SUMO User Conference

TIBconferencesession

TIB-OP will set DOI with \TIBdoi

Submitted: 1970-01-01

hybridPY

The Simulation Suite for Mesoscopic and Microscopic Traffic Simulations

Jörg Schweizer[1] * , Fabian Schuhmann[2*]
, and Cristian Poliziani[3]

[1]University of Bologna, Italy

[2]Technische Universität München, Germany

[3]Lawrence Berkeley National Lab, USA

Correspondence: Joerg Schweizer, joerg.schweizer@unibo.it

Abstract: Mesoscopic, agent-based simulations efficiently model and assess entire regions' daily activities and travel patterns, exemplified by smaller countries like Switzerland. The queue-based simulation represents a compromise between computational speed on the one hand and the necessity of detailed modeling infrastructure on the other hand. Thus, mesoscopic simulations enable an efficient and reasonably detailed analysis of the complex interplay between supply and demand in mobility research. Conversely, microsimulations excel at reproducing individual speed profiles and behavior by modeling the interactions between traffic participants, including pedestrians, bicycles, and scooters. Although allowing for more detailed system analysis, the downside is the high computational burden, which often prevents large-scale microscopic simulations from running in optimization or calibration loops. hybridPY, an extension of SUMOPy, aims to close the gap and benefit from both environments. The simulation suite allows the running of mesoscopic as well as microscopic traffic simulations based on the core idea: running a microscopic simulation in a smaller dedicated area, using the routes or mobility plans generated from a larger mesoscopic model. The main features of this software are: (i) import, editing and visualization of MATSim and BEAM CORE networks; (ii) conversion of MATSim plans to SUMO routes or plans within the SUMO area; (iii) configuring and running of MATSim simulations. The capability of hybridPY is demonstrated by two applications: the simulation of Schwabing, Germany, based on the MITO MATSim model, and the San Francisco municipality, USA, based on the mesoscopic BEAM CORE model of the entire San Francisco Bay area. Both scenarios demonstrate that the hybrid approach results in significant computational gains with respect to a pure microscopic approach.

Keywords: MATSim, hybrid traffic simualtion

*These authors contributed equally to this work.

1 Introduction

1.1 Motivation and scope

The population's daily activities and trips of entire regions or even small countries like Switzerland [1] can be modeled, simulated and evaluated with multi-modal, mesoscopic simulations [2]. This queue-based simulation approach appears to represent a virtuous compromise between computational speed on the one hand and the necessity to model infrastructure and travel times to a sufficient detail on the other hand [3]. Moreover, transport services such as ride pooling or car sharing can be simulated [4] as the system keeps track of individual persons and vehicles.

Microsimulations are capable of reproducing speed profiles by modeling the interaction between different types of vehicles (including bikes and scooters) as well as between vehicles and persons [5]. In addition, the microsimulator SUMO features a sub-lane model, where vehicle movements within the lane are simulated, for instance cars can overtake bikes within the same lane. The detailed modelling of person- and bike movements, and their interaction with cars is of significant importance as walking and cycling are exactly the modes that are considered "sustainable" and poised to play an increasing role in transport system analysis. Emerging vehicle technologies, such as autonomous vehicles with dedicated car following behavior can also be modelled with microsimulations [6], even at device-level[7], if necessary. This makes microsimulations the natural choice for transport system developers. However, it would be desirable to integrate and evaluate new technologies or services into the the city's overall mobility system.

The downside of microscopic models is clearly the high computational burden – a fact that often prevents large-scale microscopic simulations from running in optimization or calibration loops as microsimulations of larger areas typically execute much slower than real time [8], [9]. This raises the question whether there are ways to combine mesoscopic and microscopic simulation approaches to take the benefit from both?

An important aspect to consider is the way flows and carrying capacities are handled in mesoscopic and microscopic models. With the used mesoscopic models (e.g. MATSim), the link-capacity is pre-calculated based on link attributes, such as length, allowed speed and number of lanes. During the simulation, the algorithm allows link flows to increase up to this capacity limit. After hitting the limit, the link-flow saturates and the link flow remains at this carrying capacity, independent from the pressure of vehicles from upstream links. This fact allows to simulate a scenario with a down-scaled demand: For instance a scenario can be simulated with only 10% of the vehicles which result exactly in 10% of the real link flows. But this is only possible if the pre-calculated capacity limits are also scaled down by a factor of 10. This is of course a gain of an order of magnitude in terms of computational speed.

By contrast, microscopic model shows a completely different behavior: the capacity is not explicitly defined, but is an implicit function of the many attributes of links, lanes and vehicles. In addition, the particular topology such as merges and intersections as well as traffic light programs limit vehicle flows. When vehicle flows hit this implicit capacity limit, then often a complete traffic breakdown is the consequence. In strong contrast with the mesoscopic simulation where the flow remains at capacity limit, the microscopic simulation shows a behavior where the link flow drops to near zero after hitting the capacity limit. While this may be a desired effect, as it resembles reality, the modelling and calibration of all the attributes and parameters is generally difficult, especially with large networks. In essence, the mesoscopic simulation will not experience traffic-flow breakdowns when flows hit capacity limits (but upstream spill-backs may occur). However, the set capacity limits are likely to differ from those in reality. With the microsimulation, capacity limits are potentially closer to reality, but if attributes and parameters are not modelled correctly, traffic breakdown will occur at unrealistic times and places. Moreover, with a micro-simulation it is generally not possible to scale down demand to reduce computational burden, as it is the case with mesoscopic simulations. While there is no definitive solution to this problem, it is clear that mesoscopic and microscopic models have complementary properties which may be explored in a combined approach.

