Configuring Networks with Semi-Randomized Location Data

N. Fiore, S. Neumann, G. Moores

May 3, 2025

Abstract

Spatial graphs are increasingly used to model relationships in fields ranging from epidemiology to urban planning. However, such datasets frequently include either overly precise location information that poses privacy risks or incomplete geographic data that hinders meaningful analysis. Existing anonymization techniques often compromise graph-theoretic properties or struggle in regions with atypical population density.

We introduce an algorithm and companion Python package that addresses these challenges by enabling privacy-preserving transformations and uncertainty-aware modeling for spatial networks. Our approach preserves the structural integrity of the original graph while either obfuscating sensitive coordinates or generating plausible spatial configurations from uncertain inputs. This dual functionality supports both the protection of personal or organizational privacy and the analytical macro-analysis of networks from partial data.

Our improvement comes from the new algorithm being region-aware, which preserves the network's core insights while not revealing private information or relying too heavily on known locations. We show that, in most cases, our algorithm perturbs graphs less than the industry standard while making the original location completely unrecoverable.

1 Introduction

In an increasingly data-driven world, appropriate visualization techniques allow for rapid shared understanding. Visualizations offer comprehensive perspectives for decision makers but typically incur a risk of breaching individual privacy in meeting this demand. Privacy-preserving techniques enable creation of visualizations that maintain a desired anonymity of individuals while communicating the broader trends and ideas about a dataset [DA21].

Spatial graph visualizations provide a way to understand interactions and dynamics of entities across a geographic space. Significant privacy concerns arise from the use of precise location data in modeling. Many situations do not require precise geographic locations, and coordinates with uncertainty radii of 10 meters or even 100 meters can suffice. Spatially-anonymized graph visualizations conform to legal and ethical standards without compromising core characteristics of the graph's major features.

In this paper, we present GeoJitter, a lightweight Python tool that works with community standard network tools and GIS data, generating spatially-anonymized representations of underlying networks. Using standard shapefiles as regions, any network with location data may be rendered with randomized locations within their regions. This provides a k-anonymity protection to all nodes in the graph, with k being the smallest population among any region.

Privacy-preserving data publishing [AM21] literature contains many techniques to provide protections on tabular data or charts and a host of techniques considered state-of-the-art for anonymizing spatial attributes. Neither of these provides direct control for the analyst to respect critical characteristics of the spatial graph. Our package allows for region inputs to refine and constrain the randomized points. The key characteristics of this graph are unperturbed in the spatially anonymized revision, providing a true capture of high-level characteristics with strong anonymity protections.

We also discuss current community best practices for privacy-preserving data visualization and formally define the process used for configuring networks with semi-randomized location data. We provide an empirical analysis of results on perturbation of graph characteristics using our tool compared to other community standard practices for spatial anonymity. In closing, we explore considerations for

use of this tool and recommend the public repository to future users to get acquainted with use of the software tool.

We anticipate two classes of use-cases for our tool. The first is the case in which too much information is known and must be obfuscated before release; this could be due to privacy concerns or other criteria. In the case where not enough information is known, and coarse location data can be simulated through multiple trials to analyze structure and other graph characteristics.

2 Background

Privacy-preserving data publishing (PPDP) is a wide body of work with domain-specific tools across many fields. These tools and studies at large seek to provide rigorous methods for selectively protecting personal information in datasets while still maximally communicating other characteristics.

A common assumption is that smoothing or coarsening data makes information more secure, but it is still possible to recover some of the information lost. Brownstein et al. investigate how publicly released, low-resolution disease incidence maps—intended to protect individual privacy—can still be reverse-engineered to approximate the original patient-level geographic data [BCKM06]. The authors introduce an unsupervised classification technique that infers likely point sources of disease cases by analyzing visual patterns in smoothed maps. Despite the aggregation and smoothing (techniques meant to anonymize data), the method from Brownstein et al. can estimate the original geographic coordinates of cases with surprising accuracy.

de Montjoye et al. arrive at a similar conclusion studying human mobility data collected from cell towers [dMHVB13]. They found that 95% of the 1.5 million person data set could be uniquely identified using only four geolocated points. As this finding shows, location data can be incredibly revealing, making robust privacy protection from researchers all the more important.

In her foundational work on k-anonymity, Sweeney demonstrated that seemingly innocuous attributes-such as religion, race, and gender—can uniquely identify a large fraction of individuals when combined [Swe02]. If, for any individual, the data is so obfuscated that the person is indistinguishable from $k-1$ others in the data set, then the data set is k-anonymous.

Ensuring k-anonymity is a challenging task when location data acts as such a highly effective quasi-identifier. Even coarsened location information, such as a home ZIP code or neighborhood, can drastically reduce anonymity when cross-referenced with external datasets. The uniqueness of human movement patterns or residential areas, especially in sparsely populated regions, makes location a potent means of re-identification. While millions of people might share a race, and thousands might share a date of birth, how many share their daily commute?

In a field study, Majeed and Lee note that, while many techniques have been proposed and implemented to preserve individual privacy in multiple disciplines, the risk of re-identification via aggregating datasets remains a concern [AM21]. They also note that the risks surrounding re-identification are higher than historically, as more detailed data unlocks many more methods to do harm, including identity theft, user profiling, and social engineering.

There does not yet exist a definitive method to guarantee spatial anonymization, but some methods do exist to mitigate re-identification risk. In their proceedings on spatial anonymization, the United Nations Statistical Commission observes that census regions vary in resolution depending on geographic region and data source, but are generally considered accurate enough to serve as a basis for anonymization [Com21]. Their proposed method involves aggregating multiple adjacent districts into larger composite areas and reporting only the coarsened geographic information, thereby increasing the uncertainty surrounding any individual's precise location. To satisfy the k-anonymity criterion, regions are combined to create a new region that contains at least k individuals.

