Your Friends Reveal Where You Are: Location Estimation based on Friends' Locations in Geosocial Networks

Ketevan Gallagher
University of Toronto
Toronto, Ontario, Canada
ketevan.gallagher@mail.utoronto.ca

Vishwanath Seshagiri
Emory University
Atlanta, Georgia, USA
vishwanath.seshagiri@emory.edu

Theodoros Chondrogiannis
NTNU
Trondheim, Norway
theodoros.chondrogiannis@ntnu.no

Panagiotis Bouros
Johannes Gutenberg University Mainz
Mainz, Germany
bouros@uni-mainz.de

Andreas Züfle
Emory University
Atlanta, Georgia, USA
azufle@emory.edu

Abstract

Geosocial networks serve as a critical bridge between cyber and physical worlds by linking individuals to locations. In many real-world scenarios, both the structure of social networks and the spatial distribution of places are known—yet the information that links people to locations is missing. This absence is often intentional to ensure user privacy. In this work, we investigate the feasibility of estimating locations based solely on network structure and a limited set of known user-location pairs. We propose and evaluate four algorithms for linking social and spatial networks: (i) a greedy assignment algorithm, (ii) a hierarchical approach using graph partitioning, (iii) a spatially-aware adaptation of force-directed graph drawing, and (iv) a modified version of Spatial Label Propagation. Each method is enhanced to incorporate a small number of anchor vertex—users with known locations. Using social network data from Virginia, USA, our empirical evaluation shows that even a sparse set of anchor points can enable accurate estimation of users' home locations. These findings highlight both the analytical value and the privacy risks associated with linking social and spatial data.

CCS Concepts

• **Information systems** → **Geographic information systems**.

Keywords

Geosocial Networks, Link Prediction, Location Inference

ACM Reference Format:
Ketevan Gallagher, Vishwanath Seshagiri, Theodoros Chondrogiannis, Panagiotis Bouros, and Andreas Züfle. 2025. Your Friends Reveal Where You Are: Location Estimation based on Friends' Locations in Geosocial Networks. In *The 1st ACM SIGSPATIAL International Workshop on Spatial Intelligence for Smart and Connected Communities (SpatialConnect '25), November 3–6, 2025, Minneapolis, MN, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3764924.3770897

1 Introduction

Location-Based Social Networks (LBSNs) [29], often also called geosocial networks [4], capture both (1) social relationships such as friendship between individuals or populations, and (2) the locations of these individuals. By bridging physical and virtual spaces, LBSNs have numerous applications, ranging from online-physical dating to recommendation and advertising [19, 29]. An example of a real geosocial network is shown in Figure 1. Here, locations of ZIP code centroids are linked based on the strength of their social connectedness. This network is generated using the Facebook Social Connectedness Index [3].

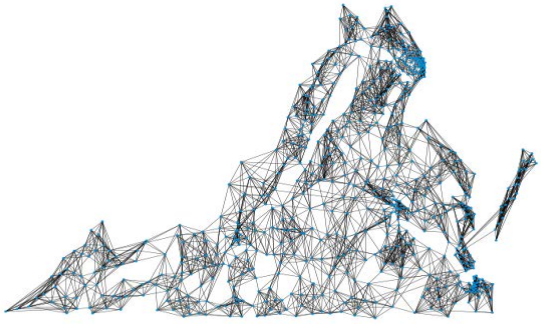


Figure 1: Geosocial Network using Facebook Social Connectedness Data between Zone Improvement Plan (ZIP) Region Centroids for the State of Virginia (VA), USA.

In this work, we investigate the severity of the privacy risks of being part of a geosocial network. Specifically, we explore whether it is possible to estimate the location of a user who does not share their location information by using the location of those in their social network who do share their location. Estimating the location of users would incur severe privacy risks, as past research [8, 25] has shown that only three location points are required to identify most individuals among millions of individual people. Beyond privacy loss, the exposure of location information creates tangible safety threats for communities: stalkers, burglars, or malicious actors can exploit mobility patterns to determine when people are away from home, identify vulnerable populations, or target individuals, such as minors or elderly, for physical harm. These risks are amplified in large-scale mobility datasets that cover entire urban regions, as inference models can uncover behavioral regularities and predict future movements. Consequently, location inference is not only a technical challenge in data management and anonymization but also a fundamental ethical concern for safeguarding personal security in spatial computing research.

SpatialConnect '25, Minneapolis, MN, USA

ACM ISBN 979-8-4007-2187-8/2025/11
https://doi.org/10.1145/3764924.3770897

The question we investigate in this paper is not whether we can identify individuals based on where they go or where their direct connections are, but where others in their social network are. The remainder of this work is organized as follows. Section 2 surveys existing research on location-based social networks and location privacy. Section 3 formally defines the problem of identifying vertices in a location-based social network based on the location of others in their social network. Section 4 then presents four algorithms to solve this problem: (1) a greedy approach which maps users to locations by starting with users with a high degree and matching them to dense locations, (2) an approach that hierarchically clusters both the social and the spatial network and then maps similar clusters to each other, (3) an approach based on graph-drawing, by minimizing distance to friends with known location and maximazing distance to non-friends with known locations, and (4) an approach that modfies the Spatial Label Propagation algorithm [15] so that vertices are matched to discrete and not continuous locations. Section 5 shows both qualitative and quantitative experiments to support our hypothesis that it is possible to identify the location of users if a large fraction of other users share their locations. Finally, Section 6 provides a brief conclusion to the paper and a discussion of areas of further work.