The scope of this article is to demonstrate the feasibility of the ”hybrid” (mesoscopic - microscopic) approach: the software framework called ”hybridPY” that enables the microsimulation of a smaller area within a larger mesoscopic demand model. The feasibility is demonstrated in two ways: 1.) Two use-cases are presented where the hybrid approach has been successfully applied; 2.) from these use-cases it is shown that it is impractical (due to the computational burden) to simulate the entire network with a microsimulation-only approach, hence the hybrid approach is needed in these cases.

Note that the presented hybridPY framework is unique as it allows to run mesoscopic and microscopic of the same scenario on from the same platform and exchange results between the two models.

The paper is structured as follows: In the remaining section 1.2 a brief overview of mesoscopic traffic simulation packages, namely MATSim and BEAM CORE, are introduced. In addition, existing simulation suits are discussed and the hybrid simulation is presented and motivated. Section 2 introduces the hybridPY framework and its core elements. Afterwards, Sec. 3 demonstrates the capabilities of hybridPY within the use-cases Sec. 4 draws preliminary conclusions of the current hybridPY development and outlines future developments.

1.2 State of the art

The following section gives a brief overview of the Multi-Agent Transport Simulation MATSim, the BEAM CORE Framework, existing simulation suits as well as hybrid traffic simulation approaches.

1.2.1 The Multi-Agent Transport Simulation MATSim

MATSim is an open-source framework designed for large-scale, agent-based traffic simulations. The framework is based on a co-evolutionary algorithm where agents egoistically optimize their daily activity schedules by adjusting e.g. the departure time, the used mode of transport, or the route. The optimization is run within a loop until an equilibrium state is reached. The microscopic traffic model of MATSim simplifies the traffic dynamics using a queue-based model. The framework allows modeling different modes, like a car, walking, bike, and public transport, individually but without interaction. Despite its limitations coming along with the simplified dynamics, the framework efficiently simulates large scenarios, prioritizing speed over precise traffic and driving dynamics.

1.2.2 BEAM CORE

BEAM CORE [10] is a comprehensive modeling framework that simulates the dynamics of passengers and freights in urban areas with BEAM CORE [11], using the PILATES models pipeline. The framework consists of various interconnected models that create a synthetic population, vehicle ownership, daily activities, mode choices, and routes, see Fig. 1. SynthPop, DEMOS, and Urbansim generate a synthetic population and its evolution over the years, accounting for demographics, land use, and household characteristics. ADOPT, FASTSim, and ATLAS predict vehicle ownership and characteristics over the years. ActivitySim models daily activities, locations, and transportation modes, and provides plan-information to BEAM CORE. This latter then simulates the trips using an event-based mesoscopic approach for all transport modes, adaptable to large-scale studies. The transit service on BEAM CORE is reproduced through GTFS files, while on-demand services are modeled by an internal vehicle-request manager. Finally, RouteE calculates the energy consumption of vehicles. Each BEAM CORE submodel can be calibrated independently, allowing for scenario-specific adjustments without the need of rebuilding the entire model. BEAM CORE iterates between simulating travel plans with BEAM CORE and adjusting mode choices on ActivitySim based on transport skims, aiming to achieve equilibrium in travel patterns. After a first case-study calibration, successive simulations can start from already calibrated skims, drastically reducing the iterations needed to converge the results again. As anticipated, the BEAM CORE model can be enabled to reproduce yearly iterations, to analyze long-term impacts of specific scenarios. However, BEAM CORE is computationally intensive and typically requires more computational power than a standard laptop, especially for large-scale simulations. In fact, it can run smaller-scale studies by focusing on a subset of the population and scaling the results in postprocessing. This scaling also applies to road networks, ridehail fleets, and public transit capacities to accurately simulate congestion effects. The BEAM CORE model already has some built-in post processing, and one in particular aggregates the event-based results as door-to-door trips to evaluate an accessibility measure called INEXUS [12]. This database is particularly important for this study since it's compatible to other simulation platforms like SUMO.

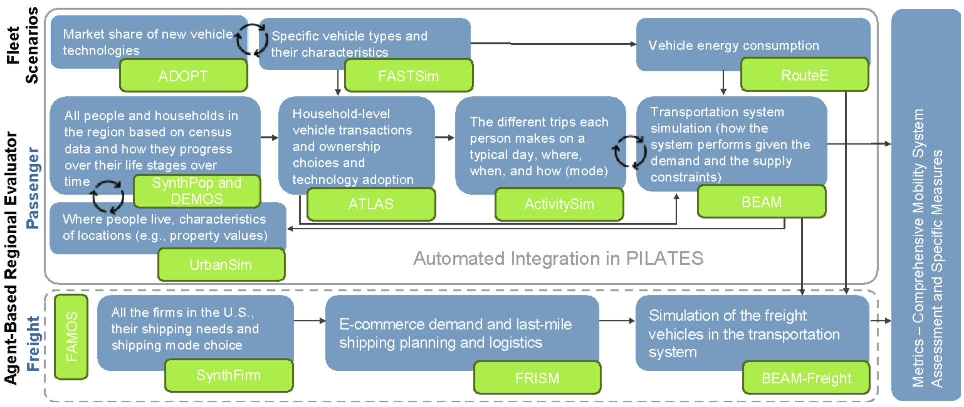


Figure 1. *BEAM CORE structure*

1.2.3 Simulation Suits for Open Source Traffic Simulations

The initial hurdle for the use of traffic simulations is the creation of the scenario. Multiple simulation suites could be identified in the field of open source traffic simulations, namely MATSim and SUMO.