Nevertheless, this method exhibits significant limitations in regions with nonuniform population densities. In densely populated urban areas, a single region may already encompass a population substantially larger than k, rendering additional aggregation unnecessary and resulting in degradation of spatial resolution. Conversely, in sparsely populated rural areas, achieving k-anonymity may require the combination of numerous EAs over a large geographic expanse, producing regions so diffuse that the released data loses much of its value for localized spatial analysis.

Radius-based anonymization methods have been proposed to address these challenges, wherein the radius for anonymization is dynamically adjusted according to local population density: smaller radii are employed in densely populated urban centers and larger in sparsely populated regions. Although

such adaptive methods improve the balance between privacy preservation and data utility, they introduce additional complexities. In urban environments, even minimal radii may encompass large populations, leading to the loss of fine-grained spatial detail. In rural contexts, the radii required to satisfy k-anonymity may extend across tens or hundreds of kilometers, resulting once again in overly coarse spatial representations. Furthermore, the implementation of variable-radius approaches often necessitates access to auxiliary demographic information to appropriately calibrate thresholds. This in itself may pose privacy risks if improperly managed.

As we have observed in our review of the literature, location data is a strong method for re-identification of anonymized data, and current methods for spatial anonymization encounter problems in areas with non-uniform population density. Our new method aims to provide an alternate way to anonymize data while working to mitigate deviations introduced by non-uniform population densities.

3 Preliminaries

The primary inputs to our algorithm are a graph and a set of regions. A **graph**, or **network**, is comprised of nodes (vertices) and edges (links). Nodes represent entities and edges represent a relationship between nodes of some kind. To be a spatial graph, nodes must have either a region or a specified location. A **region** is a filled polygon, defined using geolocated points on the polygon's vertices and straight edges between them, forming a closed polygonal chain which eventually returns to the starting vertex. All points in the polygon, including the boundaries, are considered "inside" the region.

To **jitter** or **perturb** a point, we move that point to a new location in the same region it started in. In doing so, we change no other information that node might contain, nor its relationships to any other nodes in the network.

3.1 Radius-Based Spatial Anonymization

The industry standard for spatial anonymization is the radius-based random perturbation method. Based on some maximum radius, a point is uniformly chosen from a circular area centered at the point to be perturbed. To do so, we calculate the values $r = max_r * \sqrt{random(0,1)}$ and $\theta = random(0, 2\pi)$. The point's longitude (horizontal position) and latitude (vertical position) are shifted by $\Delta long = r\cos(\theta)$ and $\Delta lat = r\sin(\theta)$, respectively.

Methods for choosing this maximum radius vary. In our analysis, the maximum radius is defined using this equation:

$$r_{max} = \sqrt{\frac{1}{2\pi N}\sum_{r\in R} r_{area}}$$

Where R is the set of input regions and $N = |R|$. By averaging the areas of the input regions, we get an approximation of how far a jittered point could land from where it started. For fair comparison between our method and the industry standard, we can then calibrate the radius-based method to match. Our measurements of graph perturbation rely heavily on changes to distance between nodes, so we must ensure that these changes will not be skewed by drastically different selection areas.

3.2 Region-Based Spatial Anonymization

Figure 1 shows two networks, with the second having been jittered by our package. Every point maintains its original region in the final network, only in a different position within the region. Furthermore, their connections with one another remain intact, creating a new network while retaining important structural qualities of the original graph.

3.3 Tile-Based Spatial Anonymization

As an intermediate approach, we also created the tile-based method, which considers the total bounding box of the network (that is, the smallest possible rectangle that contains the whole network) and creates a matrix of smaller tiles, arranged in a 10×10 grid. Each point in the network lies in one of thest tiles, so that each tile can be considered its "region." These newly-associated regions and nodes are

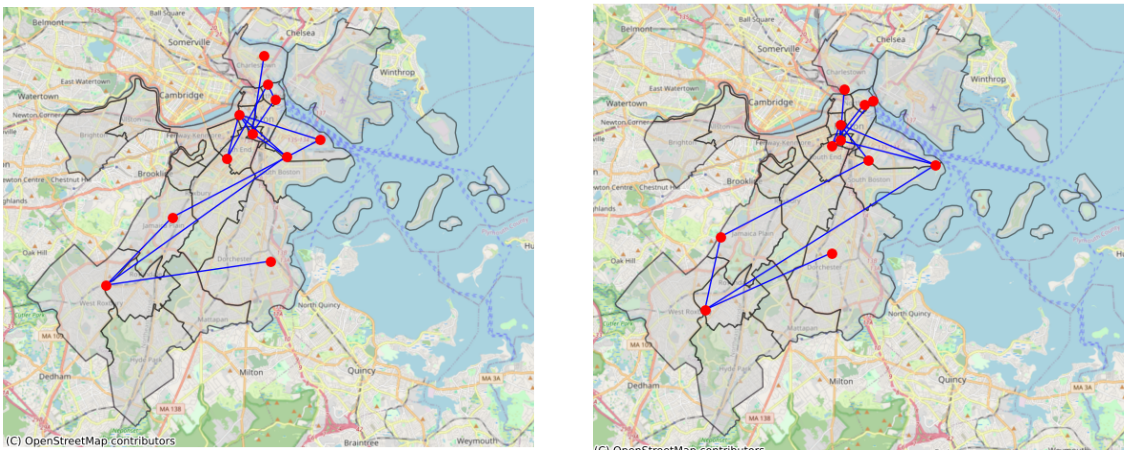

Figure 1: Boston Area Network Pre- and Post-Jitter

then processed using our region-based perturbation algorithm. It should be noted, however, that this method does not account for the average area of possible perturbation, as both the radius- and region-based methods do. In trials where networks span large total areas while containing small regions, poorly-calibrated tiles could lead to much larger deviations than the other methods. Future work on the tile-based approach could alleviate this problem by forcing the height and width of the matrix change such that each tile's area approximates the average area of the input regions.

