The Journal of Open Source Software

utdf2gmns: A Python Package for Mobility Simulation from UTDF to SUMO

Xiangyong Luo [1], **Yiran Zhang** [2], **and Xuesong (Simon) Zhou** [3]

1 Oak Ridge National Laboratory, United States **2** University of Washington, Seattle, WA, United States **3** Arizona State University, Tempe, AZ, United States

DOI: 10.21105/joss.10619

Software

- Review
- Repository
- Archive

Editor: Fangzhou Xie

Reviewers:

- @anitagraser
- @georgiastuart

Submitted: 11 April 2025
Published: 26 November 2026

License
Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License (CC BY 4.0).

Summary

utdf2gmns implements an automated workflow for network coordination, traffic signal integration, and traffic flow conversion from Synchro, commercial traffic signal-timing software, to Simulation of Urban Mobility (SUMO). The process begins with a comparative analysis of network topologies, data representations, and signal timing schemas in both environments. Converting Synchro Universal Traffic Data Format (UTDF) data to a network ready for microsimulation poses several challenges, including accurate signal integration, spatial transformation, and preservation of turning flow fidelity. Signal conversion represents a primary bottleneck, as it demands precise mapping of phasing plans, timing parameters, and coordination strategies to ensure valid simulation results. Network conversion is further complicated by translating Synchro's relative coordinate system into georeferenced formats compatible with geographic information system (GIS) tools. Furthermore, accurate transformation of turning movement data is essential for realistic intersection modeling.

Existing methods address only isolated conversion tasks and lack a fully automated, scalable end-to-end workflow. To fill this gap, we introduce utdf2gmns, an open-source Python library that automates the transformation of Synchro UTDF files into SUMO networks compliant with the General Modeling Network Specification (GMNS). utdf2gmns provides automatic geocoding, integration with Sigma-X for signalized-intersection analysis and visualization, robust SUMO network generation, and extensibility to other microsimulation platforms. Future work will extend support for adaptive signal control, integrate real-time data inputs, and enhance interoperability with additional simulation frameworks to promote reproducibility and collaborative traffic-modeling research.

Statement of need

Traffic microsimulation is essential for evaluating and improving urban transportation systems by providing high-resolution analysis of flow, congestion, and infrastructure performance. Such simulations depend on precise modeling of signal control, network geometry, and turning movements. Although Synchro's Universal Traffic Data Format (UTDF) delivers comprehensive intersection data, converting UTDF into simulation-ready networks remains manual, labor-intensive, and error-prone, limiting seamless interoperability with microsimulation platforms.

Several critical challenges remain when converting Synchro UTDF data into microsimulation-compatible networks, such as those required by Simulation of Urban Mobility (SUMO) (Lopez et al., 2018). First, accurate signal conversion demands detailed extraction and mapping of phasing, timing, and coordination parameters into standardized control formats; errors here can substantially degrade simulation fidelity. Second, network conversion requires transforming Synchro's relative coordinate system into georeferenced longitude–latitude coordinates for seamless GIS integration, a labor-intensive and error-prone process that limits scalability. Third,

realistic intersection dynamics hinge on precise turning movement conversion, which typically involves extensive manual preprocessing; inaccuracies at this stage can propagate through the simulation, undermining the validity of subsequent analyses.

To address these gaps (Ban et al., 2022; Coogan & Thitsa, 2021; Singh et al., 2018; Udomsilp et al., 2017; Zhang et al., 2024), we present utdf2gmns, an open-source Python tool that automates the conversion of Synchro UTDF files into GMNS-compliant networks (Smith et al., 2020) and generates simulation-ready inputs for SUMO. By leveraging the GMNS, a robust framework for standardized network representation (Berg et al., 2022; Lu & Zhou, 2023; Luo, 2024; Luo et al., 2024), utdf2gmns enhances data consistency, reproducibility, and collaboration through four core capabilities: it automates geocoding of Synchro's relative coordinates into accurate longitude–latitude pairs; integrates with Sigma-X to extract and optimize key intersection metrics (phasing diagrams, turning volumes, movement capacities, volume-to-capacity ratios, and control delays); generates GMNS-compliant SUMO networks that fully preserve signal coordination, traffic flows, and turning movements; and provides a modular architecture for extension to additional microsimulation platforms, thereby promoting broader standardization and community-driven development.

The documentation is available at Read the Docs.

State of the field

Traffic microsimulation scenarios are commonly created from open GIS/map data, built directly in a simulator's native formats, or derived from proprietary planning and operations tools. GIS-based pipelines scale well for network geometry, but key operational inputs (turning movements, lane connectivity, detectors, and signal timing) are typically added manually. When a study starts from Synchro exports, UTDF provides many of these operational details, but they are not available in an open-source simulator-ready representation.