*www.replan.city** is a platform designed to assist urban planners with agent-based modeling for addressing urban mobility challenges. The platform is based on the open-source simulation MATSim and offers tools for scenario generation, automation of model generation, simulation of various transportation modes, and the use of web and cloud computing for calculations and visualizations.

A comparable platform is *Tramola*. Tramola is designed to simulate and analyze mobility systems in urban environments. Similar to replan.City, Tramola is also built on MATSim. Despite their user-friendliness and simplicity of operation, both platforms are commercial and not open-source. Additionally, to the best of our knowledge, both platforms do not support other simulation cores than MATSim.
The MATSim ecosystem provides open-source tools as well. The following can be listed as examples: [13], [14]. Nevertheless, those tools are specialized only on parts of the modelling process, e.g. on network editing and visualization.
For microscopic traffic simulation SUMO, SUMOPy is a user-friendly, graphical tool to create traffic simulation scenarios easily and efficiently. **SUMOPy** is open-source †. Apart from SUMOPy, several other graphical user interfaces exist for SUMO. Cloud-based, commercial solutions ‡, specialized tools for creating simulation scenarios simulating vehicular networks [15] [16], as well as demand management tools, comparable to SUMOPy,[17] [18]. Nevertehless, the latter tools do not offer the same flexibility as SUMOPy, they are specialized for particular use cases or are no longer maintained.

1.2.4 Hybrid Traffic Simulation Approaches

Existing approaches for coupling the mesoscopic simulation MATSim and the microscopic simulation SUMO can be divided into four different approaches. These are depicted in figure 2.

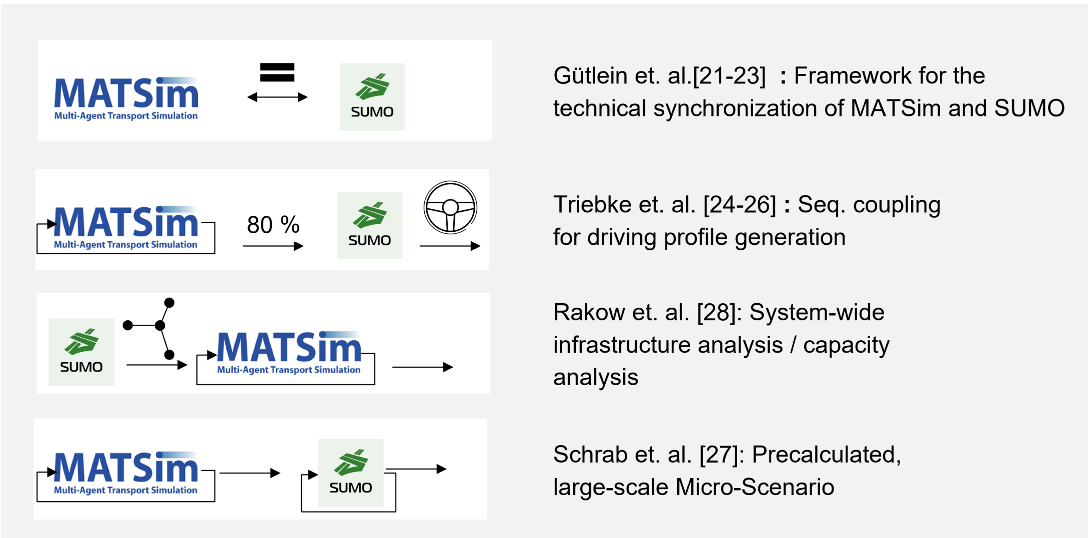


Figure 2. *Hybrid concepts existing in literature.*

M. Gütlein [19] propose a co-simulation framework running MATSim and SUMO in parallel. Nevertheless, their focus is on operational efficiency and technichal synchronization instead of scenario consistency of the simulation environments [19] [20] [21].

*www.replan.city
†https://github.com/schwoz/SUMOPy
‡https://sesam.co4e.com/

The motivation of Triebke [22] for coupling MATSim and SUMO is to efficiently deduce drive cycles for shared, autonomous vehicles. Triebke [22] suggest simulating the fleet strategy inside MATSim and afterward enriching the drive cycles by microsimulating the resulting strategy. In [23], [22], an automated approach for a tool-coupling and calibration is presented. Their approach relies on the imperfect, uncleaned microscopic networks imported from OSM. Their results show differences regarding traffic dynamics on a link-level and junction capacities from the different network representations. The authors identify the microscopic simulation and its creation as the hybrid traffic simulation bottleneck in terms of efficient scenario creation [24].

Schrab has used a calibrated demand definition from a MATSim Scenario and has projected the demand in a scaled and randomized manner to a manually cleaned SUMO network, see [25]. He has demonstrated the feasibility of this method by comparing the results to measured traffic counts. Thus, he proved that this approach is usable for analyzing new mobility solutions. Nevertheless, his method relies on re-estimating the user equilibrium by running the SUMO simulation and route choice iteratively, which is expected to show long simulation times, as shown below.
In order to tackle the overestimation of junction capacities by mesoscopic models, Rakow [26] use a hybrid modeling approach combining SUMO and MATSim. The framework integrates the microscopic simulation SUMO for assessing traffic-flow capacities at signalized intersections in order to run downstream MATSim simulations for exploring system-wide and long-term impacts.
The state-of-the-art shows that hybrid traffic scenarios have the potential to show good results; nevertheless, the above mentioned approaches do not provide a process or a framework to run hybrid traffic simulations. Nevertheless, the current state of the art acknowledges that the accurate modeling of the microscopic infrastructure is crucial for a successful simulation.

