of different GNN models.

| GNN | GeoText | | | Twitter-US | | | Twitter-World | | |
|---|---|---|---|---|---|---|---|---|---|
| | *Mean* | *Median* | *Acc@100* | *Mean* | *Median* | *Acc@100* | *Mean* | *Median* | *Acc@100* |
| GCN | 336 | 27 | 61 | 300 | 43 | 62 | 719 | 113 | 54 |
| GAT | 358 | 34 | 59 | 291 | 42 | 63 | 831 | 144 | 50 |
| SGC | 347 | 28 | 61 | 295 | 43 | 62 | 722 | 71 | 54 |
| GraphPool | 418 | 110 | 54 | 349 | 70 | 51 | 923 | 178 | 48 |
| PGNN | 404 | 102 | 55 | 365 | 88 | 50 | 888 | 160 | 49 |

of users, which can easily suffer from data sparsity problem (e.g., the users with fewer social interactions) and inaccurate prediction (e.g., the friends of the current user have unknown home locations).

Recently, graph neural networks have emerged as *de facto* tools for network information fusion in graph-structured data. Notably, GCN4Geo is a GNN-based UG method and relies on *flat* GCN [32] for node feature learning and fusion. However, GCN4Geo only leverages the explicit user-mentioned graph and therefore exhibits inferior performance compared to our HGNN that uses hierarchical knowledge distillation. In addition, our method explicitly captures the features of isolated nodes and their relations to the clusters. That is, the rich information of the unlabeled data has been modeled for knowledge fusion and improving the prediction performance in HGNN.

## 5.3. Ablation Study (RQ2)

We now discuss the ablation study which we conducted to explore the effect of each component in HGNN. Specifically, we investigate two important information fusion components in our model: the structural learning and linguistic representation.

### *5.3.1. Which GNN is better?*

Since HGNN is a general framework for hierarchical network structure learning, any GNN model can be adapted into HGNN. Here we compare several representative GNN models using the publicly released implementations. As shown in Table 4, their performance are very similar – while GCN slightly outperforms others on GeoText and Twitter-World, GAT performs better on Twitter-US. This result suggests that the performance of HGNN lies in the hierarchical structure learning and leveraging unlabeled and isolated nodes rather than specific graph neural networks.

We also compare HGNN with a hierarchical GNN framework – GraphPool [67] and a position-aware GNN model – PGNN [68] by adapting them to solve the UG task. Table 4 reports the results, which show that both of them are not competitive. GraphPool is originally proposed for graph classification, whose node clustering in each layer is based on nodes' topological proximity but fails to leverage the features of unlabeled nodes. PGNN, on the other hand, learns the topological relative position of nodes – rather than geographical relative location of nodes in HGNN – and, consequently, does not perform well on the UG task. Meanwhile, both GraphPool and PGNN cannot handle the prevalence of isolated nodes in the user mention graph.

### *5.3.2. How to represent user content?*

Though how to learn a better text representation is not the main scope of this work, we provide a new choice in addition to TF-IDF and doc2vec by tuning a light Bert [35] for text representation. Table 5 shows that Bert-style content

**Table 5**
HGNN performance with various content representations.

| Model | GeoText | | | Twitter-US | | | Twitter-World | | |
|---|---|---|---|---|---|---|---|---|---|
| | *Mean* | *Median* | *Acc@100* | *Mean* | *Median* | *Acc@100* | *Mean* | *Median* | *Acc@100* |
| HGNN-TF | 329 | 25 | 61 | 304 | 45 | 62 | 747 | 74 | 53 |
| HGNN-d2v | 365 | 36 | 59 | **263** | **37** | **65** | 688 | 65 | 54 |
| HGNN-Bert | **316** | **23** | **63** | 289 | 43 | 63 | **636** | **62** | **56** |