Utilizing the tile-based method presents two key advantages. First, our algorithm for selecting a new point (Algorithm 1) always succeeds on the first attempt. The bounding boxes of tile-based regions are no different from the tiles themselves. Another advantage is that the tile a point belongs in can be determined analytically in constant time. Rather than the more intensive problem of determining whether a point lies inside a region polygon, the program can simply use the point's coordinates to determine which tile it lies within. Because the tile-based method is somewhat region-aware but is faster computationally, we classify this as the intermediate method between the radius- and region-based methods.

3.4 Measuring Graph Perturbation

We calculate the **Wasserstein distance** between the original network's and anonymized network's edge-length distributions as our primary metric to quantify how much a network changes over the course of a perturbation. As our goal is to preserve as much of the original graph structure as possible while anonymizing the data, our measure of success is a lower Wasserstein distance of the network's edge lengths pre- and post-jitter.

Before jittering, we measure all the edge lengths and create a continuous distribution function (CDF). We normalize the distribution to get CDF_1. We do the same after jittering to get CDF_2. The Wasserstein distance between two CDFs is the area of the space between the two curves [Kan60].

$$W = \int_0^1 CDF_2 - CDF_1 dx \tag{1}$$

The worst case Wasserstein distance is 1. This scenario represents two networks A and B where each has a all edges of a single length (d_A and d_B respectively) and further $d_A \neq d_B$. Anything close to 1 represents networks with edge-length distributions with little or no overlap. Due to the stochastic nature of the jittering, for a sufficiently large network, the probability of such a network arising by chance is vanishingly low. The best-case Wasserstein distance would be 0, meaning the distribution of edge lengths between the two networks is the same.

We also analyze the distributions using the **Kolmogorov-Smirnov test** (KS test or KS distance), which returns the maximum distance between the two distributions. As all quantities are normalized, the KS distance also lies between 0 and 1. The Wasserstein distance focuses on the distribution as a whole, reflecting how well the networks compare to each other in aggregate. The KS test is complementary, ignoring the distribution in aggregate and focusing on the most dramatic difference.

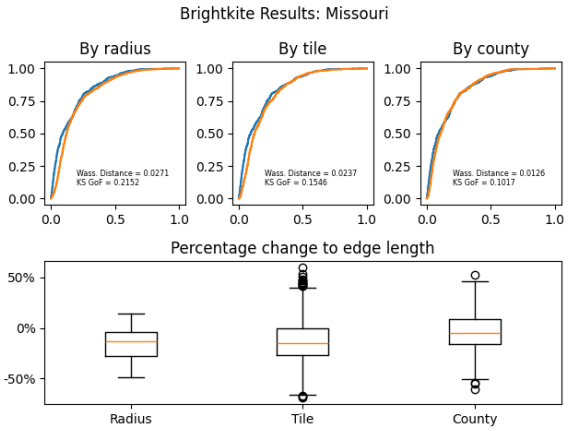

Figure 2: An Example State-Specific Statistical Summary

Taken together, the Wasserstein distance and KS test provide a clear analysis of network differences at large. If both statistics are low, it is a strong sign of success in creating a new, anonymous network without destroying any important information in the process.

As these metrics take cumulative distribution functions as inputs, they are inherently a summary statistics and there is some information lost upon the conversion of a network to a Wasserstein distance. A network which has the same edge length population as the original, but possibly because some got shorter while others got longer, would present as identical to the original network.

Therefore, our last measure of graph perturbation tracks the percentage change of each edge in the network. This provides us with a summary of how each edge is modified. To observe this, we created box-and-whisker plots displaying the percentage change of the edges. At 0%, the edge did not change in length. Though rare to see this exactly, this is where we would like the mean of the distribution to fall. If an edge changed 50%, then its length in the new graph is 1.5 times its length in the old one. Compact distributions centered around 0% signal that the graph was minimally perturbed.

We now have the means to observe the aggregated perturbation of the whole network, the largest deviation in edge lengths, and the distribution of percentage edge length change between the original and jittered networks. A successful jitter keeps the Wasserstein and KS distances low, while keeping the percentage change distribution compact and centered at 0%. Figure 2 provides an example of the figures shown later in the Results section.

4 Methods

To test our algorithm, we use two geolocated networks from the Stanford Network Analysis Program (SNAP) and region boundaries from the United States Census Bureau (USCB) [LK14] [USC]. Both networks from SNAP map users of social media sites (Brightkite and Gowalla) as nodes and connections between those users as edges. We conducted our analysis over all 50 states in the United States, the District of Columbia, and Puerto Rico.

We conducted the analysis using Python and standard scientific libraries, to include `numpy`, `geopandas`, `shapely`, and `scipy`.

For analysis, 25 trials were conducted, and the resulting jittered networks were averaged, using county boundaries from the USCB as the regions within each state [USC]. Networks were filtered to contain only edges where both nodes were in the state. We also removed any nodes without connected edges to create a strongly connected network within each state.

4.1 The Dartboard Method

Before discussing the overall method, we must first establish a necessary subcomponent. Algorithm 2 requires a strategy function as an input, which randomly picks a point within a region; our implementation leverages two algorithms to find such a point. The first is **the Dartboard Method**, which

creates a bounding box around the region and picks a random point inside it. If that point is in the region, it is returned, otherwise, the function loops.

On any given attempt, the Dartboard Method's probability of success is $1 - \frac{region\_area}{box\_area}$. Applied repeatedly, the probability a suitable point is found after x iterations is $(1 - \frac{reg}{box})^x$. The number of executions, therefore, can be calculated using the equation:

$$\log_{\frac{box}{box-reg}}(p),\ p = 0.999$$

where p is the probability of convergence. For example, this implies that a region taking up half its bounding box would find a point in 9 iterations. This method is best-suited for datasets using standard, mostly convex regions, as we can observe most county boundaries are. These regions take up most of their boundary box, allowing the algorithm to find a suitable point in far fewer iterations.