The present work uses also MATSim/BEAM CORE as mesoscopic and SUMO as microscopic traffic models. The main resaon for this choice is that both simulators are well established in their respective domains. Regarding MATSim, this simulator has been already integrated with different demand generation models, and many large scale urban scenarios have been generated and calibrated. The same applies for BEAM CORE.

In theory, it is possible to use the SUMO-internal mesoscopic simulator, which would avoid the merge of two different network types (e.g the mesoscopic and microscopic network). However, in this case, one would need to (i) redo the large scale networks already available under MATSim or BEAM CORE, (ii) adapt the demand-output of the various activity-generators to SUMO format, (iii) rerun the entire generation, calibration and validation processes. In other words, one would need to rebuild entirely new scenarios. Regarding the SUMO mesoscopic simulator, there is also a technical issue: as mentioned above, with MATSim (and BEAM CORE), the link-capacity can be set as a link-attribute, which is not the case with the Mesoscopic SUMO simulator, to the knowledge of the authors. But this feature is essential to be able to scale the demand, as explained in the introduction Sec.1.1. However, this issue is not a severe technical problem and may be resolved in future SUMO versions.

2 The hybridPY framework

2.1 Overview

hybridPY is an open-source simulation suite for running microscopic, mesoscopic, and hybrid simulation scenarios. The framework is based on **SUMOPy** and written in Python 3.9.
It relies on a containerized architecture for running the traffic simulations easily and operating the system independently. The strength of hybridPY is that it allows the run of SUMO and MATSim Scenarios independently and can transform the data between the two simulation models.
Therefore, the core features of the simulation framework are:

- Running and analyzing microscopic simulation scenarios
- Running and analyzing mesoscopic simulation scenarios
- Hybrid-simulations scenarios, wherein MATSim Scenarios and SUMO Scenarios are executed sequentially

A structural overview of the simulation framework is given in figure 3. The framework is capable of reading, writing and also creating configuration files, population files, and network files for both the mesoscopic as well as the microscopic traffic simulation. An example of the resulting graphical user interface is given in figure 4. The data

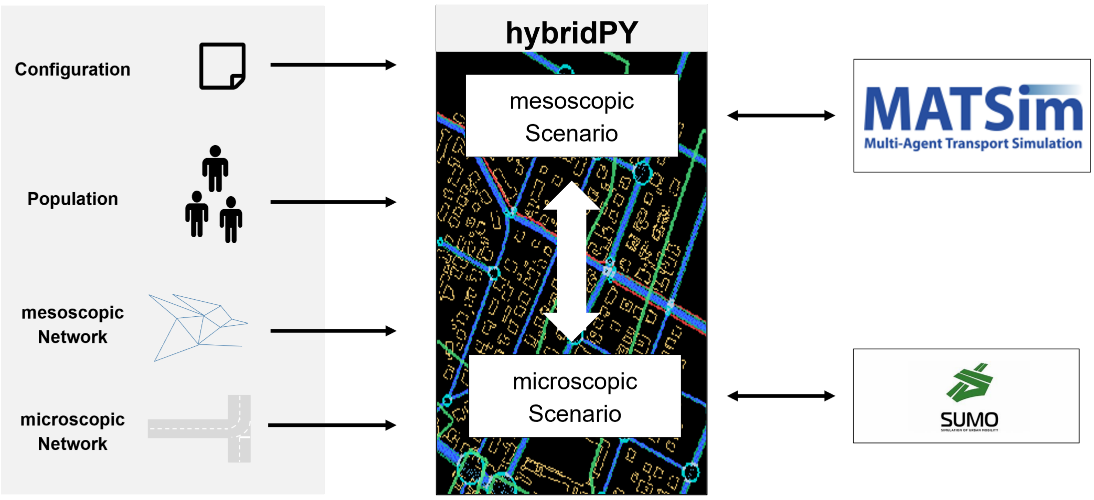


Figure 3. *Concept of hybridPY*

browser is displayed on the left side of the application. All the processes and functions can be launched from this GUI, or alternatively via scripting. The data browser gives the user a structured view of the required simulation data. Due to the object-oriented implementation of the underlying data architecture, it is possible to extending it and adapting it to new use cases. The network editor is shown on the right-hand side, which enables microscopic and mesoscopic networks to be displayed at different levels. Thus, the network editor permits a detailed and targeted examination of the the hybrid network model.

2.2 The hybrid modeling workflow

This pipeline integrates MATSim and SUMO to leverage their strengths in simulating agent-based travel behavior, enabling a more comprehensive analysis of transportation systems and urban mobility. In this section, the eighth step of the modeling framework