SUMO offers flexible signal control modeling and is widely used for research on controller representation and validation (Halbach & Erdmann, 2022; Schrader et al., 2022). These capabilities primarily address how to represent and execute control once a scenario is specified; they do not provide a standard, reusable pathway for translating Synchro's phasing, timing, and coordination plans from UTDF into consistent, reproducible simulation inputs.

To our knowledge, there is no commonly used research software package that converts Synchro UTDF directly into a complete, SUMO-ready simulation scenario (network, flows, and signal control). Instead, published and practical workflows are typically assembled from project-specific scripts and manual steps, and they often cover only subsets of the required information (e.g., geometry, turning counts, or a subset of timing). Extending general-purpose GIS/OpenStreetMap (OSM) import utilities is therefore insufficient: supporting UTDF requires a dedicated parser and a semantics-preserving mapping of signal control and demand, plus validation tooling to ensure fidelity across intersections and time plans.

`utdf2gmns` provides this missing conversion layer as an end-to-end workflow from Synchro UTDF to a standardized intermediate representation (GMNS) and then to simulation-ready SUMO artifacts. The software's contribution is a reproducible translation of UTDF operational semantics (signal structure, timing/coordination plans, and turning movement demand) into open outputs suitable for large-scale scenario generation and comparative experiments, with optional Sigma-X integration to support intersection-level diagnostics and auditing.

Software design

The `utdf2gmns` package is designed with a modular architecture to facilitate the conversion of Synchro UTDF data into GMNS-compliant networks and SUMO simulations. The core

component is the `UTDF2GMNS` class, which orchestrates the entire conversion process through a series of method calls. The software is organized into several key modules:

- **Main Module (`_utdf2gmns.py`)**: Contains the primary `UTDF2GMNS` class, which initializes with UTDF file paths and region names. It provides high-level methods such as `geocode_utdf_intersections()`, `utdf_to_gmns()`, and `utdf_to_sumo()` to perform the conversions.
- **Functional Libraries (`func_lib/`)**: Divided into submodules for specific functionalities:
  - `utdf/`: Handles reading and processing UTDF data, including geocoding intersections and converting lane data.
  - `gmns/`: Manages the conversion to GMNS format, including node and link generation, and integration with Sigma-X for signal intersection processing.
  - `sumo/`: Generates SUMO-compatible Extensible Markup Language (XML) files for nodes, edges, connections, and traffic flows, while also handling signal controls and U-turn removal.
  - `plot_net.py`: Provides visualization capabilities using libraries like Matplotlib and Kepler.gl.
- **Utility Libraries (`util_lib/`)**: Includes helper functions for tasks such as distance calculations, time conversions, and package settings.
- **Engine (`engine/`)**: Integrates Sigma-X for advanced signalized intersection analysis, enabling the extraction of phasing diagrams, turning volumes, and control delays.

The modules are grouped by conversion stage rather than by a strict separation of input/output (I/O) and computation. Some helpers in `func_lib/` both transform data and write files; this keeps the staged workflow accessible through the high-level class, but couples those helpers to file formats and makes isolated testing more difficult. The architecture therefore provides stage-level organization, rather than complete independence between processing and I/O.

Using GMNS as an intermediate representation exposes node, link, lane, and movement tables for inspection and reuse outside SUMO, at the cost of maintaining mappings between UTDF records, GMNS tables, and SUMO XML. SUMO-specific signal and connection handling is implemented in the SUMO modules; GMNS lane generation also reuses geometry helpers from those modules, so these stages are not fully decoupled. A caller-supplied reference intersection coordinate avoids reliance on address-geocoding services for reproducible examples; automatic address lookup remains available when a reference coordinate is unavailable. Sigma-X is optional because its Excel automation requires desktop Microsoft Excel on Windows or macOS, while the core conversion workflow can run without it. These boundaries let users inspect intermediate artifacts and run the core workflow without the optional visualization and spreadsheet dependencies.

Research impact statement

The `utdf2gmns` package has demonstrated impact in traffic microsimulation by facilitating the conversion of Synchro UTDF data into SUMO-compatible networks. A peer-reviewed Winter Simulation Conference paper presents `utdf2gmns` as an automated workflow for converting UTDF network representation, signalized intersections, and turning volumes into GMNS-compliant SUMO inputs, and reports case-study validation for complex urban corridors (Luo et al., 2025). The same workflow directly addresses the manual Synchro-to-SUMO preparation steps identified in earlier traffic simulation studies (Singh et al., 2018; Udomsilp et al., 2017; Zhang et al., 2024). The subsequent Sim2Signal preprint cites this UTDF-to-SUMO work when describing the construction of its Tempe and Bullhead City benchmark networks from real UTDF signal-plan data (Rafi et al., 2026), providing evidence of scholarly uptake in traffic signal control research.