The probability of success is significantly diminished, however, in regions where most of the boundary edges are clustered together, with a small number of far-flung outliers. These outliers increase the size of the bounding box, dramatically decreasing the proportional size the region occupies. This is the reason we include a secondary method, one which is much slower but guarantees success.

If a maximum number of iterations is reached (default 50) without the selection of a suitable point, we use the `triangulate` function from the `triangle` library which first, randomly selects a triangle based on the weighted average of their areas, and second, chooses a random point within that triangle using a known algorithm [She96]. This method is slower by a factor of s (the maximum number of sides in all the regions), but is guaranteed to work. No development trials to date on USCB data have required more than 50 attempts to generate. Algorithm 1 outlines both methods and how they interact with one another.

4.2 Region-Aware Perturbation

Our full algorithm, outlined in Algorithm 2, runs in $O(ns^2)$ time, meaning the runtime scales with respect to the number of input nodes (n) and highest number of sides in all the regions (s), squared. This runtime increases by a factor of the number of regions (r), however, if each point's region is not initially known. Some of the program inputs are themselves functions which allow the algorithm to translate data from its native form into a form it can work with. For these functions, for the purposes of our runtime analysis, we will use the defaults provided in the library and their associated runtimes.

Algorithm 1 Dartboard Method

function RANDOM_POINT_IN_TRIANGLE(**triangle**) $O(1)$
  Let $\vec{a}$, $\vec{b}$, and $\vec{c}$ be the vertices of the input **triangle** in vector form
  Let r_1 and r_2 be two independent, random numbers
  $u = 1 - \sqrt{r_1}$
  $v = \sqrt{r_1} - (1 - r_2)$
  $w = r_2\sqrt{r_1}$
  return $u\vec{a} + v\vec{b} + w\vec{c}$
end function

function TRIANGLE_AREA(**triangle**) $O(1)$
  Let $\vec{a}$, $\vec{b}$, and $\vec{c}$ be the vertices of the input **triangle** in vector form
  return $0.5 * (\vec{b} - \vec{a}) \times (\vec{c} - \vec{a})$
end function

function POINT_GENERATOR(**point**, **region**) $O(s^2)$
  Let *box* represent the **region**'s bounding box (the smallest rectangle that covers the whole region). Its sides are represented using min_x, min_y, max_x, and max_y
  for $\_ \in \{0 \ldots \textbf{max\_iter}\}$ **do**
    Let c_x be a random number between min_x and max_x
    Let c_y be a random number between min_y and max_y
    The candidate point is the ordered pair (c_x, c_y).
    ▷ Note: to check if a point is in a region takes $O(s)$ time[1]
    if $(c_x, c_y) \in$ **region then** $O(s)$
  return (c_x, c_y)
    end if
    ▷ Switch to deterministic, but slower method...
    Let *triangulated* be the triangulated **region**[2]. $O(s^2)$
    Let *weights* be the percentage of the total area made up of each triangle.

    Let the *chosen_tri* be a random choice from *triangulated* using *weights*.
  return RANDOM_POINT_IN_TRIANGLE(*chosen_tri*)
  end for
end function

[1]To check if a point is in a region requires solving the Point-In-Polygon problem; we use the method outlined by Sutherland et al [SSS74]. [2]Triangulated leverages algorithm from [She96].

Algorithm 2 Obfuscate Network

function *obfuscate_network*(**original_net**, **regions**, **point_access**, **region_access**, **strat**)
  An empty mapping *new_nodes* is created, which will later be added to a graph.
  for *point* $\in$ **original_net**.points **do** $O(ns^2)$, see [3]
    new_point = **strat**(*point.coords*, *region*(*point*)) $O(s^2)$
    new_nodes $\leftarrow$ *new_point*
  end for
  new_net.add(*new_nodes*) $O(n)$
  new_net.add(*original_net.edges*) $O(e)$
  return *new_net*
end function

[3]The runtime of this algorithm increases by a factor of r when the points do not already have a region associated with them.

5 Results

We have found that our method of region-aware perturbation generally matches the existing method of spatial anonymization studied (radius-based perturbation), with both these methods narrowly outperforming the tile-based method. Though individual edges themselves, when examined before and after, can display larger perturbations for the region-aware case, these changes in aggregate form similar CDFs to the radius-based CDFs.

We tested all methods against two data sets: Brightkite and Gowalla, both from SNAP [LK14]. Both data sets are derived from geolocated social media applications, though Brightkite has since shut down. The Brightkite dataset contains $58,228$ nodes and $214,078$ edges, while the Gowalla dataset contains $196,591$ nodes and $950,327$ edges. As expected based on our runtime analysis of Algorithm 2, a trial using Gowalla takes approximately 4 times as long as a trial using Brightkite, mirroring the theoretical linear scale factor introduced by the number of nodes n.

5.1 Aggregated Results: United States

We focused on the percentage change in edge length as our key measure of graph perturbation throughout this paper. After conducting 25 trials over the 50 states in the United States (plus other areas such as the District of Columbia and Puerto Rico), we aggregated the resulting Wasserstein and KS distances, differentiating each data point by state and which technique was used to generate it. Overall, we found that all three techniques perform well by our metric of edge length perturbation, maintaining generally small Wasserstein and KS distances across states with diverse overall sizes, county sizes, and localized network cardinality. Any of the three techniques would be an effective way to anonymize a spatial dataset while preserving key characteristics of the network.