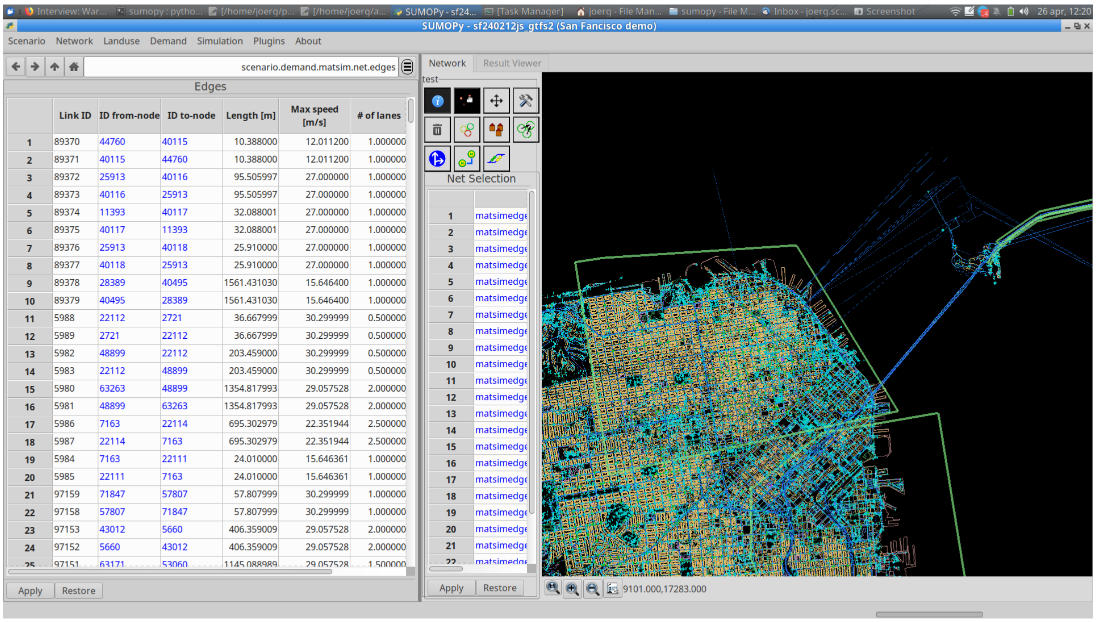


Figure 4. *Graphical user interface to interact with the MATSim and SUMO network and to run the processes. The table on the left shows the MATSim link attributes.*

is explained.
First, the hybridPY library is used to import road network data from OpenStreetMap (OSM). Secondly, the mesoscopic network data is imported. This step involves projecting the mesoscopic network into the coordinate system defined by the microscopic simulation, see Sec. 2.4.1. Thirdly, the configuration settings required for running the MATSim simulation are loaded into the hybridPY Scenario. This includes parameters related to simulation duration, agent behavior, scoring functions, mode choice models, and other simulation-specific settings. Additionally, the population data is linked to the hybridPY Scenario. This population data typically includes individual agents' or travelers' demographics, activity schedules, origin-destination pairs, and mobility preferences. Afterward, the MATSim scenario can be run. The MATSim simulation uses the imported network, configuration, and population data. During this step, MATSim simulates the travel behavior of individual agents based on the defined parameters and generates output files containing simulation results. After the MATSim simulation is complete, the output files generated by MATSim are imported into the hybridPY scenario. For reasons of efficiency only population data form the MATSim output inside the SUMO area is imported. The demand import can be done in two different ways, the trip-based or the population-based approach, see Sec. 2.4.3 and Sec. 2.4.2, respectively.

Finally, SUMO is used to simulate traffic dynamics and vehicle movements based on the imported data from MATSim.

2.3 Demand representation

The transport demand in MATSim is represented in a plan-XML file, which contains the detailed travel plans of every single person in the population; see Fig. 5. The demand of BEAM CORE can also be converted into this MATSim compatible format. A plan of a person consists of two elements: "activities" and "legs" of a multi-modal trip. The activities contain information about the location, type and timing, while the legs contain

```
<person id="4">
  <plan score="242.173886086" selected="yes">
    <activity type="Home" x="567919.6149779453" y="4189907.8363222484"
              end_time="15:17:06"> </activity>
    <leg mode="car" dep_time="15:17:06" trav_time="00:06:44">
      <route type="links" start_link="62009" end_link="45050"
              distance="9327" trav_time="00:06:44">31025 108348 ... 73295</route>
    </leg>
    <activity type="othdiscr" x="566908.2465720219" y="4183217.0123480107"
              start_time="15:23:50" end_time="16:10:04"> </activity>
    <leg mode="car" dep_time="16:10:04" trav_time="00:15:04">
      <route type="links" start_link="45050" end_link="120216"
              distance="5023" trav_time="00:15:04">45051 73294 ... 89906</route>
    </leg>
    <activity type="social" x="565895.0902456312" y="4184989.135202894"
              start_time="16:25:08" end_time="16:30:57"> </activity>
    <leg mode="car" dep_time="16:30:57" trav_time="00:08:59">
      <route type="links" start_link="112191" end_link="62008"
              distance="7082" trav_time="00:08:59">112043 78048 ... 31024</route>
    </leg>
    <activity type="Home" x="568031.3048479672" y="4189448.878315132"
              start_time="16:39:56"> </activity>
  </plan>
</person>
```


Figure 5. *MATSim plan.xml example with the daily activity and mobility plan of a single person.*

information about the route (as a sequence of link IDs), the transport mode and timing. Legs in MATSim are almost identical to the "stages" of a plan in SUMO.

2.4 Processes

The following part describes the core elements of the framework. These functions realize the data transfer from MATSim/BEAM CORE to SUMO.

2.4.1 MATSim/BEAM CORE network import

The networks of the two mesoscopic simulators, MATSim and BEAM CORE can be imported. Both simulators use almost identical network attributes, the only difference is that MATSim uses XML and BEAM CORE prefers a CSV type of file. The importer takes also care of the coordinate projection between MATSim/BEAM CORE and SUMO. Apart from the node IDs, link IDs and location, the following link- attributes are read: length, free-speed, capacity, lane number, and transport modes which have access. Note that the capacity is the pre-defined capacity for each link, which needs to be scaled down in case the demand is scaled down for computational speed, as mentioned in the introduction.