2 Related Work

A substantial body of research has demonstrated that social networks exhibit significant spatial autocorrelation, whereby individuals are more likely to form social ties with geographically nearby others. This phenomenon, often referred to as spatial homophily, is rooted in both opportunity and preference: individuals who live, work, or socialize in close geographic proximity are more likely to interact and form relationships [9, 22]. Early studies of offline networks, such as those examining high school friendship patterns, found strong evidence of spatial proximity as a key determinant of tie formation [23]. With the advent of large-scale digital social networks, these findings have been corroborated at scale. For instance, analyses of Twitter and Facebook data have shown that the probability of a social connection decays with geographic distance, often following an inverse power-law relationship [2, 5]. Moreover, research on mobile phone communication networks has revealed spatially clustered social structures, where communities detected in the call graph tend to align with administrative or natural geographic boundaries [12]. These observations underscore the importance of incorporating geographic context into the analysis and modeling of social networks. In this work, we aim to leverage the spatial homophily of social networks to infer the location of users who do not share their location. Thus, our goal is to infer the location of a user based on the location of their friends. Recent studies have demonstrated that even well-anonymized datasets, such as mobility or transaction logs, can be re-identified with alarming accuracy using a small number of spatiotemporal points [8, 25, 28]. This has raised significant concerns in the context of data sharing and geosocial research, as location data is often uniquely identifying due to the regularity and sparsity of individual movement patterns [13]. This paper complements this existing research by investigating how individuals can be identified not based on their own location data, but based on the location data of those in their network. As a result, research on privacy-preserving algorithms—such as differential privacy, k-anonymity, and geo-indistinguishability—has become a critical area for enabling data-driven innovation without compromising individual privacy [26, 27].

Recent research investigated the use of social networks to estimate geographic locations of users based on spatial homophily. Early works in this area showed that user location can be estimated with surprising accuracy by leveraging the locations of a user's friends or followers in the social graph [2, 7, 24]. Backstrom et al. [2] demonstrated that the location of a Facebook user can be predicted with high accuracy using the locations of their friends. Pontes et al. [24] explored cross-platform location inference using data from Foursquare, Google+, and Twitter, demonstrating that social proximity, combined with profile attributes, could predict users' home locations at city and neighborhood scales. While these works solve the same problem defined in this work, the proposed algorithm can only be used to infer the location of users having at least one (direct) friend with known location, and cannot be applied in the case where only a few sparse user locations are known. Jurgens et al. [15] solves a very similar problem to ours, but locations are generated continuously. In our work, we aim to match users to a set of known locations. Further refinements have incorporated deep learning and probabilistic models. Li et al. [20] introduced the Multiple Location Profiling (MLP) model to account for users having multiple significant locations (e.g., home, work, travel destinations). More recently, Luceri et al. [21] applied graph-based neural networks to infer user locations in Twitter, showing that even when only a fraction of users share geo-tagged posts, the location of others can be inferred through social ties with non-trivial accuracy. While these works also aim at inferring the location of users of a social network, they assume that additional content (such as microblogs or check-ins) is available to provide location samples of users. In contrast, our problem definition assumes that the user does not share any information, and our goal is to infer the location of users solely on the locations of others in their social network.

3 Problem Definition

This section illustrates the problem proposed in this work and provides a formal problem definition.

Example 3.1. Figure 2 shows information about eleven users of an LBSN: The left section of the figure shows the users' geolocations, and the right shows their social network. However, the mapping between these two sets—that is, which user is located where—is only known for a small subset of users. Our goal is to estimate this mapping for all users. In the example, we see that users 5, 6, 9, and 10 form a clique in the social network. Following the assumption of spatial homophily, it seems likely that these social network users may correspond to locations *B-E*, which are close to each other. Users 0-2 also form a social network clique that may correspond to the locations *G-I*. In practice, we may have thousands of users and locations, and there may be many spatial clusters of the same size, making such a trivial inference impossible. We formalize this problem as follows:

Definition 3.2 (LBSN Location Estimation Problem). Let $LBSN = (V, E)$ be a location-based social network consisting of a set of vertices (users) V and a set of links (friendship relations) $E \subseteq V \times V$

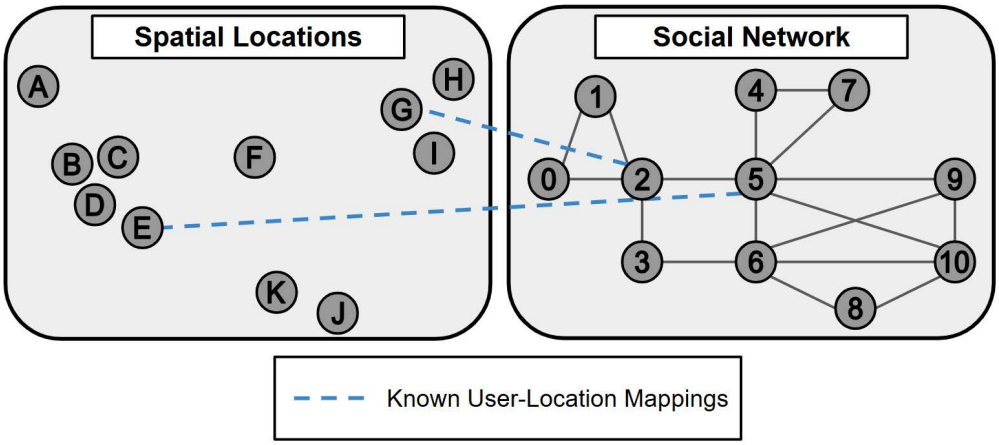


Figure 2: Example of the LBSN Location Estimation Problem: (Left) Spatial locations of users. This may be a geographical space with axes corresponding to geo-coordinates. (Right) Social network of users. Vertices are users and edges are social links between users. The axes of this space do not have semantic meaning. Links between nodes across the two spaces indicate the known location of users. The LBSN Location Inference Problem seeks to infer the location of all users.

between vertices. Furthermore, let S be a set of spatial locations (home locations or regions). Each vertex $v \in V$ has a geolocation $loc(v) \in S$. We assume that this location is observable only for a subset $V_{obs} \subset V$. Our goal is to estimate the location of other vertices $V \subseteq V_{obs}$ based on the locations of vertices in V_{obs} and based on the network topology of $LBSN$.

We assume that the set of all possible locations is known, such as residential buildings available in OpenStreetMap [1, 14]. We also assume that the mapping $loc(v) : V \rightarrow S$ is injective, such that no two social network users can be mapped to the same location. This assumption comes without loss of generality, as we can include multiple vertices with the same location in S to denote locations that may have multiple users living in the same place. We also assume this mapping to be surjective, such that each spatial location must have one social network user mapped to it. We note that this assumption does incur a loss of generality, as in practice, many or even most individuals may not be users of the same social networking service. However, we keep this assumption to simplify the problem, and we note that relaxing this assumption remains an open research direction. Thus, we assume that function $loc(v)$ is bijective such that there exists a one-to-one matching between the social network vertices V and the spatial locations S.

