The Journal of Open Source Software

PySGN: A Python package for constructing synthetic geospatial networks

Boyu Wang [1¶], **Andrew Crooks** [1], **Taylor Anderson** [2], and **Andreas Züfle** [3]

1 Department of Geography, University at Buffalo, Buffalo, New York, USA **2** Geography & Geoinformation Science Department, George Mason University, Fairfax, Virginia, USA **3** Department of Computer Science, Emory University, Atlanta, Georgia, USA ¶ Corresponding author

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor: James Gaboardi

Reviewers:

- @tomalrussell
- @gegen07

Submitted: 05 June 2025
Published: unpublished

DRAFT

Summary

Synthetic networks are commonly used to study the structure and dynamics of social systems, transportation infrastructure and other complex phenomena. Classical random graph models, such as the Erdős-Rényi, Watts-Strogatz and Barabási-Albert models, generate abstract networks with different structural characteristics: the Erdős-Rényi model connects nodes at random with equal probability, the Watts-Strogatz model rewires a ring lattice to produce small-world networks, and the Barabási-Albert model yields scale-free networks through preferential attachment (Barabási & Albert, 1999; Erdös & Rényi, 1960; Watts & Strogatz, 1998). In their standard form these models ignore the spatial positions of nodes; yet in many empirical settings (e.g., human social networks, commuting patterns or infrastructure networks) proximity strongly influences who connects to whom.

PySGN (**Py**thon for **S**ynthetic **G**eospatial **N**etworks) is an open-source Python package that extends the classical models to geospatial contexts. It embeds nodes in geographic coordinate space, modifies connection rules to decay with distance, and allows users to incorporate clustering and preferential attachment while respecting spatial constraints. By combining these ingredients, PySGN generates synthetic geosocial networks that mimic the spatial relationships observed in real-world networks. The package integrates with the PyData ecosystem through libraries like GeoPandas (Bossche et al., 2024) and NetworkX (Hagberg et al., 2007) and provides a flexible interface for research and simulation.

With PySGN, users can specify parameters such as average degree and decay function among other options, and construct a geospatially embedded network as NetworkX graph objects. They can also fine-tune the generation process by defining custom distance functions or constraints. The resulting networks can be further analyzed and visualized thereafter. The package is intended for researchers and practitioners in fields such as urban planning, epidemiology, infrastructure resilience and social science who require robust tools for simulating and analyzing complex geospatial networks.

Statement of Need

The need for synthetic geospatial networks arises from their utility in social simulations, including modeling of transportation systems, pedestrian movements, and the spread of infectious diseases (Züfle et al., 2024). Traditional synthetic populations often lack the integration of geographic social networks, which are crucial for accurately capturing social connections and spatial dynamics, to explore the effects of spatial proximity on social interactions, mobility patterns, and network robustness (Jiang et al., 2024).

PySGN addresses this gap by providing a tool that not only generates geographically explicit networks but also incorporates key network properties, such as clustering, preferential attachment and spatial decay. These features allow users to explore different network properties and configurations (e.g., average node degree). This is essential for a variety of simulation scenarios, where understanding spatial relationships and social dynamics is critical for analyzing and modeling complex systems. This makes PySGN suitable for diverse applications, including infrastructure resilience studies, agent-based modeling, and geospatial data analysis.

Related Work

There are some Python packages that work with spatial networks but their focus is primarily on processing or downloading real-world street data rather than generating new networks. For example, `neatnet` simplifies complex street geometries - such as dual carriageways and roundabouts - so that analysts can work with cleaner, more interpretable road layouts (Fleischmann et al., 2026). OSMnx provides tools to download, analyze and visualize street networks and other features from OpenStreetMap with a single command (Boeing, 2025). Both of these libraries operate on existing network data, however, neither offers a way to create synthetic networks or adjust the underlying spatial distribution of nodes.

Within the PySAL ecosystem, `spaghetti` is an open-source library for analyzing network-based spatial data. It originated from PySAL's network module and provides tools to build network objects from line geometries and to analyze network-constrained events along those networks (Gaboardi et al., 2021). In parallel, PySAL has introduced `libpysal.graph`, an experimental representation for spatial weights matrices (intended for spatial-statistical workflows rather than network-event analysis) (Rey & Anselin, 2007). These tools support analysis and representation of existing spatial relationships and networks, but they do not provide generative models for synthetic geospatial network construction, which is the focus of PySGN.

General-purpose network libraries such as NetworkX and `igraph` include random graph generators. NetworkX offers functions to generate Erdős-Rényi, Watts-Strogatz and Barabási-Albert graphs (Hagberg et al., 2007), and `igraph` includes similar random graph models (Barabási-Albert, Erdős-Rényi, Watts-Strogatz and other stochastic graph models) (Csárdi & Nepusz, 2006). However, these models do not incorporate geographic information or spatial distances: nodes are considered abstract or are uniformly distributed, and edge probabilities do not depend on spatial proximity. Consequently, while NetworkX and igraph are excellent tools for general network analysis and for generating abstract random graphs, they are unsuitable for constructing geospatial networks where spatial proximity influences connectivity (Gallagher et al., 2023).