2.4.2 Population import

The population import generates a synthetic population in hybridPY **SUMOPy** based on the information contained in the MATSim plan.xml file, see Sec. 2.3. The population is only created in the area covered by the SUMO network. In particular, each activity of a person needs to be associated with a facility (building, area, etc.). The facilities are typically imported as polygons in the area of the SUMO road network. During the parsing of the MATSim plan.xml, only the plans with a chain of activities of at least two activities located within the SUMO-network are considered. Also the begin time and end time of imported activities can be specified to filter activities.

Once an activity chain is filtered, the respective person ID, the transport mode of the legs and the activity location and times are imported and added to the synthetic population in SUMOPy. From the location of each activity, the nearest facility and the nearest SUMO network edge are identified. From the synthetic population, hybridPY generates plans accordingly: each facility (and hence activity) is associated with a SUMO-edge and edge-position. The plan consists basically of a multi-modal routing between these facility edges, where the choice of modes are according to the attributes of the imported plans. In this way the plans contain SUMO-edges instead of MATSim edges. Note that, at this development stage, there is no perfect match between the MASim links and SUMO edges because only the activity locations are used from BEAM CORE, routes of the plans. After this step, the synthetic population can be simulated.

2.4.3 Trip import

SUMO trips are defined as vehicles that travel at a given time from an edge of origin to an edge of destination. In contrast with plans, trips in SUMO do not contain persons.

The trips are extracted from the MATSim plans (see Sec. 2.3) to generate the external trips when using the population import or to generate trip-based microscopic simulation scenarios. This means the MATSim legs from outside the SUMO network to inside and vice versa are represented by SUMO trips. In addition, MATSim legs which cross the SUMO area do also generate SUMO trips. Finally, a time window can be defined to filter the legs to be imported.

For each imported leg, the SUMO edges are identified where the respective MATSim route enters or leaves the SUMO area. For this purpose, all ”fringe edges” of the SUMO network are identified. Fringe edges are all edges that are at the border of the SUMO network, typically there is a turnaround or a dead end node at the end of a fringe edge.

After the entering or exiting SUMO fringe edges are identified for each leg, the respective SUMO route is found by any of the routing methods from the SUMO toolbox. Note that the MATSim route is only used to determine the entering or exiting fringe edge of the SUMO network. Again a more precise matching between the MATSim route and the SUMO route could be realized by a direct mapping from MATSim to SUMO egdes.

2.4.4 Run MATSim

A basic MATSim scenario is defined by a network, a population file (see section 2.3), and a configuration. The configuration specifies simulation-specific parameters, like the number of runs as well as scoring and replanning parameters. hybridPY allows users to import and adjust a template configuration to the specific run. Additionally, a population file is linked to the scenario. For execution, the configuration and the network are provided by hybridPY. The population file is copied according to the specified link.

3 Example Applications

The capability of hybridPY is demonstrated by two example applications: the first scenario is the SUMO simulation of Schwabing, Germany, which uses the demand generated by the MATSim model of the larger Munich area. The second scenario is the

SUMO simulation of San Francisco municipality, USA, with the demand generated by the mesoscopic BEAM CORE model of the entire San Francisco Bay area.

3.1 The San Francisco scenario

This study analyze a BEAM CORE transport system daily scenario calibrated and validated for the whole San Francisco Bay Area, see Fig. 6, in a way to properly simulate the trips in San Francisco generated or/and directed outside, and traveling through the city. The BEAM CORE Bay Area Scenario consist of simulating only 10% of the active population, as explained on Section 1.2.2, which translate on simulating roughly 650,000 agents making 4 trips per day. The mode split is as following: 51.1% used a car alone, 30.9% shared a car trip with other people, 9.3% walked, 1.6% used a bike, 6.0% used transit and 1.0% used ride hail. The vehicles traveled 25.7 millions kilometers in 888,000 hours, consuming 63.6 TJ of propulsion energy, while agents traveled 26.2 millions kilometers in 910,000 hours. The network imported from OSM and successively simplified contains 182,000 links and 73,000 nodes, for a total of 46,000 kilometers. Additional information of the data used for the BEAM CORE sub-models calibration and validation can be found in the relative report [10].

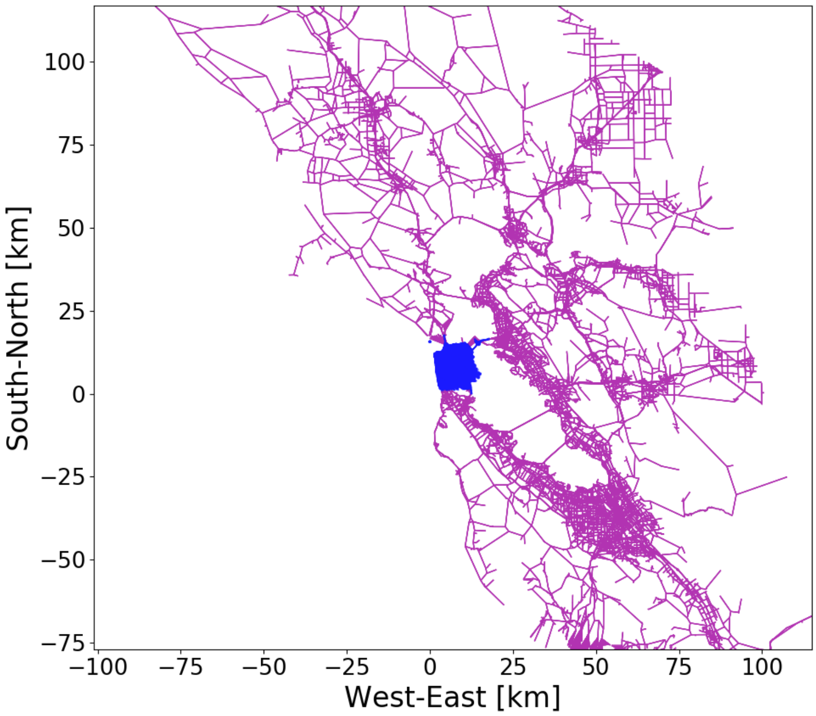
Regarding the microsimulation model, the SUMO network has been converted from OSM and manually improved. The public transport has been created from the GTFS of the San Francisco Municipal Transportation Authority[27]. The BEAM CORE mesoscopic network has been imported into hybridPY as descibed in 2.4.1. From the BEAM CORE database of the synthetic population, a MATSim compatible plan XML file has been generated by a converion script and imported with the process from 2.4.2. From the same plan file, the external trips are extracted with the method from 2.4.3.