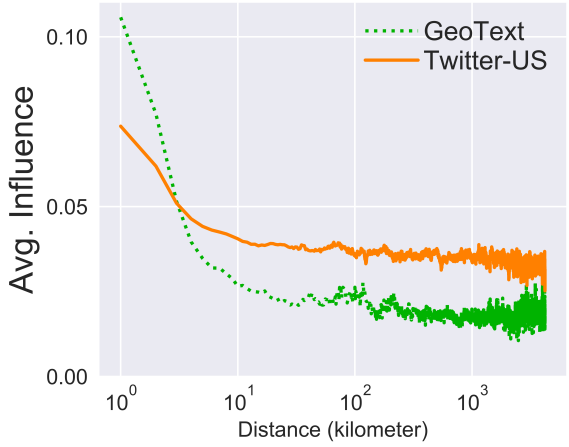
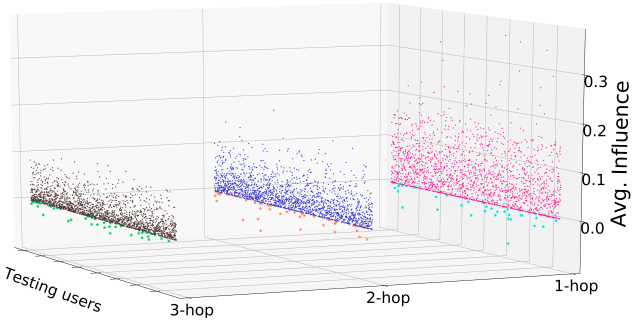
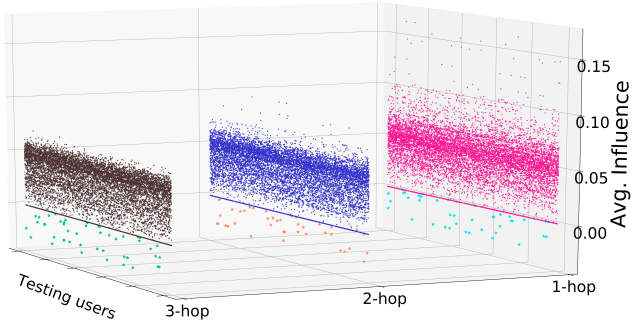



**Figure 4:** The average influence varies with distance (KM) between training and testing samples.




(a) GeoText.




(b) Twitter-US.

**Figure 5:** The influence of $\lambda$-hop ($\lambda = 1, 2, 3$) neighbors. Positive and negative nodes are marked as dots (•) and pentagons (★), respectively.

representation works well on sparse datasets (i.e., GeoText and Twitter-World) but degenerates on dense datasets (i.e., Twitter-US). This is because text similarity is more important when the number of nodes is small. However, user interactions become a dominant factor if sufficient mention is available.

## 5.4. Fusion and Prediction Interpretability (RQ3)

To better understand the model behavior, we leverage influence functions [33] to estimate the effect of individual training sample and interpret the prediction results made by our HGNN. The main idea is to estimate the effect that removing of a particular training node has on the model's final prediction. This is achieved by tracing the geolocation results back to the nodes/clusters in the mention graph that are important for prediction, both positively and negatively. Next, we turn to interpret the results made by HGNN on GeoText and Twitter-US from three aspects.

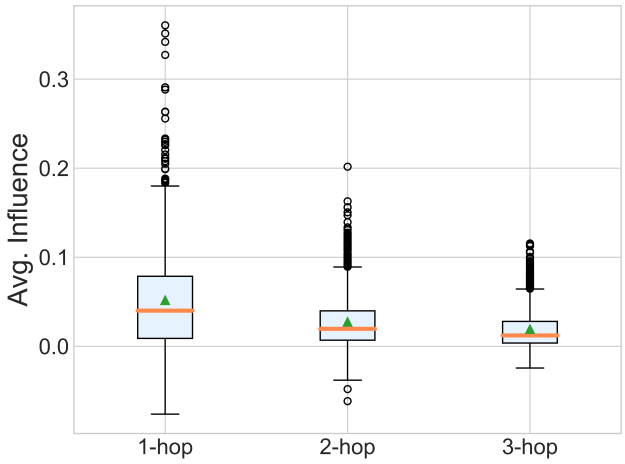
### *5.4.1. Influence of geographic distance*

Intuitively, the geographically closed neighbors for a particular user play more important roles in locating this user, which is also the primary motivation of many previous works [5, 15, 42, 43, 48, 50]. However, there are few prior studies on quantifying such impact. In addition, we used the influence function to estimate each training samples' influence, which does not use any information w.r.t. user geographical locations. Therefore, we would like to discover the underlying relations between the two aspects.

