SUMO User Conference 2025

Conference paper

https://doi.org/10.52825/xxxx........ DOI placeholder (WILL BE FILLED IN BY TIB Open Publishing)

© Authors. This work is licensed under a [Creative Commons Attribution 4.0 International License](https://creativecommons.org/licenses/by/4.0/)

Published: (WILL BE FILLED IN BY TIB Open Publishing)

utdf2gmns: An Open-Source Python Implementation for Converting Synchro UTDF with Signalized Intersections to SUMO using GMNS Standard

Xiangyong Luo1[](https://orcid.org/0000-1111-2222-3333)https://orcid.org/0009-0003-1290-9983, Yiran Zhang3, Guanhao Xu1, Wan Li1, Ross Wang1[](https://orcid.org/0000-1111-2222-3333)https://orcid.org/0000-0001-8073-7683, Jeff Ban (TBD), and Xuesong Simon Zhou2[](https://orcid.org/0000-1111-2222-3333)https://orcid.org/0000-0002-9963-5369

1 Oak Ridge National Laboratory, US

2 Arizona State University, US

3 University of Washington, US

\*Correspondence: Xiangyong Luo, luox1@ornl.gov

**Abstract.** Modern transportation research relies on seamlessly integrating traffic signal data with robust network representation and simulation tools. We present utdf2gmns, an open-source Python-based tool that converts Synchro Universal Traffic Data Format (UTDF) files, including signalized intersections, into the widely adopted General Modeling Network Specification (GMNS). The resulting GMNS network is then for simulation in SUMO (Simulation of Urban Mobility). By automating the extraction of intersection control parameters and aligning them with GMNS conventions, utdf2gmns minimizes manual efforts and data loss. This streamlined workflow empowers researchers and practitioners to build accurate network models, test scenarios more efficiently, and maintain data consistency across platforms. Validation with case studies confirms the tool’s reliability in modeling complex corridors with multiple signalized intersections. As a free and open-source solution, utdf2gmns promotes reproducibility and collaboration in the transportation community. Comprehensive documentation and tutorials are available on GitHub and PyPI, facilitating broader adoption of standard practices in micro network modeling.

**Keywords:** UTDF, SUMO, GMNS, Network Simulation, Signal Control

# Introduction

Existing work and shortages

UTDF

GMNS

Challenges in signal conversion

The current methodology and necessaries

Diagram

Description automatically generated

## 1.1 Subsection

Please notice that there is no indent in the first paragraph of a section or subsection.

New paragraphs are indented.

### 1.1.1 Three levels of headlines may be used

**Please only use the formatting styles predefined in this file**. Use the “Standard” formatting for the text. The paragraph format is justified. Please use hyphenation. **Please do not use footnotes.** Make sure to use the format “Paragraph after table” for the paragraph directly following the table.

**Table 1**. Centered table captions should be placed above the tables. Please use the word functionality “insert caption” and then format the caption with the “captions” format.

|  |  |
| --- | --- |
| **Hazard Class** | **A 1 to A III** |
| Flash point | < 21 °C / > 55 °C |
| Density at 15 °C | 720 kg/m3 to 860 kg/m3 |
| Kinematic Viscosity | 0,65 to 4,0 x 10–6 m2/s |

Tables and Figures should be centered. You can cross-reference your tables and figures in the text by using the respective functionality in Word (e.g. *Figure 1*). **Please do not merge cells in tables**. Please do not use in-line math in table captions. Make sure to use the format “Paragraph after table” for the paragraph directly following the table.

Bullet points may be used

….

1. Numbering may be used, too.
2. However, please do not use roman numbers.

Equations should be centred and set on a separate line. All equations need to be numbered 1-n. Please do not use 1a, 1b…

*x + y = z* (1)

where x is a variable.

In addition to the equations in the text, **Authors must provide in-line math and equations formatted in** [**KaTeX**](https://katex.org/docs/supported.html) **in a separate file.** In this file, equations must be numbered as in the Word document.