3.2 The Munich scenario

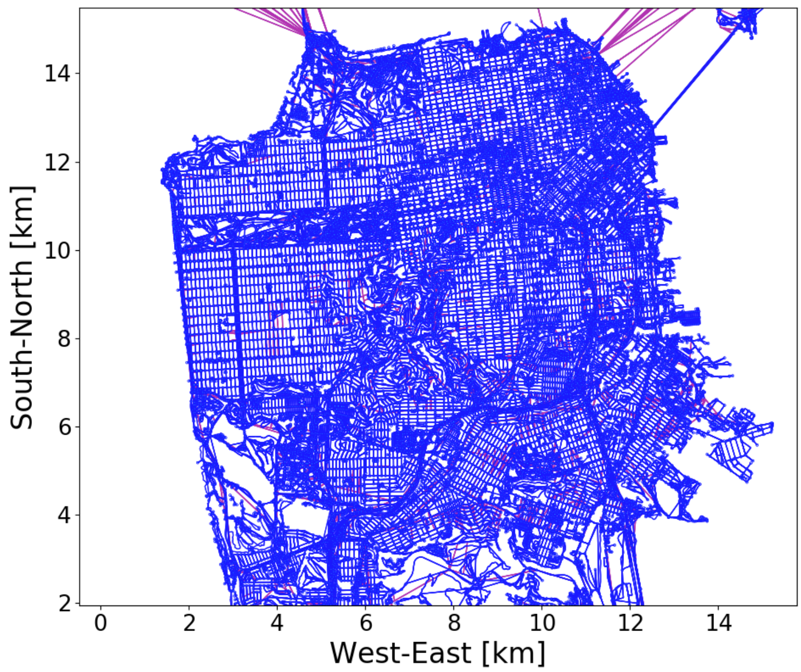
Moeckel et al. [28] proposed a hybrid travel demand model combining trip and activity-based approaches, termed Microsimulation Transport Orchestrator (MITO). The framework comprehensively addresses mobility across diverse areas, encompassing urban and rural regions, while focusing on individual-level trip generation.
However, to streamline model complexity, compromises were made in trip assignment, potentially leading to inconsistencies in spatial and temporal representation at the personal level.
The greater Munich MATSim network and the Schwabing network are shown in Fig. 7. Both networks were converted from OSM. Additionally, the SUMO network has been manually improved.
Inside the mesoscopic simulation, car, public transport, walk, and bike modes are considered. Nevertheless, only the mode car is simulated all other modes are teleported. This means that the travel time of the teleported modes is determined by the distance as the crow flies multiplied by a constant factor.
Due to the nature of MITO, the trip-based hybridPY demand import is used to transfer the MATSim demand to the microscopic level. To create comparability between the microscopic and the mesoscopic simulation, a car-only microscopic scenario is created.

3.3 Comparison of simulation execution times and discussion

The execution times of the mesoscopic models of the two scenarios are summarized in Tab. 1, while the execution times of the microscopic models of the two scenarios is summarized in Tab. 2. For bench marking (except for the BEAM CORE runs) a



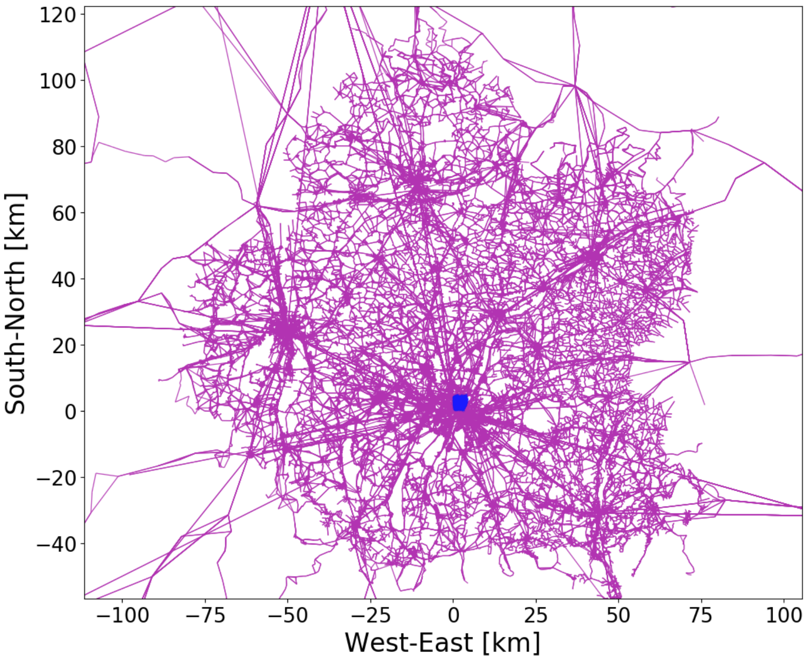
(a)