4 LBSN Location Estimation Algorithms

This section describes four algorithms to solve the LSBN Location Estimation Problem (Definition 3.2). The first algorithm described in Section 4.1 is a Greedy baseline approach, which follows the intuition that is used in the toy example (Example 3.1): Iteratively matching the social network vertex with the largest number of connections to the spatial locations with the highest local density. Then, Section 4.2 presents an approach based on hierarchically clustering both the social network and the spatial locations and iteratively mapping social clusters to location clusters to each other. Then, Section 4.3 presents an approach inspired by the force-directed graph drawing algorithm [16] which positions vertices using attractive forces along edges and repulsive forces between vertices. We augment this algorithm by including spatial locations as fixed vertices (whose locations cannot change) that exhibit attraction forces, ensuring that spatially dense regions attract social network vertices. Finally, in Section 4.4, we augment the Spatial Label Propagation algorithm presented in [15] so that vertices are matched to given locations. Additional implementation details and Python code for each algorithm can be found in the GitHub repository at https://github.com/KetevanGallagher/Geosocial-Network-Location-Estimation.

4.1 Greedy Algorithm

The first algorithm we present for matching vertices in a social network to spatial locations leverages spatial homophily, i.e., the fact that links in geosocial networks exhibit spatial correlation. The Greedy algorithm iteratively processes the unprocessed vertices that have the highest number of connections to vertices with known locations. Ties are broken by selecting, among these vertices, the vertex with the highest number of total connections. Let v denote the selected social network vertex and let $known(v)$ denote the (possibly empty) set of vertices connected to v for which their location is known. The selected vertex v is assigned to the location that minimizes the sum of distances to the connected known locations. If v is not connected to any vertices with known locations, then we choose the medoid of the unassigned locations. Then, the social network neighbors of v whose locations are unknown are assigned to the spatial k-nearest neighbors of the selected location. Vertices that have a degree of zero are not included in this process and are randomly assigned a location from the remaining available locations. This approach aims at making sure that (1) the user with the most social ties is mapped to the densest location and (2) that the neighbors of this user are mapped to the nearest locations.

Example 4.1. We illustrate the greedy algorithm using the example in Figure 2. We start by finding the social network vertex having the largest number of connections to vertices with known

locations. Since locations are known only for social network vertices 2 and 5, this results in a many-way tie. Vertices 0, 1, 3, 4, 6, 7, 9, and 10 are all connected to either Vertex 2 or Vertex 5, but none are connected to both. We break this tie by selecting among these vertices the vertex having the largest degree, which is Vertex 6 with a degree of 5. Thus, we select v to be Vertex v_6. The set $known(v_6)$ of neighbors of v_6 having a known location includes only vertex v_5. We then map v_5 to the vertex that minimizes the distances to $known(v_6)$, which is the vertex with the least distance to $loc(v_5)$. This is known to be Location E. Location D has the least distance to Location E, so we map Vertex v_6 to Location D. We then identify the $deg(v_6) - |known(v_6)| = 5 - 1 = 4$ nearest neighbors of D among the unassigned locations which includes Locations A, B, C, and F. We next map the unassigned neighbors v_3, v_8, v_9, and v_{10} to these locations by having vertices with higher degree connect to locations closer to D. Since Vertex 10 has the highest degree of 4 it is assigned to the closed Location B; then Vertex 9 with a degree of 3 is assigned to the second closest location C; and then Vertices 3 and 8 each have a degree of 2 and are arbitrarily assigned to Locations A and F, respectively. In this example, we see that the social clique of Vertices 5, 6, 9, and 10 is correctly mapped to Locations E, D, C, and B. However, Vertex 3 is incorrectly mapped to Location A instead of Location F. We iterate this process until all vertices are assigned to locations. We omit the remaining iterations but note that, typical for Greedy Algorithms, incorrect assignments made in previous iterations will cascade into later iterations as there is no mechanism to correct bad decisions. In this example, Vertex v_8, which is the vertex that is actually located at Location A, will be mapped to a location far from Location A since all nearby locations have already been assigned.

4.2 Partitioning-Based Algorithm

The Greedy algorithm iteratively processes vertices one by one and wrong assignments cause incorrect assignments in later iterations. Rather than mapping individual vertices, we propose an algorithm that maps entire clusters of similar social vertices and spatial locations to each other. To identify clusters, we use the METIS network partitioning algorithm [18] to hierarchically partition both the social network and the spatial location graph, where links correspond to spatial proximity. We use METIS to divide a network into k partitions. For our experiments, we use $k = 10$ while the population is greater than 100. Otherwise, the population is divided into partitions of ten vertices. Social network partitions and location clusters are matched by minimizing the sum of the distance between centroids multiplied by the number of connections between two partitions for each combination of partitions. Each subdivision is recursively partitioned until the resulting partitions have fewer than 30 vertices. Within partitions, vertices are matched to locations in descending order of their degree, with higher degrees matched to the medoid of the remaining locations. For each vertex processed, its neighbors are assigned to the location closest to that vertex. Similar to the Greedy algorithm, vertices that have a degree of zero are matched randomly to remaining locations once all other vertices have been assigned to a location. In this algorithm, vertices that form similar communities in the social network are often matched to locations that resemble these communities.

4.3 Graph Drawing

While the Partitioning-Based algorithm maps clusters rather than points, it still follows the Greedy paradigm, and erroneous assignments made in early iterations cascade into later iterations, as there is no mechanism to revise decisions. To allow reassignment, we next propose an optimization approach based on the force-directed placement graph drawing algorithm [10], which is commonly used to visualize networks. In this algorithm, vertices that are connected attract each other while disconnected vertices repulse each other. To estimate spatial locations, we make two adjustments to the algorithm: (1) Social network vertices with known locations are fixed to the location that represents their spatial location. This ensures that known locations are mapped to the correct location. (2) Spatial locations exhibit a small attraction force to ensure that spatially dense areas will attract social network users. The corresponding forces (i) attract/repulse social network vertices, (ii) known locations hold vertices in place, and (iii) spatial locations attracting social network vertices are simulated until convergence. Thus, a new position is generated for each vertex. The "temperature" of the system, a factor that controls the movement speed of the vertices, is capped at 0.1 to maintain stability and avoid erratic movements. For our data, the weighting of the attraction between vertices is increased from one, the default, to the number of the population. This adjustment prioritizes the influence of connected vertices in the layout and ensures that vertices connected to fixed vertices are drawn more strongly toward them, while diminishing the effect of repulsive forces from unconnected vertices. If there are no fixed vertices, the centroid parameter is used. The centroid parameter is the coordinate pair around which the layout is centered. The centroid parameter is set to the centroid of the locations. The Spring Layout also has a parameter for the optimal distance between vertices. This parameter is set to the average distance between locations. The Spring Layout returns a new set of positions for each vertex, and these vertices are matched to the available location using two different procedures.