By contrast, PySGN synthesizes geospatial networks by embedding nodes in geographic coordinate space and incorporating distance-decay functions and other constraints into the generation process. It extends the classical random graph models (Erdős-Rényi, Watts-Strogatz and Barabási-Albert) to spatial contexts and integrates with GeoPandas and NetworkX to provide geospatially explicit network generation and analysis. Thus, PySGN fills a gap between packages that simplify or analyze existing spatial networks and those that generate abstract random graphs.

Example

To illustrate how spatial constraints alter both the structure of a network and the distribution of connection distances, Figure 1 contrasts a **geospatial Watts-Strogatz network** with its **classic Watts-Strogatz** counterpart. In this demonstration we placed 120 nodes uniformly at random in a unit square. For the spatial model we used the `geo_watts_strogatz_network()` function provided by PySGN with parameters `k=6` and `p=0.1`. This model first connects each node to its six nearest neighbours and then rewires edges according to a distance-dependent rule: existing edges are rewired with probability p, but new targets are chosen with a probability that decays exponentially with Euclidean distance. Nearby nodes are therefore much more likely to be connected than distant nodes. For comparison we generated a classic Watts-Strogatz network with the same values of n, k and p using NetworkX's `watts_strogatz_graph()`. In the classic formulation, rewired edges connect to a **uniformly random node**, which ignores the geometry of the embedding and often results in long edges that criss-cross the domain.

The top row of Figure 1 shows the resulting networks. The geospatial Watts-Strogatz network (right) features mostly short edges and tightly clustered neighbourhoods, whereas the classic network (left) includes many long-range shortcuts, producing a tangled appearance. The bottom row plots histograms of the Euclidean lengths of all edges. Because edges in the classic model are rewired without regard to distance, the lengths span the full range of possible distances between random points in the unit square; by contrast, the geospatial model's distance-decay function leads to a sharp peak at short distances and very few long edges. Together, these panels demonstrate how incorporating geography into a Watts-Strogatz model yields networks that better reflect the localised interactions observed in real-world systems. For more elaborate examples based on empirical geospatial networks, see Züfle et al. (Züfle et al., 2024) or Gastner and Newman (Gastner & Newman, 2006).

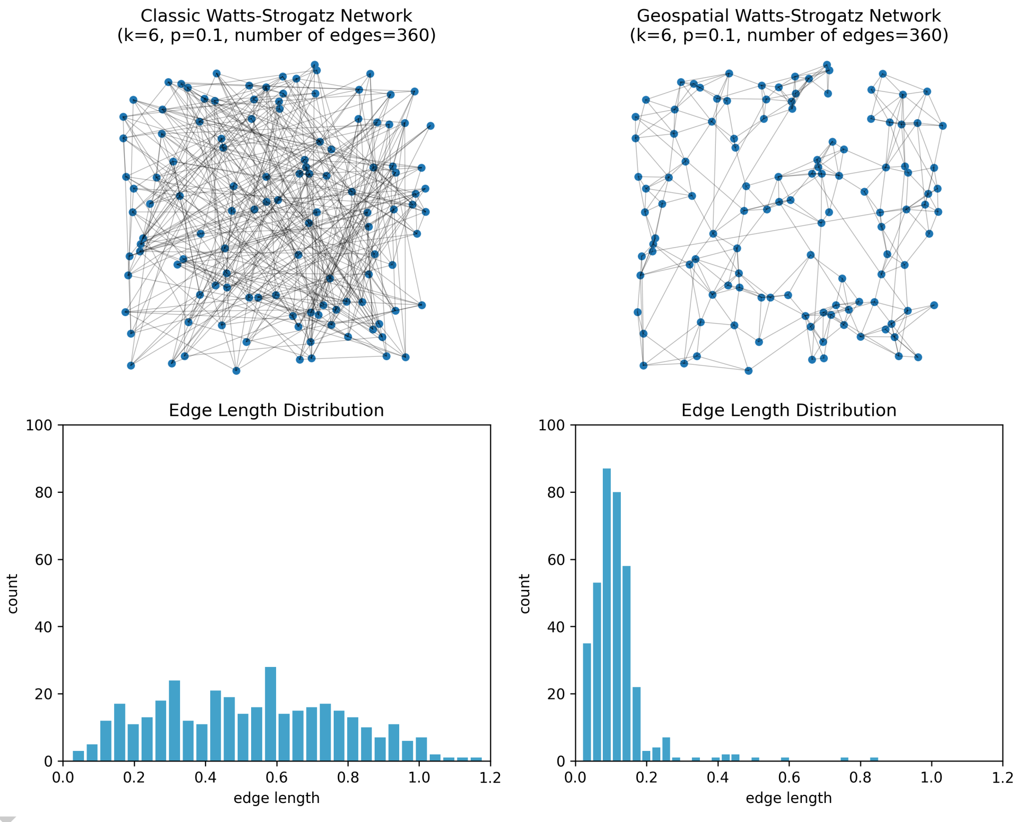

Figure 1: Comparison of classic (left) and geospatial (right) Watts-Strogatz networks (top row) and histograms of edge lengths (bottom row) for 120 nodes with k = 6 and p = 0.1.