We illustrate the average Wasserstein distances of each state in Figure 3 colored by technique. The most important insight is that all the Wasserstein distances are exceedingly low, with the majority being under 0.05, implying that the distributions of edge lengths is left almost intact by all three perturbation methods. Of the three, however, the radius-based method generally performs the best, followed closely by the region-based and tile-based perturbation techniques. Of the outliers, the radius-based method has the fewest, yet most pronounced, while the tile-based and region-based methods have more frequent, less noticeable outliers.

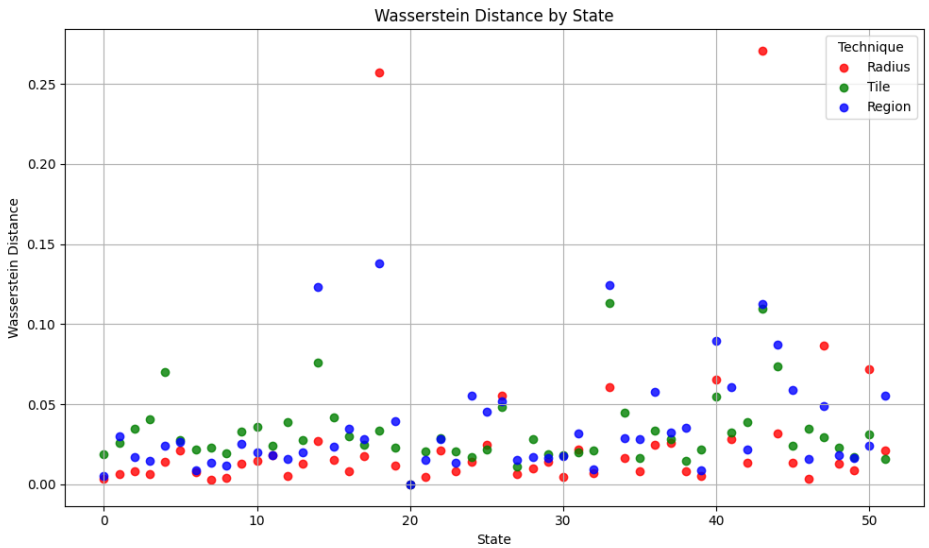

Figure 3: Wasserstein Distances by State and Technique

Applying the same analysis to the KS Distance of each of the distributions yields similar results. Here, we see the maximum vertical difference between the two distributions. Again, we see relatively consistent performance from all three techniques, with the radius-based method generally having the smallest KS Distance, followed closely by region-based perturbation. The tile-based method is more

erratic, generally having the highest KS Distance of the three, though further analysis is needed to rule out the non-normalized tile area as a factor in this deviation.

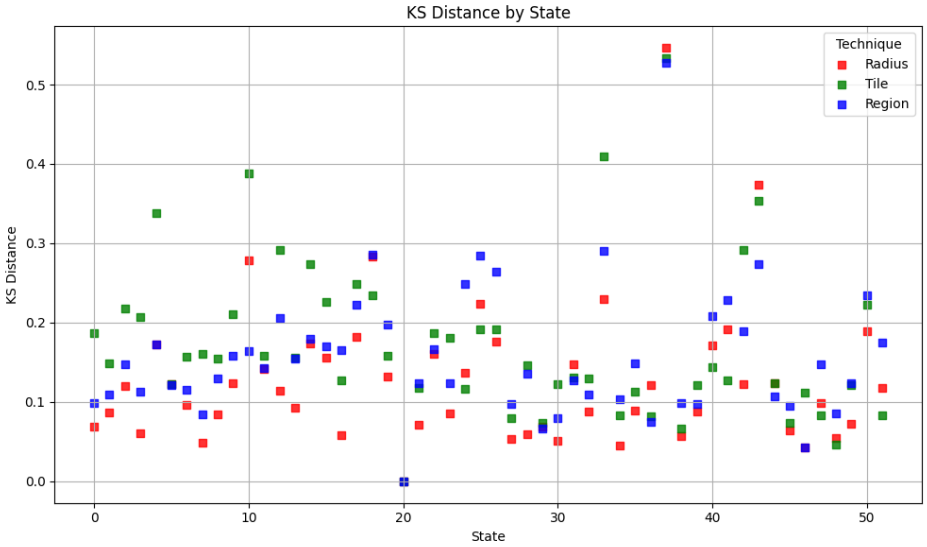

Figure 4: KS Distances by State and Technique

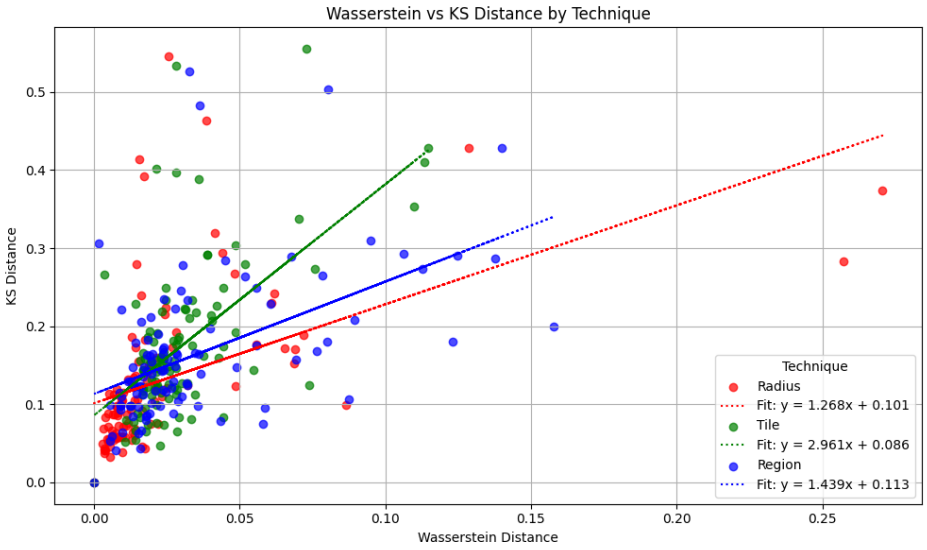

Figure 5: Wasserstein Distance vs. KS Distance, colored by Technique

5.2 Selected Results

The charts below summarize the data on a state-by-state basis, with descriptions noting key insights. We have selected four states we consider to offer the most insight into the performance of our algorithm. The first three: California, Kansas, and Texas, had the lowest Wasserstein distances across both data sets, implying that these three states had the best results after our algorithm perturbed the networks. The last, Alaska, had the highest KS Distance, which represents the greatest deviation between the starting and ending networks.