The first is by using a greedy method: the vertex with the highest degree is assigned to the closest available true location, and its neighbors are assigned to the locations closest to them. The vertex with the next highest degree is assigned to its closest available location, and the process repeats until all vertices have been assigned a location.

The second is by using an optimal method: vertices are assigned to a location using a modified Jonker-Volgenant algorithm[6].

4.4 Spatial Label Propagation

The final algorithm is Spatial Label Propagation, outlined in [15]. If a vertex has any neighbors with a known location, its new location is the average of its neighbors' locations. This process repeats until all vertices have been assigned a location. Although this generates locations for all vertices, the locations are generated in continuous space, so they will not match the true locations that are already known. Thus, we expand this algorithm to match the generated locations to given locations using either the greedy or optimal method described in 4.3. If any vertices have a degree of zero, they are assigned randomly to available remaining locations after all other vertices have been assigned to locations. One disadvantage of this algorithm is that it cannot be used for networks that do

Known Locations	Random	Greedy	Partitioning Based	Graph Drawing Greedy	Graph Drawing Optimal	SLP Greedy	SLP Optimal
0	209.6	170.7	104.7	199.3	201.7	N/A	N/A
3	208.0	182.8	122.1	139.5	204.9	120.5	122.9
68	188.4	164.1	125.5	88.82	91.72	24.12	16.35
206	146.4	123.3	96.35	39.91	22.97	10.90	6.061
344	105.7	92.52	86.91	24.55	11.87	5.905	2.373
481	61.92	30.81	58.20	13.49	5.850	2.881	0.8500

Table 1: Average distance (in kilometers) between estimated and true locations on the Facebook VA dataset.

Known Locations	Random	Greedy	Partitioning Based	Graph Drawing Greedy	Graph Drawing Optimal	SLP Greedy	SLP Optimal
0	14.32	11.92	13.29	13.22	12.28	N/A	N/A
3	14.00	12.29	12.87	12.32	13.80	8.941	8.672
24	12.83	9.662	8.866	11.84	11.35	3.563	3.094
72	9.867	7.778	8.184	8.769	8.734	2.004	1.633
121	7.095	5.699	6.672	5.612	5.352	1.278	0.9474
169	4.169	3.139	3.820	3.355	3.147	0.6676	0.4058

Table 2: Average distance (in kilometers) between estimated and true locations on the Fairfax Mobility dataset.

not have any known locations. Although this algorithm may face similar issues as the Greedy algorithm, where vertices that are matched incorrectly early on have a cascading negative effect on the rest of the vertices, this effect is minimized because vertices are assigned continuous locations. The new location of each vertex is informed by all neighbors with locations, and not just one of them.

5 Experimental Evaluation

This section presents an empirical evaluation of the proposed algorithms. Section 5.1 describes the datasets used for our evaluation. Then, Section 5.2 uses these datasets to evaluate all algorithms quantitatively by measuring the distances between estimated and ground truth locations. Additionally, we present a qualitative evaluation to interpret and understand the strengths and weaknesses of each algorithm in Section 5.3 and finally, Section 5.4 gives an overview of the runtimes of each algorithm.

5.1 Datasets

To evaluate whether the proposed algorithms can identify the locations of users in a social network, we use two real-world geosocial network datasets and a synthetically generated dataset. First, we leverage data provided by Facebook Data for Good, referred to as the Social Connectedness Index (SCI) [3]. The dataset provides a measure of social connectedness between all pairs of 688 ZIP codes for VA, USA. We connect each ZIP code to the ten ZIP codes with which it has the highest SCI. The resulting network is depicted in Figure 1 and has an average degree of 12.55. The second dataset delineates space into census tracts for the region of Fairfax County, VA, USA. Human mobility data from SafeGraph [17] from the date 1/4/2020 is used to estimate geosocial connections between census tracts. We selected the top 87% of population flows as links between census tracts, resulting in an average degree of 10.7.

We also provide experiments for synthetically generated geosocial data. We utilize the Geosocial Erdős-Rényi network model detailed in [11]. This model creates random locations in a two-dimensional $[0, 1]^2$ unit space with links based on distance, following a power law having closer locations more likely to be connected.

5.2 Location Estimation Results

Prediction results on the Facebook data are shown in Table 1 for different levels of known locations. Each of our proposed algorithms was run with the locations of ZIP codes in Virginia and a variable number of known locations. The social network was generated using the Facebook Social Connectedness Index. As the number of ZIP codes in Virginia is 688, the set of known locations to test with was chosen as 3, 68, 206, 344, and 481, which is roughly 0%, 10%, 30%, 50%, and 70% of the locations, respectively. Each level of known locations was run for 30 trials on each proposed algorithm, and the average of these trials is displayed in Table 1. First, we observe that all six algorithms consistently outperform a random baseline, which maps social network vertices to locations randomly. We observe that the Partitioning-Based algorithm can outperform the other algorithms for cases with no known locations, but the Graph Drawing algorithm, and especially Spatial Label Propagation, benefit from having more a priori location information. For cases with only a few known locations, the greedy version of the Graph Drawing algorithm and Spatial Label Propagation outperform the optimal version, but for all cases with a large number of locations that are known, the optimal version has a shorter average distance. For all cases with known locations, both versions of Spatial Label Propagation outperform all other algorithms.