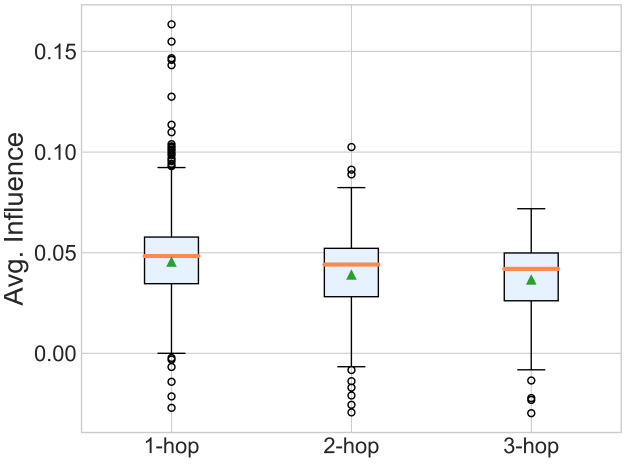
In Figure 4, we compute the impact of all training samples on each testing user with the influence function and average the results on the two datasets. As illustrated, the estimated influence is strongly correlated with the distance, especially when users are within 10 KM, as the users in both datasets are from the U.S. Besides, we can also observe that for those who are far away, their influence is trivial. This result not only verifies our motivation to investigate the training samples but also bridges the gap between influence estimates and user geographical locations.

### *5.4.2. Influence of $\lambda$-hop neighbors*

As shown in Figure 5, we quantify the influence of $\lambda$-hop neighbors on each testing user in GeoText and Twitter-US, e.g., each pink point is the averaged influence of all the 1-hop users on the testing user, and so on. In addition, we




(a) GeoText.




(b) Twitter-US.

**Figure 6:** The box plot of the influence distribution of corresponding neighbors. The triangles (▲) and orange lines in the boxes represent the mean and median values, respectively. The circles (∘) denote the outliers with extreme influence.

also draw a boxplot for each hop of nodes in Figure 6, where the green triangle and orange line in the box denote the mean and median of the corresponding set of nodes, respectively.

As Figure 5 illustrates, immediate neighbors generally contribute more (i.e., positive influence) on the testing samples. By contrast, the positive influence of topologically further nodes (e.g., 2-hop and 3-hop neighbors) gradually decrease. However, we can also observe that in some cases the immediate neighbors are noisy signals (or have negative influences) on user geolocation, as those pentagons whose influence value is less than 0. This phenomenon indicates that aggregating features of neighboring nodes in a GNN manner may confront the problem of noisy and unstable information fusion and, consequently, may yield unsatisfactory prediction results.

Furthermore, we also have some insights from Figure 6. First, both the mean and median values decrease as increasing the number of hops. This result meets our expectation, i.e., the closer a training sample to the testing user, the more positive impact. As for the outliers, the number is gradually decreasing as the hops increase. This result also raises an open problem for estimating the influence of in-network nodes, i.e., the closer nodes (e.g., 1-hop neighbors) are mixed with more noise even they have more significant influence. Therefore, it is an interesting topic to identify the anomaly samples in user geolocation, e.g., the nodes marked as ∘. Finally, the influence fluctuation of IQR (Inter-Quartile Range, i.e., the height of the box) in Geotext is larger than that in Twitter-US. This phenomenon happens because GeoText is a sparser mention graph compared to Twitter-US. Therefore, it is too difficult for the HGNN model to sample sufficient users from immediate neighbors for stabilizing the influence estimation.

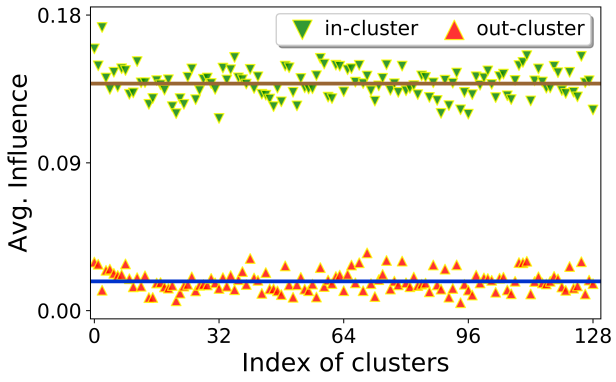
#### 5.4.3. *Influence of regions*

Since our model relies on the cluster effect to geolocate the users, we now investigate the influence of regions when aggregating knowledge in the HGNN model. An important assumption in the previous method is that physically close users usually have an interactive relationship in the network topology, or vice versa. To confirm this hypothesis and quantify such relations, for each cluster, we regard the training samples in a region as *in*-cluster nodes, and the training samples outside it as *out*-cluster nodes. Through computing the average influence of *in* -cluster and *out* -cluster nodes for each testing user, we can gauge the impact of each cluster.