For each of the state-specific plots, we include representations of each technique's CDF with an accompanying box-and-whisker plot showing the percentage deviation of particular edges before and after the perturbation.

5.2.1 California

Figure 6 shows the results using the Brightkite and Gowalla data California subgraphs. We see in both

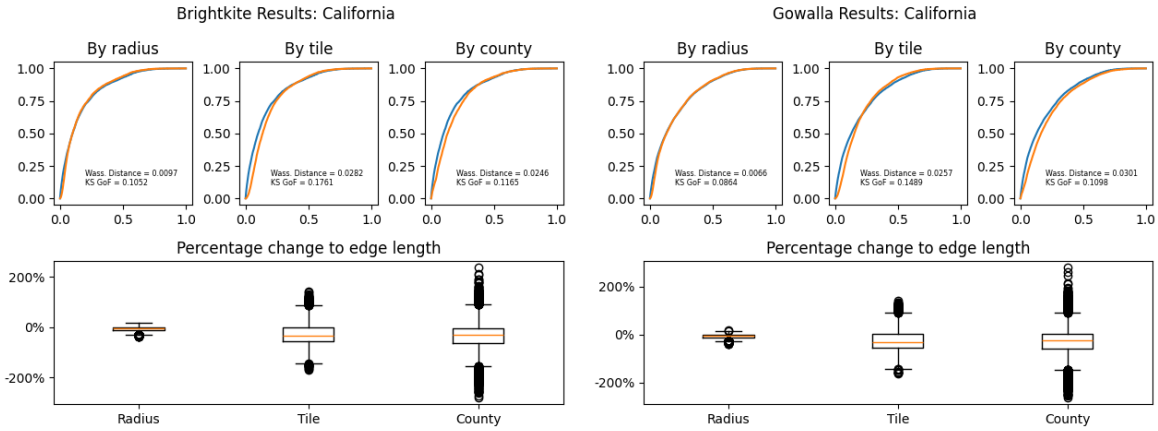

Figure 6: Performance Metrics for California, US

datasets that the radius-based method outperforms our method by a relatively wide margin, while also keeping perturbations minor in comparison to both geography-based methods. We theorize that this is due to California's numerous, comparatively small counties surrounding its urban centers (see Figure 7). These counties bring down the average area, which in turn brings down the maximum radius used to jitter the points. Conversely, the larger rural counties present a much larger area for a region-based jittered point to land, far further than the maximum radius would allow.

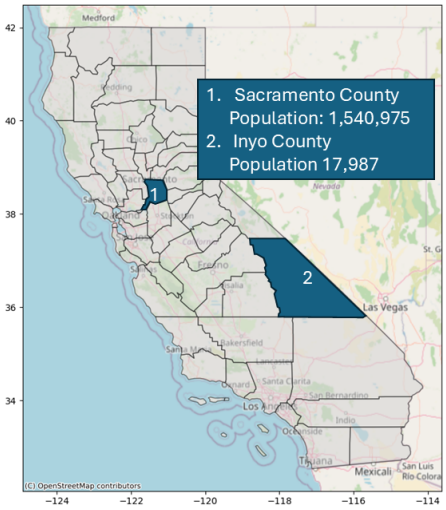

Figure 7: County Map of California, US with Selected Populations

5.2.2 Kansas

In Kansas we see very accurate results from both the radius-based and region-based methods, though the tile-based method falls short. For the radius-based and region-based perturbations, the distributions of percentage edges changes both lie near the ±50% mark, with their means falling very close to 0%. The Wasserstein and KS Distances are some of the lowest recorded of all the states, both at 0.0055. Furthermore, the maximum deviations, shown in the KS Distance, are both near 0.05.

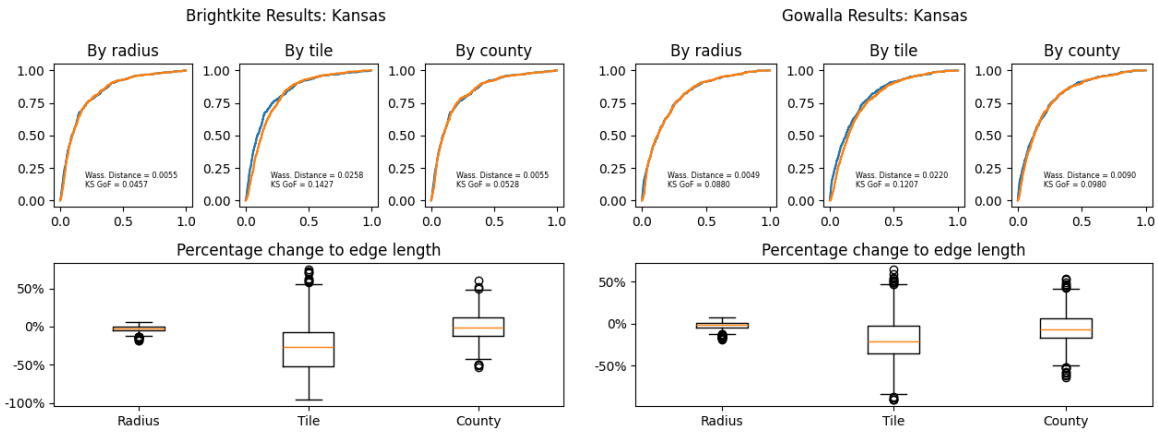

Figure 8: Performance Metrics for Kansas, US

5.2.3 Texas

Similarly to both California and Kansas, Texas displays good results for the radius-based and region-based methods, with suboptimal results for the tile-based method. The box-and-whisker plot's scale is somewhat skewed by how wide the margin is for the tile-based method, but we do observe similar results to Kansas, where the radius method is highly compact and symmetrical, with the region method less compact, but still symmetrical and centered at 0%.