**Figure 1.** A centered figure caption is always placed below the illustration. Please use the word functionality “insert caption” and then and then format the caption with the “captions” format

In addition to the embedded figures in the Word file, **figures must be submitted as separate jpeg or png files.** If your figure consists of more than one panel, please merge all parts into one file. If your figure consists of more than one panel, please merge all parts into one file.

For citations of references, we prefer the use of square brackets and consecutive numbers, e.g. “as shown by Author et al. [2], [3, pp. 5–10]”, “as mentioned earlier [1], [3], [9]”. The first reference mentioned must be [1], the second reference mention [2] etc. **Please do not use reference linking in Word** (e.g. Word functionality “Insert Citation”, Citavi, or Mendeley) **as this causes problem during the XML conversion**. **However, we recommend adding references through the Zotero Word plugin**. If available, **DOIs must be provided.** The bibliography below provides the basic formats as a reference list with entries for journal articles [1], book chapter [2], as well as a URL [3]. For further guidance please refer to <https://ieeeauthorcenter.ieee.org/wp-content/uploads/IEEE-Reference-Guide.pdf>. Please do not forget that data (and other material underpinning the findings) need to be *cited* in the text and the respective reference must be included in the manuscript’s reference list. Please refer to the [data citation principles of FORCE11](https://www.force11.org/datacitationprinciples) or the [FORCE11 software citation principles](https://peerj.com/articles/cs-86/), respectively.

# 2. Source Code and Datasets

2.1. Source Code and Installation

The mgwr source code (the examples in this paper were composed using mgwr version 2.0.1) is organized as a module of the Python spatial analysis library (PySAL) (https://pysal.org) and is therefore available from a repository on the PySAL project GitHub page (https://github.com/pysal/mgwr). Each PySAL module is complete with ‘docstrings’ (i.e., input and output documentation) for all available functions and code examples (i.e., Jupyter notebooks) that make it simple to replicate and extend the examples to new applications. In addition, ‘unit tests’ are provided that allow the source code to be continuously integrated while being developed. This ensures that new features and dependency updates do not unknowingly break existing features.

Currently, mgwr has four dependencies: numpy, scipy, libpysal, and spglm. The first two dependencies, numpy and scipy, are elementary within the Python scientific computing ecosystem and provide core data structures and data manipulation functions. The third dependency, libpysal, is central to PySAL and provides a repository of example datasets. Since libpysal is dependent upon pandas, then pandas is an indirect dependency for mgwr and is often useful for reading and managing data tables. The final dependency, spglm, provides a light-weight generalized linear model framework for calibrating each of the local parameter estimates within (M)GWR via iteratively weighted least squares. The most recent stable version of mgwr, along with these direct and indirect dependencies, may be installed from the Python packaging index (PyPI) using the pip package manager:

pip install mgwr

To obtain in-development features, it is also possible to install mgwr directly from the source code:

pip install https://github.com/pysal/mgwr/archive/master.zip

Additional packages, namely matplotlib and geopandas, are used for presenting results from empirical demonstrations and can also be obtained via pip; however, they are not required for the core mgwr functions. Once all the necessary packages are installed, they can be imported for use in the following examples as such:

>>> import numpy as np

>>> import pandas as pd

>>> import libpysal as ps

>>> from mgwr.gwr import GWR, MGWR

>>> from mgwr.sel\_bw import Sel\_BW

>>> from mgwr.utils import compare\_surfaces, truncate\_colormap

>>> import geopandas as gp

>>> import matplotlib.pyplot as plt

>>> import matplotlib as mpl

2.2. Datasets

Two datasets are utilized throughout this paper to illustrate various (M)GWR functionality. First, is the well-known Georgia dataset that is described in [2] (2002) as well as subsequent publications [7,11]. The second is a sample of Airbnb rental data from the Prenzlauer Berg neighborhood of Berlin from InsideAirbnb, which provides a more recent example with a relatively larger sample size.