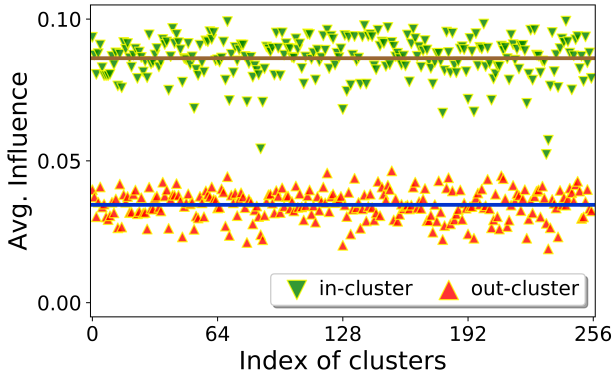
As illustrated in Figure 7, *in*-cluster nodes play dominant role in user geolocation. For example, the nodes in the region have more significantly positive influences than those out of the region, which are applicable to both datasets. This result provides intuitive explanations on the performance of HGNN and justifies our motivation – i.e., using clustering effect to provide robust signals for user geolocation, rather than only relying on individual users as in previous works. On the other hand, the influence variations on GeoText are smaller than on Twitter-US, caused by the differences of region split on two datasets. That is, there are more clusters (with smaller areas) in Twitter-US (cf. Figure 3), resulting in larger variances for individual *in*- and *out*-cluster – the densely region partition would have more regions that could more easily confuse the model in terms of accurate geolocation compared to sparse and distinct region partition. In other words, the more clusters, the greater the variance when estimating the influence of a single cluster.

#### 5.4.4. *Visualization of the latent space*

In order to better understand the benefits of the hierarchical architecture in the proposed model, we plot the user embeddings learned by different approaches. Towards that, we randomly select four regions from GeoText and project the latent space of users from those regions into the 2D space using t-SNE [40]. Figure 8 shows the latent space of MENET, GCN4Geo, GausMix, and our HGNN, where nodes with the same color are from the same regions. As

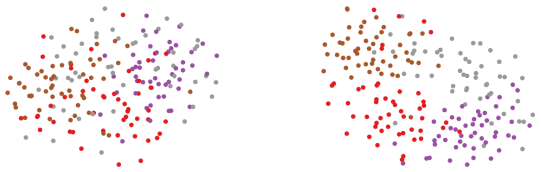



(a) GeoText.

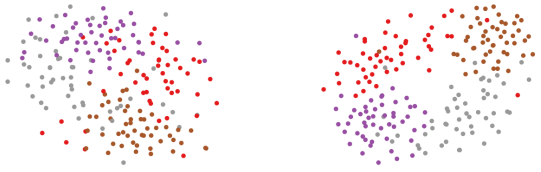



(b) Twitter-US.

**Figure 7:** The average cluster influence of both *in*-cluster (upper) and *out*-cluster (bottom) samples. The means of *in*-cluster and *out*-cluster influence are shown by the brown and blue lines, respectively.



(a) MENET. (b) GCN4Geo.



(c) GausMix. (d) HGNN.

**Figure 8:** Visualization of the learned latent space. We randomly select four regions from GeoText and plot the learned user embeddings using t-SNE.

a direct feature concatenation method, MENET cannot discriminate the user embeddings. GausMix and GCN4Geo can generate more distinct user embeddings, although some entangling users may not be correctly identified. In comparison, we can observe a more obvious clustering effect generated by HGNN, which provides a more intuitive explanation of the superiority of the hierarchical aggregations in the proposed architecture.

## 6. Conclusions and Future Work

We introduced a novel approach, Hierarchical Graph Neural Networks (HGNN) for UG prediction, that leverages robust signals from geographically close crowds rather than individuals (as done in previous works). HGNN couples the topological structure and the physical locations of users with a relation mechanism learning, which allows exploiting both unlabeled nodes and isolated nodes in the data. Empirical results demonstrated that HGNN not only achieves the state-of-the-art geolocation performance, but also enables interpretability of prediction results. We also presented a framework for explaining the GNN-based models by extending the influence function to estimate the effect of samples in graph data, which provides explanations of information fusion and allows us to quantify the influence of both individual nodes and the clusters during training our model. In the future, we plan to investigate the impact of different space-partitioning and spatial clustering methods. Another part of our future work is to explore how to utilize the explainable results for improving the collaborative user geolocation performance.

## Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant No. 62072077) and the NSF SWIFT grant 2030249.

## References

[1] Amitay, E., Har'El, N., Sivan, R., Soffer, A., 2004. Web-a-where: geotagging web content, in: Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 273–280.

[2] Arrieta, A.B., Rodríguez, N.D., Ser, J.D., Bennetot, A., Tabik, S., Barbado, A., García, S., Gil-Lopez, S., Molina, D., Benjamins, R., Chatila, R., Herrera, F., 2020. Explainable artificial intelligence (XAI): concepts, taxonomies, opportunities and challenges toward responsible AI. Information Fusion 58, 82–115. doi:10.1016/j.inffus.2019.12.012.

[3] Backstrom, L., Sun, E., Marlow, C., 2010. Find me if you can: improving geographical prediction with social and spatial proximity, in: Proceedings of the International Conference on World Wide Web Conferences (WWW), pp. 61–70.

[4] Bahdanau, D., Cho, K., Bengio, Y., 2015. Neural machine translation by jointly learning to align and translate, in: Proceedings of the International Conference on Learning Representations (ICLR).

[5] Bakerman, J., Pazdernik, K., Wilson, A., Fairchild, G., Bahran, R., 2018. Twitter geolocation: A hybrid approach. ACM Transactions on Knowledge Discovery from Data (TKDD) 12, 34. doi:10.1145/3178112.

[6] Baldassarre, F., Azizpour, H., 2019. Explainability techniques for

graph convolutional networks, in: ICML Workshops on Learning and Reasoning with Graph-Structured Representations.

[7] Bentley, J.L., 1975. Multidimensional binary search trees used for associative searching. Communications of the ACM 18, 509–517. doi:10.1145/361002.361007.

[8] Cheng, Z., Caverlee, J., Lee, K., 2010. You are where you tweet: a content-based approach to geo-locating twitter users, in: Proceedings of the International Conference on Information and Knowledge Management (CIKM), ACM. pp. 759–768.

[9] Chong, W.H., Lim, E.P., 2019. Fine-grained geolocation of tweets in temporal proximity. ACM Transactions on Information and Systems (TOIS) 37, 1–33. doi:10.1145/3291059.

[10] Chung, J., Gulcehre, C., Cho, K., Bengio, Y., 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555 .

[11] Davis Jr, C.A., Pappa, G.L., de Oliveira, D.R.R., de L. Arcanjo, F., 2011. Inferring the location of twitter messages based on user relationships. Transactions GIS 15, 735–751. doi:10.1111/j.1467-9671.2011.01297.x.

[12] Devlin, J., Chang, M.W., Lee, K., Toutanova, K., 2019. Bert: Pre-training of deep bidirectional transformers for language understanding, in: Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT), pp. 4171–4186.

[13] Diez-Olivan, A., Ser, J.D., Galar, D., Sierra, B., 2019. Data fusion and machine learning for industrial prognosis: Trends and perspectives towards industry 4.0. Information Fusion 50, 92–111. doi:10.1016/j.inffus.2018.10.005.

[14] Ding, R., Wang, X., Shang, K., Herrera, F., 2019. Social network analysis-based conflict relationship investigation and conflict degree-based consensus reaching process for large scale decision making using sparse representation. Information Fusion 50, 251–272. doi:10.1016/j.inffus.2019.02.004.

[15] Do, T.H., Nguyen, D.M., Tsiligianni, E., Cornelis, B., Deligiannis, N., 2017. Multiview deep learning for predicting twitter users' location. arXiv preprint arXiv:1712.08091 .

[16] Eisenstein, J., O'Connor, B., Smith, N.A., Xing, E.P., 2010. A latent variable model for geographic lexical variation, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1277–1287.

[17] Ester, M., Kriegel, H.P., Sander, J., Xu, X., 1996. A density-based algorithm for discovering clusters in large spatial databases with noise, in: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD), pp. 226–231.

[18] Graham, R., 1972. An efficient algorithm for determining the convex hull of a finite planar set. Information Processing Letters 1, 132–133. doi:10.1016/0020-0190(72)90045-2.

[19] Grover, A., Leskovec, J., 2016. node2vec: Scalable feature learning for networks, in: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (SIGKDD), pp. 855–864.