For the levels of zero, three, and 68 known locations for the Graph Drawing algorithm, it can be observed that the optimal version has a higher average distance to the correct location than the greedy version. While this may seem counterintuitive, the optimal version does not know the ground truth. Thus, although it matches the new, generated locations in a way such that the distances from the new locations to the given locations are minimized, these new locations may be far from their correct locations. In this case, while the greedy

Population	Known Locations	Random	Greedy	Partitioning Based	Graph Drawing Greedy	Graph Drawing Optimal	SLP Greedy	SLP Optimal
100	0	0.5080	0.4991	0.4525	0.5198	0.5101	N/A	N/A
100	3	0.5005	0.4501	0.4857	0.3330	0.3010	0.3531	0.3346
100	10	0.4699	0.3832	0.4359	0.1356	0.0931	0.2050	0.1745
100	50	0.2599	0.2108	0.2384	0.05462	0.03213	0.06192	0.03515
500	0	0.5176	0.5115	0.3746	0.4881	0.4840	N/A	N/A
500	3	0.5168	0.4905	0.4172	0.4459	0.4418	0.3776	0.3707
500	10	0.5122	0.4835	0.4144	0.2755	0.2471	0.2393	0.2190
500	50	0.4701	0.4307	0.3128	0.1343	0.07462	0.1180	0.09681
1000	0	0.5218	0.5176	0.3561	0.5069	0.5032	N/A	N/A
1000	3	0.5202	0.4867	0.3976	0.4691	0.4549	0.3754	0.3784
1000	10	0.5169	0.4860	0.3853	0.3288	0.3051	0.2424	0.2278
1000	50	0.4912	0.4709	0.3717	0.1656	0.09407	0.1187	0.09981

Table 3: Average distance between estimated and true locations on synthetic spatial Erdős-Rényi data.

version does not minimize the distances from the new locations to the given locations, the locations assigned by the greedy version may be closer to their correct location.

Results for the Fairfax Mobility data and the synthetic dataset for different population sizes are shown in Table 2 and Table 3, respectively. For the synthetic dataset, random locations were generated, and a Geosocial Erdős-Rényi Network was generated from these random locations. Three levels of the total number of locations were used (100, 500, and 1,000), and four levels of known locations were used (0, 3, 10, and 50). Our proposed algorithms were then run with these random locations and generated Geosocial Erdős-Rényi Networks. Similar to the Facebook results, we observe on the Fairfax dataset that the Spatial Label Propagation approach consistently yields the highest accuracy location prediction. We also observe that the optimal version outperforms the greedy version of Spatial Label Propagation for each level of known locations.

For the synthetic dataset, we observe that for a population of 100, the optimal version of the Graph Drawing Algorithm has the smallest average distance to the correct location. For all other levels and populations, besides the population of 500 with 50 known locations, the optimal version of Spatial Label Propagation outperforms other algorithms. We observe a similar pattern to the Fairfax dataset, and find that for the level with no known locations, the Partitioning-Based algorithm performs best.

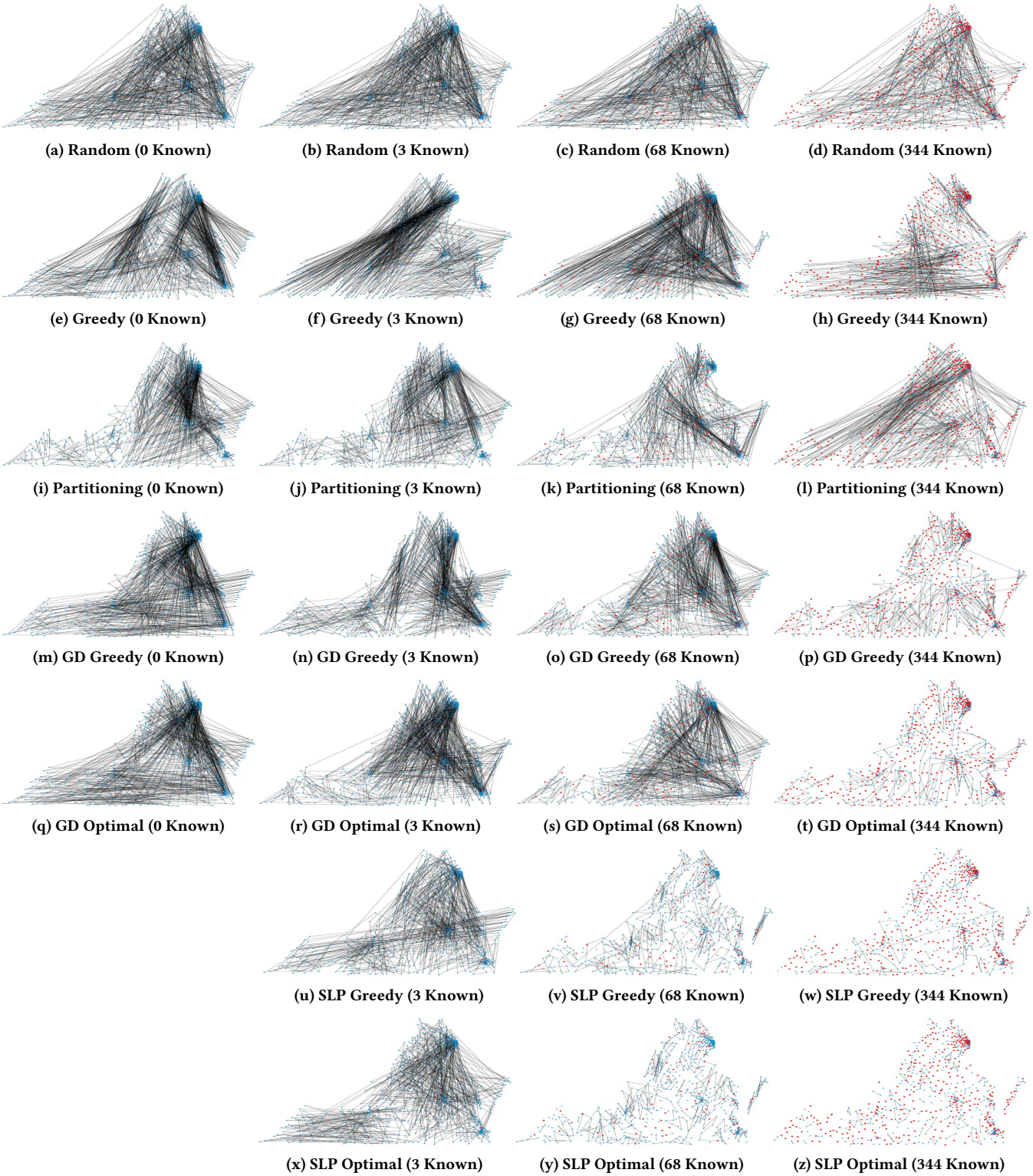
5.3 Qualitative Results

Through the qualitative results shown in Figures 3 and 4, we can see that as the number of known locations increases, vertices are assigned to locations closer to their actual location. Each figure shows unknown locations in blue and known locations in red. The edges connect the location that a vertex in the social network was assigned to its actual location. Thus, a vertex assigned correctly will have no edge. Graphs with many long edges have many vertices that are assigned far from their correct location, while graphs with short edges and a large amount of whitespace have many vertices that are assigned close to or simply to their correct location. The blank cells in both figures are present because both SLP algorithms cannot be used for cases with zero known locations.