2.2.1. Georgia Dataset

The Georgia dataset consists of 159 counties in the state of Georgia (Figure 1), and records socio-demographic characteristics from the 1990 US census. The county locations are abstracted as centroids so that inter-county distances can be computed within the (M)GWR routine, though it is convenient to visualize the model output using the county polygons, since they are the scale at which the observations are aggregated. A small subset of the available variables are selected here for an example modeling educational attainment. The covariates are described in Table 1. Python code for loading and visualizing the Georgia dataset is as follows:

Figure 1. The 159 counties within the state of Georgia. Note: basemap and scalebar added using additional code.

Table 1. Georgia dataset.

#Load Georgia dataset and generate plot of Georgia counties (Figure 1)

>>> georgia = gp.read\_file(ps.examples.get\_path(‘G\_utm.shp’))

>>> fig, ax = plt.subplots(figsize = (10, 10))

>>> georgia.plot(ax=ax, \*\*{‘edgecolor’: ‘black’, ‘facecolor’: ‘white’})

>>> georgia.centroid.plot(ax = ax, c = ‘black’)

>>> plt.savefig(‘georgia\_shp’)

>>> plt.show()

2.2.2. Berlin Airbnb Dataset

The Berlin dataset consists of 2203 observations that are geolocated instances of Airbnb rental properties (Figure 2) and their associated characteristics from 2017 in the Prenzlauer Berg neighborhood. Prenzlauer Berg is a gentrifying neighborhood known for its arts scene, shopping, and nightlife, and is therefore a popular tourist destination. A small subset of variables were selected for a rental price modeling example, which are described in Table 2. Note that the logarithm of rental price is used here to correct the skewness of the variable. Since the data are not aggregated, the analysis and visualization of the results are carried out at the point-level.

Figure 2. 2203 rental properties in the Prenzlauer Berg neighborhood of Berlin. Note: basemap and scalebar added using additional code.

Table 2. Berlin dataset.

#Load Berlin dataset and generate plot of properties (Figure 2)

>>> prenz = gp.read\_file(ps.examples.get\_path(‘prenzlauer.zip’))

>>> prenz\_bound = gp.read\_file(ps.examples.get\_path(‘prenz\_bound.zip’))

>>> fig, ax = plt.subplots(figsize = (10, 10))

>>> prenz\_bound.plot(ax = ax, \*\*{‘edgecolor’: ‘black’, ‘facecolor’: ‘white’})

>>> prenz.plot(ax = ax, markersize = 10, \*\*{‘edgecolor’: ‘black’,

‘facecolor’: ‘black’})

>>> plt.savefig(‘prenz’)

>>> plt.show()

# 3. Methodology

# 4. Case Study

# 5. Conclusion and Future Work

Utilize the advantages of GMNS standards, the existing package include visualization

# Data availability statement

The SUMO and UTDF data utilized in this study are publicly accessible via GitHub ([Link](https://github.com/xyluo25/utdf2gmns)).

# Underlying and related material

GitHub repository: <https://github.com/xyluo25/utdf2gmns>

PyPI: <https://pypi.org/project/utdf2gmns/>

DOI: <https://doi.org/10.5281/zenodo.13951369>

# Author contributions

The authors confirm their contributions to the paper as follows: Xiangyong Luo developed the original code and drafted the initial manuscript. Yiran Zhang assisted in debugging the signal code and contributed the original code to GitHub. Xuesong Simon Zhou supported the early development of the package and provided additional revisions and modifications. Ross Wang assisted with writing and revising the paper. Guanhao Xu offered signal processing suggestions and contributed to the paper’s revisions and code validation. Wan Li assisted with writing the paper. Jeff Ban provided valuable resources for signal conversion and publicly contributed signal code on GitHub. All authors reviewed the results and approved the final version of the manuscript.