[20] Guiñazú, M.F., Cortés, V., Ibáñez, C.F., Velásquez, J.D., 2020. Employing online social networks in precision-medicine approach using information fusion predictive model to improve substance use surveillance: A lesson from twitter and marijuana consumption. Information Fusion 55, 150–163. doi:10.1016/j.inffus.2019.08.006.

[21] Hamilton, W., Ying, Z., Leskovec, J., 2017. Inductive representation learning on large graphs, in: Advances in Neural Information Processing Systems, pp. 1024–1034.

[22] Hamouni, P., Khazaei, T., Amjadian, E., 2019. Tf-mf: Improving multiview representation for twitter user geolocation prediction, in: Proceedings of the International Conference on Advances in Social Networks Analysis and Mining (ASONAM), pp. 543–545.

[23] Hampel, F.R., Ronchetti, E.M., Rousseeuw, P.J., Stahel, W.A., 2011. Robust statistics: the approach based on influence functions. volume 196. John Wiley & Sons.

[24] Han, B., Cook, P., Baldwin, T., 2012. Geolocation prediction in social media data by finding location indicative words, in: Proceedings of the International Conference on Computational Linguistics (COLING), pp. 1045–1062.

[25] Han, B., Cook, P., Baldwin, T., 2014. Text-based twitter user geolocation prediction. Journal of AI Research (JAIR) 49, 451–500. doi:10.1613/jair.4200.

[26] Holzinger, A., Biemann, C., Pattichis, C.S., Kell, D.B., 2017. What do we need to build explainable AI systems for the medical domain? arXiv preprint arXiv:1712.09923 .

[27] Holzinger, A., Malle, B., Saranti, A., Pfeifer, B., 2021. Towards multi-modal causability with graph neural networks enabling information fusion for explainable AI. Information Fusion 71, 28–37.

[28] Huang, Q., Yamada, M., Tian, Y., Singh, D., Yin, D., Chang, Y., 2020. Graphlime: Local interpretable model explanations for graph neural networks. arXiv preprint arXiv:2001.06216 .

[29] Jones, K.S., 1972. A statistical interpretation of term specificity and its application in retrieval. Journal of Documentation 28, 11–21. doi:10.1108/00220410410560573.

[30] Jurgens, D., 2013. That's what friends are for: Inferring location in online social media platforms based on social relationships, in: Proceedings of the International Conference on Weblogs and Social Media (ICWSM).

[31] Kingma, D.P., Ba, J., 2015. Adam: A method for stochastic optimization, in: Proceedings of the International Conference on Learning Representations (ICLR).

[32] Kipf, T.N., Welling, M., 2017. Semi-supervised classification with graph convolutional networks, in: Proceedings of the International Conference on Learning Representations (ICLR).

[33] Koh, P.W., Liang, P., 2017. Understanding black-box predictions via influence functions, in: Proceedings of the International Conference on Machine Learning (ICML), pp. 1885–1894.

[34] Kong, L., Liu, Z., Huang, Y., 2014. Spot: Locating social media users based on social network context. Very Large Data Bases Conferences (VLDB) 7, 1681–1684. doi:10.14778/2733004.2733060.

[35] Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., Soricut, R., 2020. Albert: A lite bert for self-supervised learning of language representations, in: Proceedings of the International Conference on Learning Representations (ICLR).

[36] Le, Q., Mikolov, T., 2014. Distributed representations of sentences and documents, in: Proceedings of the International Conference on Machine Learning (ICML), pp. 1188–1196.

[37] Li, P., Lu, H., Kanhabua, N., Zhao, S., Pan, G., 2018. Location inference for non-geotagged tweets in user timelines. IEEE Transactions on Knowledge and Data Engineering (TKDE) 31, 1150–1165. doi:10.1109/TKDE.2018.2852764.

[38] Li, R., Wang, S., Deng, H., Wang, R., Chang, K.C.C., 2012. Towards social user profiling: unified and discriminative influence model for inferring home locations, in: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD), pp. 1023–1031.

[39] Longo, L., Goebel, R., Lécué, F., Kieseberg, P., Holzinger, A., 2020. Explainable artificial intelligence: Concepts, applications, research challenges and visions, in: Proceedings of the International Cross-Domain Conference for Machine Learning and Knowledge Extraction (CD-MAKE), pp. 1–16.