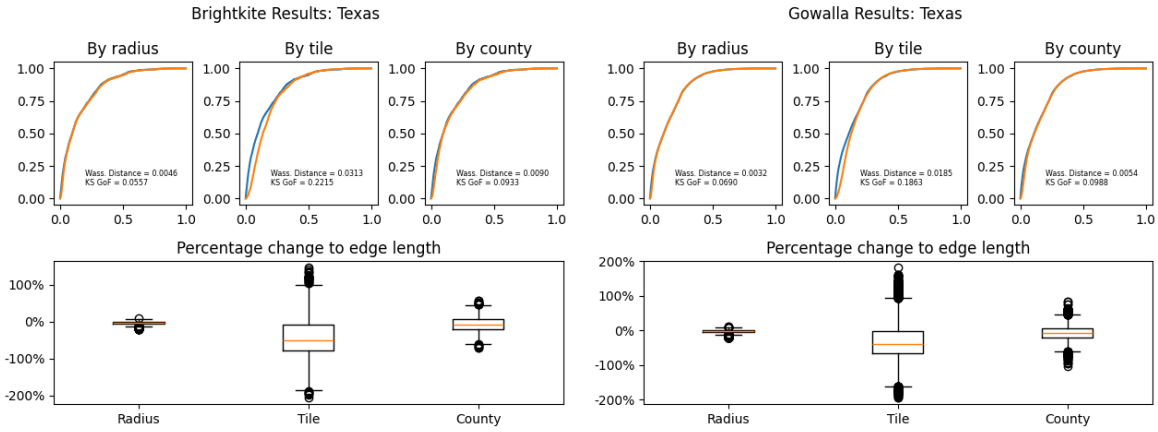

Figure 9: Performance Metrics for Texas, US

5.2.4 Alaska

The case study of Alaska exposes the shortcomings of our method, with the most pronounced failures stemming from sparsely populated regions. There are not many nodes from Alaska in either dataset, leading to the jagged CDFs and unusual box-and-whisker plots. We believe that the combination of sparse data and large counties in Alaska contributed to these outlier results. Furthermore, most of Alaska's population cluster near the coasts, even those nodes in inland counties. After jittering, the majority of these points traveled further inland into previously uninhabited areas. For situations like Alaska's, we recommend inputting smaller regions into the algorithm to prevent points from being jittered too far.

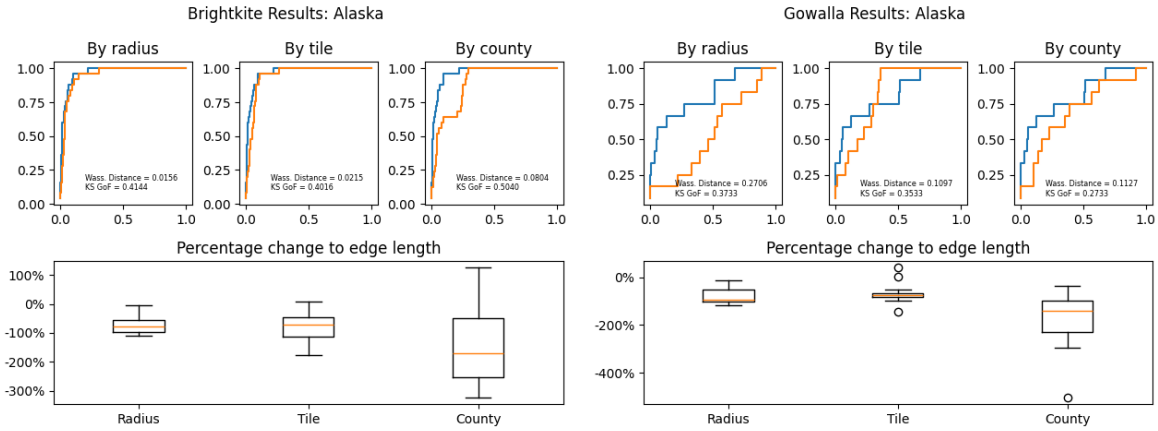

Figure 10: Performance Metrics for Alaska, US

6 Applications

We anticipate two primary scenarios in which our package will prove especially useful during the data analysis process, each representing opposite ends of the information spectrum. In the first, too much sensitive information is available: datasets may contain precise geographic coordinates or identifiable attributes that could compromise individual or organizational privacy if released unmodified. In such cases, our package enables privacy-preserving transformations that retain the structure of the underlying network while obscuring sensitive location data. In the second scenario, too little information is known: analysts may only have access to simulated, approximate, or uncertain location data, often in the form of bounded regions or probabilistic estimates. Here, our package provides tools for constructing and working with spatial graphs in the presence of location uncertainty, allowing meaningful analysis even when exact positions are unavailable.

6.1 Protecting Personal Privacy

Despite the many practices ethical researchers can use to anonymize data, any exact geographic coordinates can pose a risk to participants in a study. Even in the absence of explicit identifiers such as names, quasi-identifiers (e.g. height, weight, gender, or ethnicity) can be cross-referenced with external datasets to enable re-identification. This process, known as a linkage attack, is particularly concerning when multiple datasets are individually innocuous but collectively compromising [VRCR20].