# Competing interests

The authors declare that they have no competing interests.

# Funding

The research is partially supported by the National Science Foundation (NSF) grant -------. Any opinions, findings, conclusions, recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of NSF.

# Acknowledgement

The research is partially supported by the National Science Foundation (NSF) grant XXXXXX. Any opinions, findings, conclusions, recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of NSF.

# References

1. J. K. Author, “Name of paper,” Abbrev. Title of Journal, vol.x, no.x, pp. xxx–xxx, Abbrev. month, year, doi: https.doi.org/……
2. J. K. Author, “Title of chapter in the book,” in Title of Their Published Book, xth ed. City of Publisher, (only U.S. State), Country: Abbrev. of Publisher, year, ch. x, sec. x, pp. xxx–xxx.
3. First Name Initial(s) Last Name. “Page Title.” Website Title. Web Address (date accessed)

“Traffic Lights - SUMO Documentation.” Accessed: Jan. 23, 2022. [Online]. Available: <https://sumo.dlr.de/docs/Simulation/Traffic_Lights.html>

“SUMO Road Networks - SUMO Documentation.” Accessed: Jan. 23, 2022. [Online]. Available: <https://sumo.dlr.de/docs/Networks/SUMO_Road_Networks.html>

“Vissim - SUMO Documentation.” Accessed: Jul. 27, 2021. [Online]. Available: <https://sumo.dlr.de/docs/Networks/Import/Vissim.html>

J. Ban, O. Angah, Y. Zhang, Q. Guo, Connected Cities for Smart Mobility toward Acces-sible and Resilient Transportation Center (C2SMART), and University of Washington, “A Multiscale Simulation Platform for Connected and Automated Transportation Systems,” Dec. 2022. Accessed: Feb. 01, 2024. [Online]. Available: <https://rosap.ntl.bts.gov/view/dot/67308>

N. N. Nor Azlan and M. Md Rohani, “Overview Of Application Of Traffic Simulation Mod-el,” MATEC Web of Conferences, vol. 150, p. 03006, 2018, doi: 10.1051/matecconf/201815003006.

[2] P. A. Lopez et al., “Microscopic Traffic Simulation using SUMO,” in 2018 21st Interna-tional Conference on Intelligent Transportation Systems (ITSC), Nov. 2018, pp. 2575–2582. doi: 10.1109/ITSC.2018.8569938.

[3] X. Zhou and J. Taylor, “DTALite: A queue-based mesoscopic traffic simulator for fast model evaluation and calibration,” Cogent Engineering, vol. 1, no. 1, p. 961345, Dec. 2014, doi: 10.1080/23311916.2014.961345.

[4] “PTV Visum.” Accessed: Apr. 24, 2024. [Online]. Available: https://www.ptvgroup.com/en-us/products/ptv-visum

[5] “PTV Vissim.” Accessed: Apr. 24, 2019. [Online]. Available: http://vision-traffic.ptvgroup.com/en-us/products/ptv-vissim/

“netedit - SUMO Documentation.” Accessed: Feb. 14, 2024. [Online]. Available: <https://sumo.dlr.de/docs/Netedit/index.html>

Alvarez Lopez, Pablo and Behrisch, Michael and Bieker-Walz, Laura and Erdmann, Jakob and Flötteröd, Yun-Pang and Hilbrich, Robert and Lücken, Leonhard and Rummel, Johannes and Wagner, Peter and Wießner, Evamarie (2018) *Microscopic Traffic Simulation using SUMO.* In: 2019 IEEE Intelligent Transportation Systems Conference (ITSC), pp. 2575-2582. IEEE. The 21st IEEE International Conference on Intelligent Transportation Systems, 2018-11-04 - 2018-11-07, Maui, USA. doi: [10.1109/ITSC.2018.8569938](https://doi.org/10.1109/ITSC.2018.8569938).