[40] Maaten, L.v.d., Hinton, G., 2008. Visualizing data using t-sne. Journal of machine Learning research (JMLR) 9, 2579–2605.

[41] McGee, J., Caverlee, J., Cheng, Z., 2013. Location prediction in social media based on tie strength, in: Proceedings of the International Conference on Information and Knowledge Management (CIKM), pp. 459–468.

[42] Miura, Y., Taniguchi, M., Taniguchi, T., Ohkuma, T., 2017. Unifying text, metadata, and user network representations with a neural network for geolocation prediction, in: Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL), pp. 1260–1272.

[43] Miyazaki, T., Rahimi, A., Cohn, T., Baldwin, T., 2018. Twitter geolocation using knowledge-based methods, in: Proceedings of the Work-

shop on Noisy User-generated Text (NUT@EMNLP), pp. 7–16.

[44] Pati, P., Jaume, G., Fernandes, L.A., Foncubierta-Rodríguez, A., Feroce, F., Anniciello, A.M., Scognamiglio, G., Brancati, N., Riccio, D., Di Bonito, M., et al., 2020. Hact-net: A hierarchical cell-to-tissue graph neural network for histopathological image classification, in: Uncertainty for Safe Utilization of Machine Learning in Medical Imaging, and Graphs in Biomedical Image Analysis. Springer, pp. 208–219.

[45] Poese, I., Uhlig, S., Kaafar, M.A., Donnet, B., Gueye, B., 2011. Ip geolocation databases: Unreliable? Computer Communication Review 41, 53–56. doi:10.1145/1971162.1971171.

[46] Qiu, J., Dong, Y., Ma, H., Li, J., Wang, K., Tang, J., 2018. Network embedding as matrix factorization: Unifying deepwalk, line, pte, and node2vec, in: Proceedings of the International Conference on Web Search and Data Mining (WSDM), pp. 459–467.

[47] Rahimi, A., Baldwin, T., Cohn, T., 2017a. Continuous representation of location for geolocation and lexical dialectology using mixture density networks, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 167–176.

[48] Rahimi, A., Cohn, T., Baldwin, T., 2015a. Twitter user geolocation using a unified text and network prediction model, in: Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL), pp. 630–636.

[49] Rahimi, A., Cohn, T., Baldwin, T., 2017b. A neural model for user geolocation and lexical dialectology, in: Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL), pp. 209–216.

[50] Rahimi, A., Cohn, T., Baldwin, T., 2018. Semi-supervised user geolocation via graph convolutional networks, in: Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL), pp. 2009–2019.

[51] Rahimi, A., Vu, D., Cohn, T., Baldwin, T., 2015b. Exploiting text and network context for geolocation of social media users, in: Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, (NAACL-HLT), pp. 1362–1367.

[52] Ren, K., Zhang, S., Lin, H., 2012. Where are you settling down: Geo-locating twitter users based on tweets and social networks, in: Proceedings of the Asia Information Retrieval Societies Conference (AIRS), pp. 150–161.

[53] Rodrigues, E., Assunção, R., Pappa, G.L., Renno, D., Meira Jr, W., 2016. Exploring multiple evidence to infer users' location in twitter. Neurocomputing 171, 30–38. doi:10.1016/j.neucom.2015.05.066.

[54] Roller, S., Speriosu, M., Rallapalli, S., Wing, B., Baldridge, J., 2012. Supervised text-based geolocation using language models on an adaptive grid, in: Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL), pp. 1500–1510.

[55] Schnake, T., Eberle, O., Lederer, J., Nakajima, S., Schütt, K.T., Müller, K.R., Montavon, G., 2020. Xai for graphs: Explaining graph neural network predictions by identifying relevant walks. arXiv preprint arXiv:2006.03589 .

[56] Sinnott, R.W., 1984. Virtues of the haversine. Sky and Telescope 68, 159.

[57] Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., Bengio, Y., 2018. Graph attention networks, in: Proceedings of the International Conference on Learning Representations (ICLR).

[58] Wagstaff, E., Fuchs, F.B., Engelcke, M., Posner, I., Osborne, M., 2019. On the limitations of representing functions on sets, in: Proceedings of the International Conference on Machine Learning (ICML), pp. 6487–6494.

[59] Wang, S.H., Govindaraj, V.V., Górriz, J.M., Zhang, X., Zhang, Y.D., 2021. Covid-19 classification by fgcnet with deep feature fusion from graph convolutional network and convolutional neural network. Information Fusion 67, 208–229.