Furthermore, the integration of `utdf2gmns` with Sigma-X supports visualization and auditing of signalized intersections, including phasing diagrams, turning volumes, volume-to-capacity ratios, and control delays (Luo et al., 2025). These capabilities provide practical tools for urban planners, traffic engineers, and researchers who need reproducible network preparation rather than project-specific scripts.

The Real-Twin project uses `utdf2gmns` for file conversion in its transportation simulation workflow (Wang et al., 2025). This application provides a concrete example of research use. The package also recorded more than 36,000 Python Package Index (PyPI) downloads as of June 16, 2026 (PePy.tech, 2026), reflecting distribution through the public package index.

The project combines contributions to its conception, implementation, and testing. Xuesong Zhou proposed the initial project concept and supported project funding, while Xiangyong Luo maintains the software implementation. In February 2025, contributor `Yiran6` submitted four commits through a public pull request that Luo merged.[1] These changes corrected signal-direction mapping, replaced SUMO traffic-light logic directly in the network XML, and added simulation configuration and route files for testing. This development record documents substantive participation by another developer in signal conversion and test preparation.

The repository provides contribution instructions and public issue forms for reporting problems, requesting support, and proposing improvements.

Hands-On tutorial

Run this example from the root of a checkout of the repository's `joss` branch after installing the package with `python -m pip install -e .`. The input file, `datasets/data_bullhead_seg4/UTDF.csv`, is tracked in the repository. The Python source distribution and wheel do not include this example dataset; use the repository checkout or download the CSV separately and adjust the input path. The supplied reference coordinate uses longitude for `x_coord` and latitude for `y_coord` and avoids an external address lookup. The default steps write GMNS comma-separated value (CSV) tables and signal data beside the input file in `datasets/data_bullhead_seg4/utdf_to_gmns/`; the SUMO, visualization, and Sigma-X steps are optional.

```python
import utdf2gmns as ug


if __name__ == "__main__":

    region_name = "Bullhead City, AZ"
    path_utdf = "datasets/data_bullhead_seg4/UTDF.csv"

    # Step 1: Initialize the UTDF2GMNS
    net = ug.UTDF2GMNS(utdf_filename=path_utdf, region_name=region_name)

    # Step 2: Geocode intersections from one known intersection coordinate
    single_coord = {
        "INTID": "39",
        "x_coord": -114.59807666698381,
        "y_coord": 35.02605198650903,
    }
    net.geocode_utdf_intersections(single_intersection_coord=single_coord)

    # Step 3: convert UTDF network to GMNS format (csv)
```

[1] Pull request #1: Fix Signal Conversion, merged February 1, 2025.

```
net.utdf_to_gmns(incl_utdf=True)

# Step 4 (optional): install SUMO and put netconvert on PATH first.
# net.utdf_to_sumo(sim_name="", remove_U_turn=True, sim_duration=7200)

# Step 5 (optional): visualize the network
# Install the base extra before enabling optional visualizations.
# net_map = ug.plot_net_mpl(net, save_fig=True, fig_name="Bullhead_City.png")
# net_map = ug.plot_net_keplergl(net,
#                                save_fig=True,
#                                fig_name="Bullhead_City.html")

# Step 6 (optional): Sigma-X requires desktop Excel on Windows/macOS
# and its optional Excel-automation dependency.
# net.utdf_to_gmns_signal_ints()
```

AI usage disclosure

GitHub Copilot assisted selected repository maintenance changes, including continuous-integration fixes. ChatGPT-5.2 assisted earlier manuscript polishing. OpenAI Codex assisted a subsequent evidence audit and revisions to the design discussion, tutorial, API documentation, and test workflows, as well as final editorial corrections and bibliography verification. The authors are responsible for reviewing the revised text, verifying the reported evidence, and approving the final submission.

Acknowledgements

We thank Prof. Milan Zlatkovic (University of Wyoming) for Sigma-X, which is used to visualize signalized intersections.

We thank Anita Graser and Georgia Stuart for independent testing and feedback during review for the Journal of Open Source Software (JOSS). Their issue reports prompted clearer geocoding dependency errors, complete default runtime dependencies, explicit platform requirements for Sigma-X, and a compatible Setuptools constraint for optional Kepler.gl visualization.[2]

This manuscript has been authored in part by UT-Battelle, LLC, under contract DE-AC05-00OR22725 with the U.S. Department of Energy (DOE). The publisher acknowledges the U.S. government license to provide public access under the DOE Public Access Plan.

Additional support was provided by the DOE, Office of Energy Efficiency and Renewable Energy (EERE), Vehicle Technologies Office. Real-Twin is supported by the DOE Vehicle Technologies Office, Energy Efficient Mobility Systems (EEMS) program under project EEMS114 (Wang et al., 2025). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the U.S. Department of Transportation (USDOT), DOE, and the U.S. Government assumes no liability for the contents or use thereof.