5.3.1 Virginia Facebook Data. Figure 3 shows the distances between estimated and true locations on the Virginia Facebook data for a random baseline and all proposed algorithms having zero, three, 68, and 344 known locations. In Figures 3a-3d, for the random baseline, we do not observe, as expected, any spatial patterns and instead observe many long edges connecting vertices uniformly. For the Greedy algorithm, we observe in Figures 3e-3h that some areas match well while some communities are confused, leading to large errors (indicated by long lines) in some parts while other parts are matched well (indicated by short or no lines at all).

For the Partitioning-Based algorithm in Figures 3i-3l, we see that when communities are matched incorrectly, large errors occur due to mapping every single vertex from one partition to the true locations of the community. But when the high-level mapping of partitions is correct, we observe very good matchings within the partition. For example, for the case having 68 known locations, we see that the south-western area is matched very well. But for the case of 344 known locations, the mapping of partitions mapped the south-western area to the Northeast, creating very large lines across the entire map.

Figures 3m-3p, show the results for the Graph Drawing Algorithm. We observe in Figures 3p and 3t that for the case of having very many known locations, the Graph Drawing Algorithm, independent of the mapping choice (Greedy vs Optimal), performs extremely well. But for cases having fewer known locations, we still observe a high level of confusion between estimated and true locations. Finally, we observe the results of the Spatial Label Propagation Algorithm in Figures 3u-3z. Since this algorithm cannot be applied when there are no known locations, there are no results for cases with zero known locations. But we observe that for the case of having many known locations, this algorithm performs particularly well. For the case of having 344 known locations, the result seems even better than for the Graph Drawing Algorithm. More importantly, we see that this algorithm, unlike others, can obtain very good results even with only 68 known locations. We also see that many vertices that are unknown (indicated by the blue color in Figure 3) have no edge at all, indicating that this vertex was matched correctly to the ground truth. We also observe that the optimal matching strategy yields visibly better results compared to the greedy matching strategy, while not adding much computational overhead, as we will show in Section 5.4.



(a) Random (0 Known) (b) Random (3 Known) (c) Random (68 Known) (d) Random (344 Known)

(e) Greedy (0 Known) (f) Greedy (3 Known) (g) Greedy (68 Known) (h) Greedy (344 Known)

(i) Partitioning (0 Known) (j) Partitioning (3 Known) (k) Partitioning (68 Known) (l) Partitioning (344 Known)

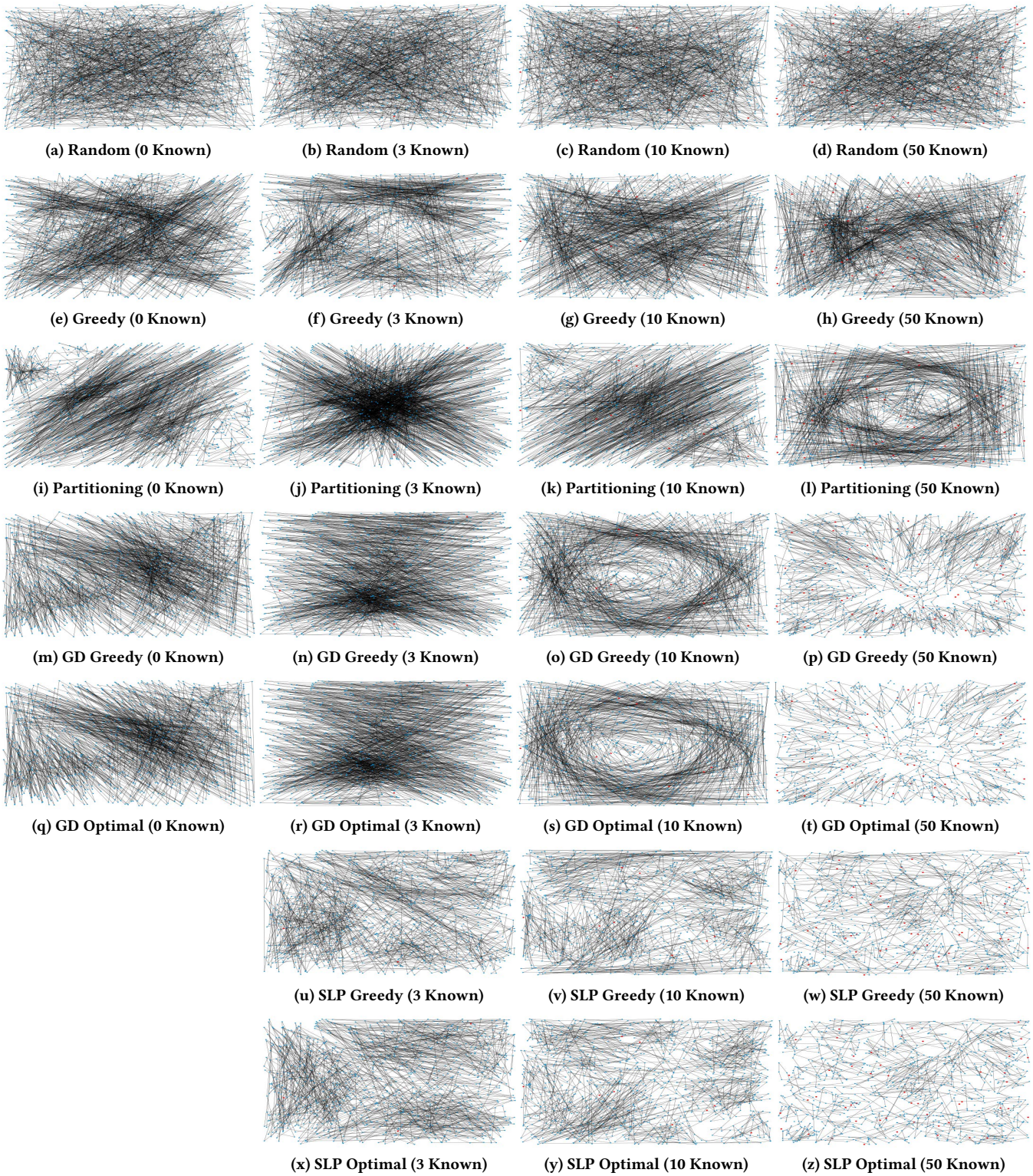
(m) GD Greedy (0 Known) (n) GD Greedy (3 Known) (o) GD Greedy (68 Known) (p) GD Greedy (344 Known)