While no anonymization strategy can eliminate this risk entirely, our method mitigates the most severe threats by obfuscating spatial coordinates in a way that preserves relational structure without exposing original locations. In the unfortunate event a linkage attack can connect anonymized entries to one another, precise geographic positions still remain unrecoverable. This allows researchers to retain meaningful insights from the spatial relationships in the data while substantially reducing the risk of exposing sensitive personal information.

This approach is especially valuable in high-risk domains such as public health, urban mobility, and social network analysis, where fine-grained location data is available for research but poses ethical and legal challenges when shared. By anonymizing location data without altering the underlying graph structure, our package helps ensure that datasets remain both useful and compliant with modern data protection standards like the European Union's General Data Protection Regulation (GDPR) and the United States' Health Insurance Portability and Accountability Act (HIPAA) [EU] [HIP].

6.2 Working With Uncertainty

In scenarios such as disease transmission modeling, military logistics, and search and rescue operations, the exact location of a key node—whether Patient Zero, enemy combatants, or a missing individual—is often unknown. Instead, analysts work with zones of probable location, relying on partial or indirect evidence to infer spatial distributions. When multiple entities are involved, and their interactions form

a network, this uncertainty quickly compounds, which can render conventional spatial graph analysis unreliable or misleading.

Our library provides tools to formally incorporate spatial uncertainty into network representations. Rather than requiring exact coordinates, it allows users to define uncertainty regions associated with each node. This flexibility enables the analysis of spatial graphs that reflect the imprecise nature of the underlying data without discarding its relational structure. Analysts can perform simulations, measure connectivity, or run optimization tasks while respecting the spatial fuzziness inherent to the input.

This capability is particularly useful in time-critical or low-visibility environments where collecting exact location data is infeasible or unsafe. For example, in outbreak modeling, early case reports may come with neighborhood-level granularity rather than specific addresses. In military contexts, known movements may be intercepted or delayed, requiring reasoning over probable rather than known positions. By natively supporting spatially uncertain inputs, our package enhances the reliability of downstream analysis while preserving the analytical rigor of graph-based methods.

7 Conclusions and Future Work

In this paper, we presented a novel method for perturbing spatial network data while preserving other important qualities of the network, such as average edge length, region-specificity, and node connections. Our new method builds upon the state-of-the-art anonymization methods in the data analytics industry while accounting for geographic boundaries that influence human behavior, such as county lines.

Our method allows for the researcher to specify boundaries a jittered point should remain within after it is moved, ensuring insights drawn from a point's approximate location can still be gleaned. We showed that the algorithm is lightweight and handles networks with hundreds of thousands of nodes with ease. By focusing on transformations that retain regional relationships when obfuscating location data, we provide a practical solution for privacy preservation and modeling uncertain information.

We theorize use-cases in both PPDP and modeling under uncertain conditions. In the first case, we create a data set where a linkage attack might rediscover a person's identity, but would never be able to recover their original location. Data can be transferred between organizations and the public without the risk of a high-profile data breach. In the second case, we allow for uncertain, granular data to be modeled as if exact positions were known, allowing the researcher to utilize techniques meant for specific locational data. Over enough trials, broad trends can be inferred from previously inscrutable information.

As development of this project and similar continue, we anticipate future work will focus on improving the speed of our algorithm, enabling multi-threading and thread-safe visualization, and determining the threshold at which the radius method outperforms the region method, if indeed such a threshold exists.

References

[AM21] S. Lee A. Majeed. Anonymization techniques for privacy preserving data publishing: A comprehensive survey. *IEEE Access*, 2021.

[BCKM06] John S. Brownstein, Christopher A. Cassa, Isaac S. Kohane, and Kenneth D. Mandl. An unsupervised classification method for inferring original case locations from low-resolution disease maps. *International Journal of Health Geographics*, 5(1):56, 2006.

[Com21] United Nations Statistical Commission. Spacial anonymization. In *Statistical Commission: Fifty-second session*, 2021.

[DA21] R. Wilson et al. D. Avraam, O. Butters. Privacy preserving data visualizations. *EPJ Data Science*, 2021.

[dMHVB13] Yves-Alexandre de Montjoye, César A. Hidalgo, Michel Verleysen, and Vincent D. Blondel. Unique in the crowd: The privacy bounds of human mobility. *Scientific Reports*, 3:1376, 2013.

[EU] The European Union's General Data Protection Regulation. https://gdpr-info.eu/.

[HIP] The United States' Health Insurance Portability and Accountability Act. https://www.hhs.gov/hipaa/for-professionals/privacy/laws-regulations/combined-regulation-text/index.html.

[Kan60] L. V. Kantorovich. Mathematical methods of organizing and planning production. *Management Science*, 6(4):366–422, 1960.

[LK14] Jure Leskovec and Andrej Krevl. SNAP Datasets: Stanford large network dataset collection. http://snap.stanford.edu/data, June 2014.

[She96] Jonathan Richard Shewchuk. Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator. In Ming C. Lin and Dinesh Manocha, editors, *Applied Computational Geometry: Towards Geometric Engineering*, volume 1148 of *Lecture Notes in Computer Science*, pages 203–222. Springer-Verlag, May 1996. From the First ACM Workshop on Applied Computational Geometry.

[SSS74] Ivan E. Sutherland, Robert F. Sproull, and Robert A. Schumacker. A characterization of ten hidden-surface algorithms. *ACM Comput. Surv.*, 6(1):1–55, March 1974.

[Swe02] Latanya Sweeney. k-anonymity: A model for protecting privacy. *International Journal on Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(5):557–570, 2002.

[USC] Data from the United States Census Bureau used for the procurement of geographic boundaries of US States and Counties. https://www.census.gov/geographies/mapping-files/time-series/geo/cartographic-boundary.html.

[VRCR20] A. Vidanage, T. Ranbaduge, P. Christen, and S. Randall. A privacy attack on multiple dynamic match-key based privacy-preserving record linkage. *International Journal of Population Data Science*, 5(1):1345, 2020.