References

Ban, J., Angah, O., Zhang, Y., & Guo, Q. (2022). *A multiscale simulation platform for connected and automated transportation systems*. Connected Cities for Smart Mobility toward Accessible and Resilient Transportation Center (C2SMART). https://rosap.ntl.bts.gov/view/dot/67308

[2] Issue #15: geocoding, issue #17: Sigma-X platform requirements, and issue #18: Kepler.gl dependencies.

Berg, I., Smith, S., & Zhou, X. S. (2022). GMNS: A Specification for Sharing Routable Road Networks. *Transportation Research Board 101st Annual Meeting*. https://trid.trb.org/View/1909441

Coogan, S., & Thitsa, M. (2021). *Coordinated Anti-Congestion Control Algorithms for Diverging Diamond Interchanges* (FHWA-GA-21-1913). Georgia Department of Transportation, Office of Performance-Based Management & Research. https://rosap.ntl.bts.gov/view/dot/56488

Halbach, M., & Erdmann, J. (2022). High fidelity modelling of traffic light control with XML logic representation. *SUMO Conference Proceedings*, *3*, 45–68. https://doi.org/10.52825/scp.v3i.114

Lopez, P. A., Behrisch, M., Bieker-Walz, L., Erdmann, J., Flötteröd, Y.-P., Hilbrich, R., Lücken, L., Rummel, J., Wagner, P., & Wiessner, E. (2018). Microscopic Traffic Simulation using SUMO. *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, 2575–2582. https://doi.org/10.1109/ITSC.2018.8569938

Lu, J., & Zhou, X. S. (2023). Virtual track networks: A hierarchical modeling framework and open-source tools for simplified and efficient connected and automated mobility (CAM) system design based on general modeling network specification (GMNS). *Transportation Research Part C: Emerging Technologies*, *153*, 104223. https://doi.org/10.1016/j.trc.2023.104223

Luo, X. (2024). *Strategic Planning for Scalable Electric Vehicle Charging Networks: Integrating Operational Dynamics and Spatial-Temporal Analysis* [PhD thesis]. Arizona State University.

Luo, X., Kuby, M. J., Honma, Y., Kchaou-Boujelben, M., & Zhou, X. S. (2024). Innovation diffusion in EV charging location decisions: Integrating demand & supply through market dynamics. *Transportation Research Part C: Emerging Technologies*, *165*, 104733. https://doi.org/10.1016/j.trc.2024.104733

Luo, X., Zhang, Y., Xu, G., Li, W., Wang, R., & Zhou, X. S. (2025). Automating Traffic Microsimulation from SYNCHRO UTDF to SUMO. *2025 Winter Simulation Conference (WSC)*, 2320–2331. https://doi.org/10.1109/WSC68292.2025.11338949

PePy.tech. (2026). *utdf2gmns PyPI Download Statistics*. https://pepy.tech/projects/utdf2gmns

Rafi, F. A., Mukherjee, S., Dekate, L. L., Lavoe, J. Y., Yao, H., Mohanty, S., Da, L., Zhou, X., & Wei, H. (2026). *Sim2Signal: Sim-to-Real Benchmarks for Traffic Signal Control*. https://doi.org/10.48550/arXiv.2609.01676

Schrader, M., Wang, Q., & Bittle, J. (2022). Extension and Validation of NEMA-Style Dual-Ring Controller in SUMO. *SUMO Conference Proceedings*, *3*, 1–13. https://doi.org/10.52825/scp.v3i.115

Singh, S. K., Komolkiti, P., & Aswakul, C. (2018). Impact analysis of start-up lost time at major intersections on sathorn road using a synchro optimization and a microscopic SUMO traffic simulation. *IEEE Access*, *6*, 6327–6340. https://doi.org/10.1109/ACCESS.2017.2739240

Smith, S., Berg, I., & Yang, C. (2020). *General Modeling Network Specification: documentation, software, and data*. Dataset and software; Volpe National Transportation Systems Center. https://doi.org/10.21949/1518740

Udomsilp, K., Arayakarnkul, T., Watarakitpaisarn, S., Komolkiti, P., Rudjanakanoknad, J., & Aswakul, C. (2017). Traffic data analysis on sathorn road with synchro optimization and traffic simulation. *Engineering Journal*, *21*(6), 57–67. https://doi.org/10.4186/ej.2017.21.6.57

Wang, C. R., Xu, G., Saroj, A., Luo, X. R., Yuan, J., & Shao, Y. (2025). *Real-twin*. U.S. Department of Energy. https://doi.org/10.11578/dc.20250602.3

Zhang, Y., Fu, M., & Ban, X. (2024). Integration Traffic Signal Control From Synchro to SUMO. *SUMO Conference Proceedings*, *5*, 147–162. https://doi.org/10.52825/scp.v5i.1112