Acknowledgements

The algorithms implemented in PySGN are based on the work of Alizadeh et al. (2017) and Gallagher et al. (2023), with several improvements and modifications, including bug fixes, performance enhancements, and additional features. We would like to thank the authors for their contributions to the field of synthetic geospatial network generation.

This work was supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior/Interior Business Center (DOI/IBC) contract number 140D0423C0025. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DOI/IBC, or the U.S. Government.

References

Alizadeh, M., Cioffi-Revilla, C., & Crooks, A. (2017). Generating and analyzing spatial social networks. *Computational and Mathematical Organization Theory*, *23*, 362–390. https://doi.org/10.1007/s10588-016-9232-2

Barabási, A.-L., & Albert, R. (1999). Emergence of Scaling in Random Networks. *Science*, *286*(5439), 509–512. https://doi.org/10.1126/science.286.5439.509

Boeing, G. (2025). Modeling and Analyzing Urban Networks and Amenities With OSMnx. *Geographical Analysis*, *57*(4), 567–577. https://doi.org/10.1111/gean.70009

Bossche, J. V. den, Jordahl, K., Fleischmann, M., Richards, M., McBride, J., Wasserman, J., Badaracco, A. G., Snow, A. D., Ward, B., Tratner, J., Gerard, J., Perry, M., cjqf, Hjelle, G. A., Taves, M., Hoeven, E. ter, Cochran, M., Bell, R., rraymondgh, … Gardiner, J. (2024). *geopandas/geopandas: v1.0.1* (Version v1.0.1). Zenodo. https://doi.org/10.5281/zenodo.12625316

Csárdi, G., & Nepusz, T. (2006). The igraph software package for complex network research. *InterJournal Complex Systems*, *1695*, 1–9.

Erdös, P., & Rényi, A. (1960). On the evolution of random graphs. In *Publ. math. inst. hung. acad. sci* (Vol. 5, pp. 17–60).

Fleischmann, M., Vybornova, A., Gaboardi, J. D., Brázdová, A., & Dančejová, D. (2026). Adaptive continuity-preserving simplification of street networks. *Computers, Environment and Urban Systems*, *123*, 102354. https://doi.org/10.1016/j.compenvurbsys.2025.102354

Gaboardi, J. D., Rey, S., & Lumnitz, S. (2021). spaghetti: spatial network analysis in PySAL. *Journal of Open Source Software*, *6*(62), 2826. https://doi.org/10.21105/joss.02826

Gallagher, K., Anderson, T., Crooks, A., & Züfle, A. (2023). Synthetic Geosocial Network Generation. *Proceedings of the 7th ACM SIGSPATIAL Workshop on Location-Based Recommendations, Geosocial Networks and Geoadvertising*, 15–24. https://doi.org/10.1145/3615896.3628345

Gastner, M. T., & Newman, M. E. J. (2006). The spatial structure of networks. *The European Physical Journal B - Condensed Matter and Complex Systems*, *49*(2), 247–252. https://doi.org/10.1140/epjb/e2006-00046-8

Hagberg, A., Swart, P. J., & Schult, D. A. (2007). *Exploring Network Structure, Dynamics, and Function using NetworkX*. Los Alamos National Laboratory (LANL). https://doi.org/10.25080/tcwv9851

Jiang, N., Yin, F., Wang, B., & Crooks, A. T. (2024). A Large-Scale Geographically Explicit Synthetic Population with Social Networks for the United States. *Scientific Data*, *11*(1), 1204. https://doi.org/10.1038/s41597-024-03970-1

Rey, S. J., & Anselin, L. (2007). PySAL: A Python Library of Spatial Analytical Methods. *The Review of Regional Studies*, *37*(1), 5–27. https://doi.org/10.52324/001c.8285

Watts, D. J., & Strogatz, S. H. (1998). Collective dynamics of 'small-world' networks. *Nature*, *393*(6684), 440–442. https://doi.org/10.1038/30918

Züfle, A., Pfoser, D., Wenk, C., Crooks, A., Kavak, H., Anderson, T., Kim, J.-S., Holt, N., & Diantonio, A. (2024). In Silico Human Mobility Data Science: Leveraging Massive Simulated Mobility Data (Vision Paper). *ACM Transactions on Spatial Algorithms and Systems*, *10*(2), 1–27. https://doi.org/10.1145/3672557