

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

Xiangyong Luo¹, Yiran Zhang², and Xuesong (Simon) Zhou³

DOI: [DOI unavailable](#)

Software

- [Review ↗](#)
- [Repository ↗](#)
- [Archive ↗](#)

Editor: [Pending Editor ↗](#)

Reviewers:

- [@Pending Reviewers](#)

Submitted: N/A

Published: N/A

License

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

Summary

UTDF2GMNS¹ implements an automated workflow for network coordination, traffic signal integration, and traffic flow conversion from Synchro to SUMO. The process begins with a comparative analysis of network topologies, data representations, and signal timing schemas in both environments. Converting Synchro 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 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 GMNS-compliant SUMO networks. `utdf2gmns` provides automatic geocoding, Sigma-X engine integration for intersection analysis, 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.

¹This manuscript has been authored in part by UT-Battelle, LLC, under contract DE-AC05-00OR22725 with the US Department of Energy (DOE). The publisher acknowledges the US government license to provide public access under the [DOE Public Access Plan](#)

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 et al., 2021; Singh et al., 2017; Udomsilp et al., 2017; Zhang et al., 2024), we present `utdf2gmns` ([Luo and Zhou 2022](#)), 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 the Sigma-X engine ([Milan 2022](#)) 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.

For detailed documentation, please refer to [Official Documentation](#)

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/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 the Sigma-X engine for signal intersection processing.
 - `sumo/`: Generates SUMO-compatible 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 the Sigma-X engine for advanced signalized intersection analysis, enabling the extraction of phasing diagrams, turning volumes, and control delays.

This modular design ensures extensibility, allowing users to customize or extend functionalities for other microsimulation platforms. The package leverages external libraries like Pandas for data manipulation and pyufunc for utility functions, promoting code reusability and maintainability.

Research impact statement

The `utdf2gmns` package has demonstrated significant impact in the field of traffic microsimulation by facilitating the conversion of Synchro UTDF data into SUMO-compatible networks. Evidence of publications utilizing `utdf2gmns` includes studies that benchmark its performance against traditional methods, showcasing improved accuracy and efficiency in traffic flow simulations. Notably, comparative analyses have highlighted the software's ability to automate complex conversion processes, reducing the time and effort required for network preparation.

Furthermore, the integration of `utdf2gmns` with the Sigma-X engine has enabled researchers to visualize signalized intersections effectively, providing a valuable tool for urban planners and traffic engineers. The software's open-source nature encourages external adoption and contributions, fostering a collaborative environment for continuous improvement and innovation in traffic simulation methodologies.

`utdf2gmns` has been downloaded more than 30k times from PyPI since its release on July 25, 2023, indicating strong demand and sustained use within the transportation modeling and microsimulation community. Its users span career stages from graduate students to faculty and established researchers, and come from institutions worldwide. This level of adoption and engagement highlights `utdf2gmns` as a widely trusted, community-facing tool that lowers barriers to data preparation and reproducible simulation workflows, and increasingly serves as core infrastructure supporting transportation simulation research.

Hands-On tutorial

```
import utdf2gmns as ug
```

```
if __name__ == "__main__":  
  
    region_name = "Region-name" # e.g. "Tempe, AZ"  
    path_utdf = "Path-to-UTDF.csv" # e.g. "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 intersection  
    net.geocode_utdf_intersections()  
  
    # Step 3: convert UTDF network to GMNS format (csv)  
    net.utdf_to_gmns(incl_utdf=True)  
  
    # Step 4: convert UTDF network to SUMO  
    net.utdf_to_sumo(sim_name="", disable_U_turn=True, sim_duration=7200)  
  
    # Step 5 (optional): visualize the network  
    net_map = ug.plot_net_mpl(net, save_fig=True, fig_name=f"{region_name}.png")  
    net_map = ug.plot_net_keplergl(net,  
                                    save_fig=True,  
                                    fig_name=f"{region_name}.html")  
  
    # Step 6: Sigma-X visualize signalized intersection  
    # net.utdf_to_gmns_signal_ints()
```

AI usage disclosure

No generative AI tools were used in the development of this software, ChatGPT-5.2 was used to improve the clarity and readability of the manuscript.

Acknowledgements

Prof. Milan Zlatkovic from University of Wyoming, for the power of sigma-x engine to visualize signalized intersections.

Additional support was provided by the U.S. Department of Energy (DOE), Office of Energy Efficiency and Renewable Energy (EERE), Vehicle Technologies Office. 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 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., & others. (2022). A multiscale simulation platform for connected and automated transportation systems. *Connected Cities for Smart Mobility Toward Accessible and Resilient ...*
- Berg, I., Smith, S., & Zhou, X. S. (2022). GMNS: A Specification for Sharing Routable Road Networks.
- Coogan, S., Thitsa, M., & others. (2021). Coordinated Anti-Congestion Control Algorithms

for Diverging Diamond Interchanges. Georgia. Department of Transportation. Office of Performance-Based

- 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, & others. (2018). Microscopic traffic simulation using sumo. *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, 2575–2582.
- 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.
- Luo, X. (2024). Strategic Planning for Scalable Electric Vehicle Charging Networks: Integrating Operational Dynamics and Spatial-Temporal Analysis [PhD thesis]. In *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.
- 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. (2017). 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.
- Smith, S., Berg, I., Yang, C., John, A., & others. (2020). *General Modeling Network Specification: documentation, software, and data*.
- 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.
- Zhang, Y., Fu, M., & Ban, X. (2024). Integration Traffic Signal Control From Synchro to SUMO. *SUMO Conference Proceedings*, 5, 147–162.