[60] Wing, B., Baldridge, J., 2014. Hierarchical discriminative classification for text-based geolocation, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 336–348.

[61] Wing, B.P., Baldridge, J., 2011. Simple supervised document geolocation with geodesic grids, in: Proceedings of the Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL), pp. 955–964.

[62] Wu, F., Jr., A.H.S., Zhang, T., Fifty, C., Yu, T., Weinberger, K.Q., 2019. Simplifying graph convolutional networks, in: Proceedings of the International Conference on Machine Learning (ICML), pp. 6861–6871.

[63] Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., Philip, S.Y., 2020. A comprehensive survey on graph neural networks. IEEE Transactions on Neural Networks and Learning Systems , 1–21doi:10.1109/TNNLS.2020.2978386.

[64] Xu, K., Hu, W., Leskovec, J., Jegelka, S., 2019. How powerful are graph neural networks?, in: Proceedings of the International Conference on Learning Representations (ICLR).

[65] Xu, K., Li, C., Tian, Y., Sonobe, T., Kawarabayashi, K.i., Jegelka, S., 2018. Representation learning on graphs with jumping knowledge networks, in: International Conference on Machine Learning, PMLR. pp. 5453–5462.

[66] Ying, Z., Bourgeois, D., You, J., Zitnik, M., Leskovec, J., 2019. Gnnexplainer: Generating explanations for graph neural networks, in: Advances in Neural Information Processing Systems, pp. 9240–9251.

[67] Ying, Z., You, J., Morris, C., Ren, X., Hamilton, W., Leskovec, J., 2018. Hierarchical graph representation learning with differentiable pooling, in: Advances in Neural Information Processing Systems, pp. 4805–4815.

[68] You, J., Ying, R., Leskovec, J., 2019. Position-aware graph neural networks, in: Proceedings of the International Conference on Machine Learning (ICML), pp. 7134–7143.

[69] Yuan, H., Tang, J., Hu, X., Ji, S., 2020. XGNN: towards model-level explanations of graph neural networks, in: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD), ACM. pp. 430–438.

[70] Zhang, Z., Bu, J., Ester, M., Zhang, J., Yao, C., Yu, Z., Wang, C., 2019. Hierarchical graph pooling with structure learning. arXiv preprint arXiv:1911.05954 .

[71] Zheng, X., Han, J., Sun, A., 2018. A survey of location prediction on twitter. IEEE Transactions on Knowledge and Data Engineering (TKDE) 30, 1652–1671. doi:10.1109/TKDE.2018.2807840.

[72] Zhong, T., Wang, T., Zhou, F., Trajcevski, G., Zhang, K., Yang, Y., 2020. Interpreting twitter user geolocation, in: Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL), pp. 853–859.

[73] Zhou, F., Cao, C., Zhong, T., Geng, J., 2021a. Learning meta-knowledge for few-shot image emotion recognition. Expert Systems with Applications , 114274.

[74] Zhou, F., Li, L., Zhang, K., Trajcevski, G., 2021b. Urban flow prediction with spatial–temporal neural odes. Transportation Research Part C: Emerging Technologies 124, 102912.

[75] Zhou, F., Wang, P., Xu, X., Tai, W., Trajcevski, G., 2021c. Contrastive trajectory learning for tour recommendation. ACM Transactions on Intelligent Systems and Technology .

[76] Zhou, F., Xu, X., Trajcevski, G., Zhang, K., 2021d. A survey of information cascade analysis: Models, predictions, and recent advances. ACM Comput. Surv. 54, 27:1–27:36.

[77] Zhou, F., Yang, Q., Zhong, T., Chen, D., Zhang, N., 2021e. Variational graph neural networks for road traffic prediction in intelligent transportation systems. IEEE Transactions on Industrial Informatics 17, 2802–2812. doi:10.1109/TII.2020.3009280.

[78] Zhou, J., Cui, G., Hu, S., Zhang, Z., Yang, C., Liu, Z., Wang, L., Li, C., Sun, M., 2020. Graph neural networks: A review of methods and applications. AI Open 1, 57–81.

[79] Zola, P., Cortez, P., Carpita, M., 2019. Twitter user geolocation using web country noun searches. Decision Support Systems 120, 50–59. doi:10.1016/j.dss.2019.03.006.