(q) GD Optimal (0 Known) (r) GD Optimal (3 Known) (s) GD Optimal (68 Known) (t) GD Optimal (344 Known)

(u) SLP Greedy (3 Known) (v) SLP Greedy (68 Known) (w) SLP Greedy (344 Known)

(x) SLP Optimal (3 Known) (y) SLP Optimal (68 Known) (z) SLP Optimal (344 Known)

Figure 3: Links between estimated location and true location using Facebook data for a random baseline and the proposed algorithms having 0 (left), 3, 68 (10%), and 344 (50%) known locations (right).



(a) Random (0 Known) (b) Random (3 Known) (c) Random (10 Known) (d) Random (50 Known)

(e) Greedy (0 Known) (f) Greedy (3 Known) (g) Greedy (10 Known) (h) Greedy (50 Known)

(i) Partitioning (0 Known) (j) Partitioning (3 Known) (k) Partitioning (10 Known) (l) Partitioning (50 Known)

(m) GD Greedy (0 Known) (n) GD Greedy (3 Known) (o) GD Greedy (10 Known) (p) GD Greedy (50 Known)

(q) GD Optimal (0 Known) (r) GD Optimal (3 Known) (s) GD Optimal (10 Known) (t) GD Optimal (50 Known)

(u) SLP Greedy (3 Known) (v) SLP Greedy (10 Known) (w) SLP Greedy (50 Known)

(x) SLP Optimal (3 Known) (y) SLP Optimal (10 Known) (z) SLP Optimal (50 Known)

Figure 4: Links between estimated location and true location using 1,000 random locations and a Geosocial Erdős-Rényi network for a random baseline and proposed algorithms having 0 (left), 3, 10, and 50 known locations (right).

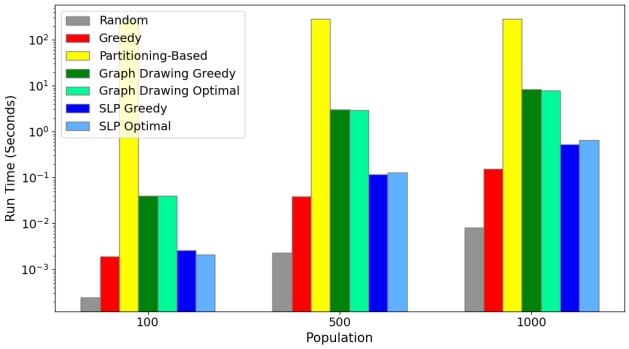


Figure 5: Average runtimes for all algorithms for three population levels with 50 known locations.

Figure 4 shows the result of all algorithms on the synthetically generated Erdős-Rényi network having 100 vertices. In addition, we also show the corresponding results for 1000 vertices in our GitHub Repository found at https://github.com/KetevanGallagher/Geosocial-Network-Location-Estimation. Figures 4e-4h again show no spatial patterns using the random baseline for matching vertices to locations. The Greedy algorithm, observed in Figures 4e-4h again shows that some areas are matched well, which may correspond to vertices that are chosen in early greedy iterations. But some areas are completely confused, leading to groups of large groups of mismatched vertices visualized by groups of long edges. For the partitioning algorithm shown in Figures 4i-4l we observe quite some different patterns for zero, three, ten and 50 known locations: For zero and ten locations we observe many top-right directed diagonal edges, and for three known locations we observe both top-right and top-left directed diagonal edges leading to a dark area in the center of the map. Conversely, for 50 known locations we observe a circular structure avoid the center of the map. This can be explained by how the partitions of the network were mapped to the ground truth partitions. For zero and ten known locations, it appears that the top-left and bottom-right regions have been confused, but the top-right and bottom-left regions were mapped correctly. For the three known location case, it appears that all regions were diagonally confused, and for the case of having 50 known locations, there was a circular confusion between quadrants of the synthetic map. Overall, we see a large error for any number of known locations, even though sometimes the algorithm was able to correctly match entire partitions. We observe similar patterns for the Graph Drawing Algorithm, as shown in Figures 4m-4t. In some of the cases, the Graph Drawing Algorithm confused entire regions, either diagonally or circularly, leading to large errors. However, in the cases with many known locations, Figures 4m and 4t, we observe, consistent with the quantitative results in Table 3, that the matching error is vastly reduced. We also see that the Graph Drawing Algorithm with 50 known locations yields the best result overall, which is also consistent with the quantitative results. We note that the optimal matching and the greedy matching yield very similar results. This is expected, as both algorithm start with the same newly generated locations, and they only differ in the heuristic to map new locations to ground truth locations. It seems that the optimal approach yields a slight visible matching improvement, particularly in the cases with ten and 50 known locations, but the differences are rather subtle. Finally, Figures 4w-4z show the result of the Spatial Label Propagation Algorithm. We observe that this algorithm yields the best results with three and ten known locations, particularly avoiding any systematic error that mismatches entire regions. But compared to the Virginia Facebook dataset, the differences are not as distinctive.

5.4 Runtime Analysis

Figure 5 describes the runtime for each algorithm for three populations levels and 50 known locations with the Geosocial Erdős-Rényi network. Each runtime was averaged over 30 trials and the results are displayed on a log scale. As shown in Figure 5, the Partitioning-Based algorithm is the longest to run by far on all population levels. The optimal and greedy versions of the Graph-Drawing algorithm are the next slowest, but they are still substantially faster than the Partitioning-Based Algorithm. The Greedy algorithm is the fastest, but both versions of the Spatial Label Propagation Approach are not considerably slower.

6 Conclusions

This work demonstrates that geosocial networks can be leveraged to estimate users' home locations even when direct location information is withheld. We present and evaluate four algorithms for estimating such links using only social network structure and a sparse set of known user-location anchors. Empirical results on real-world data from VA, USA, reveal that these methods achieve accurate location inference even when only a moderate number of user locations are known. We find that the Spatial Label Propagation Optimal algorithm outperforms all other algorithms in most cases. However, the Graph Drawing Optimal algorithm performs best for smaller populations, as shown in Table 3, where Graph Drawing Optimal outperformed Spatial Label Propagation Optimal for populations of 100 vertices. For cases with zero known locations, where Spatial Label Propagation is not applicable, the Partitioning-Based Algorithm and Greedy Algorithm generate the lowest average distance from assigned location to correct location. Our findings underscore both the utility of such techniques for spatial analysis and their implications for user privacy.

References

[1] K. S. Atwal, T. Anderson, D. Pfoser, and A. Züfle. Predicting building types using openstreetmap. *Scientific Reports*, 12(1):19976, 2022.
[2] L. Backstrom, E. Sun, and C. Marlow. Find me if you can: improving geographical prediction with social and spatial proximity. In *Proceedings of the 19th international conference on World wide web*, pages 61–70, 2010.
[3] M. Bailey, R. Cao, T. Kuchler, J. Stroebel, and A. Wong. Social connectedness: Measurement, determinants, and effects. *Journal of Economic Perspectives*, 32(3):259–280, 2018.
[4] J. Bao, Y. Zheng, and M. F. Mokbel. Location-based and preference-aware recommendation using sparse geo-social networking data. In *ACM SIGSPATIAL'12*, pages 199–208, 2012.
[5] E. Cho, S. A. Myers, and J. Leskovec. Friendship and mobility: user movement in location-based social networks. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1082–1090, 2011.
[6] D. F. Crouse. On implementing 2d rectangular assignment algorithms. *IEEE Transactions on Aerospace and Electronic Systems*, 52(4):1679–1696, 2016.
[7] P. Cui, F. Wang, S. Liu, M. Ou, S. Yang, and L. Sun. Who should share what? item-level social influence prediction for users and posts ranking. In *ACM SIGIR'11*, pages 185–194, 2011.
[8] Y.-A. De Montjoye, C. A. Hidalgo, M. Verleysen, and V. D. Blondel. Unique in the crowd: The privacy bounds of human mobility. *Scientific reports*, 3(1):1376, 2013.

[9] T. Fararo and M. Sunshine. *A Study of a Biased Friendship Net.* 01 1964.
[10] T. M. Fruchterman and E. M. Reingold. Graph drawing by force-directed placement. *Software: Practice and experience*, 21(11):1129–1164, 1991.
[11] K. Gallagher, T. Anderson, A. Crooks, and A. Züfle. Synthetic geosocial network generation. In *7th ACM SIGSPATIAL Workshop on Location-based Recommendations, Geosocial Networks and Geoadvertising*, pages 15–24, 2023.
[12] M. T. Gastner and M. E. Newman. The spatial structure of networks. *The European Physical Journal B-Condensed Matter and Complex Systems*, 49:247–252, 2006.
[13] P. Golle and K. Partridge. On the anonymity of home/work location pairs. In *International Conference on Pervasive Computing*, pages 390–397. Springer, 2009.
[14] M. Haklay and P. Weber. Openstreetmap: User-generated street maps. *IEEE Pervasive computing*, 7(4):12–18, 2008.
[15] D. Jurgens. That's what friends are for: Inferring location in online social media platforms based on social relationships. *Proceedings of the International AAAI Conference on Web and Social Media*, 7(1):273–282, Aug. 2021.
[16] T. Kamada, S. Kawai, et al. An algorithm for drawing general undirected graphs. *Information processing letters*, 31(1):7–15, 1989.
[17] Y. Kang, S. Gao, Y. Liang, M. Li, J. Rao, and J. Kruse. Multiscale dynamic human mobility flow dataset in the us during the covid-19 epidemic. *Scientific data*, 7(1):390, 2020.
[18] G. Karypis and V. Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on scientific Computing*, 20(1):359–392, 1998.
[19] J.-S. Kim, H. Jin, H. Kavak, O. C. Rouly, et al. Location-based social network data generation based on patterns of life. In *MDM*, pages 158–167. IEEE, 2020.
[20] R. Li, S. Wang, H. Deng, R. Wang, and K. C.-C. Chang. Towards social user profiling: unified and discriminative influence model for inferring home locations. In *ACM SIGKDD'12*, pages 1023–1031, 2012.
[21] L. Luceri, T. Braun, and S. Giordano. Analyzing and inferring human real-life behavior through online social networks with social influence deep learning. *Applied network science*, 4(1):1–25, 2019.
[22] M. McPherson, L. Smith-Lovin, and J. M. Cook. Birds of a feather: Homophily in social networks. *Annual review of sociology*, 27(1):415–444, 2001.
[23] J. Moody. Race, school integration, and friendship segregation in america. *American journal of Sociology*, 107(3):679–716, 2001.
[24] T. Pontes, G. Magno, M. Vasconcelos, A. Gupta, J. Almeida, P. Kumaraguru, and V. Almeida. Beware of what you share: Inferring home location in social networks. In *ICDM'12 Workshops*, pages 571–578. IEEE, 2012.
[25] E. Seglem, A. Züfle, J. Stutzki, et al. On privacy in spatio-temporal data: User identification using microblog data. In *SSTD'17*, pages 43–61. Springer, 2017.
[26] R. Shokri, G. Theodorakopoulos, C. Troncoso, J.-P. Hubaux, and J.-Y. Le Boudec. Protecting location privacy: optimal strategy against localization attacks. In *Proceedings of the 2012 ACM conference on Computer and communications security*, pages 617–627, 2012.
[27] Y. Xiao and L. Xiong. Protecting locations with differential privacy under temporal correlations. In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, pages 1298–1309, 2015.
[28] H. Zang and J. Bolot. Anonymization of location data does not work: A large-scale measurement study. In *Proceedings of the 17th annual international conference on Mobile computing and networking*, pages 145–156, 2011.
[29] Y. Zheng. Location-based social networks: Users. In *Computing with spatial trajectories*, pages 243–276. Springer, 2011.