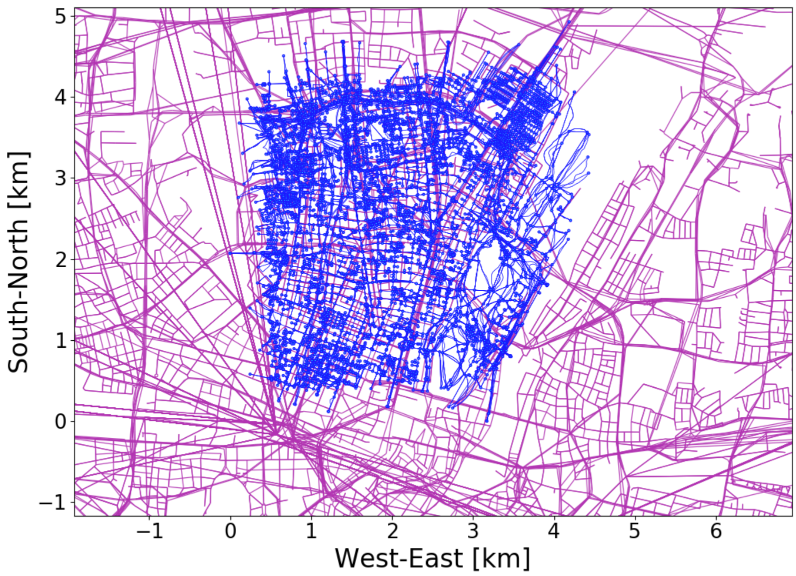
(b)

Figure 6. *Visualization of the BEAM CORE transport network of SF bay area (in puple) and the SUMO network of San Francisco city (in blue) at different zoom levels.*

workstation with 32 Core CPU, 6.00 GHz Intel i9, with 192 GB on board RAM memory was used.



(a)



(b)

Figure 7. *Visualization of the MATSim transport network of larger Munich (in purple) and the SUMO network of Schwabing (in blue) at different zoom levels. Note that main railway links and motorways to major Bavarian cities are also modelled, but not shown in (a).*

SF Bay Area: A 24h BEAM CORE run of the SF Bay Area with 10% population in a 96vCPU, 48 Core, 768Gb memory, takes around 1 hour. A 30% population scenario takes approximately 3 hours. A BEAM CORE run with 6 BEAM CORE iterations takes around 3 hours with a 10% population and 24 hours in a 30% scenario.

Table 1. *Comparison of simulation execution times necessary to establish the user-equilibrium of the two mesoscopic models.*

Scenario	**Larger Munich**	**San Francisco Bay area**
Execution time	19 h	24 h
Full population size	$4\,400\,000$	$6\,510\,000$
$\%$ of simulated population	30%	30%
Number of car trips	$1,700,000$	$25,110,00$
Iterations for user-equilibrium	50	6

Table 2. *Comparison of simulation execution times of the two microscopic models for a single simulation run.*

Scenario	**Schwabing**	**San Francisco city**
Simulated time	4h	10h
Scale	80%	80%
Number of car trips	$400\,000$	$230\,000$ ext. trips, $427\,712$ population

The 24h microscopic simulation of San Francisco City only took approximately 10 hours.

Munich: A MATSim iteration simulating 24h of the Munich Metropolitan Area with 30% population takes around 22.8 min per iteration. To reach the equilibrium state, 50 iterations are simulated. In total, 1.324 million agents are simulated in the metropolitan region, with 3.9 legs on average per person.
The mesoscopic level achieved a real-to-sim time ratio of 63.2, while the microscopic level achieved a real-to-sim time ratio of only 3. The execution of the microscopic simulation took 4 hours. Discussing the Munich example the ratio between mesoscopic and microscopic is 20. Assuming that the microsimulation execution time grows linearly with the population size, the microsimulation execution time of the 24h scenario of greater Munich would take at least 113.2 hours.
To summarize, switching from mesoscopic to microscopic simulations results in a significant slowdown. Additionally, the ratio only represents the loss in terms of simulation time and does not reflect the time needed for adjusting the microscopic network.

4 Conclusions

This paper presents a hybrid modeling suite for the mesoscopic and microscopic traffic simulations: MATSim and SUMO. The limitation of current hybrid simulation approaches of creating the microscopic simulation network can be overcome by providing a modeling suite with a graphical user interface. The simulation framework allows mesoscopic, microscopic, and hybrid simulations to run.
The results of the case studies show that switching from mesoscopic to microscopic simulation level results in a significant slowdown in terms of real-to-sim time ratio.

It is apparent that both scenarios (Larger Munich and San Francisco Bay area) cannot be micro-simulated in a reasonable time with the current processor technology – in particular if iterative simulations runs are required, for example to find the user equilibrium.

For future works, it is planned to improve the matching between MATSim links and SUMO edges, by an automated map matching of the simulation graphs and the implementation of the route transfer from MATSim to SUMO. In this way the microsimulation could profit of already assigned routes, which would avoid to determine a user-equilibrium with many iterations on the microsimulation side. A better matching of links enables also to transfer average link travel times and flows from the microscopic simulation into the mesoscopic graph model — either to improve the traffic assignment on the mesoscopic side or to validate link costs and flows. Such a direct comparison of link results could ultimately overcome the deficiencies of microscopic and mesoscopic simulations, as explained in the introduction.

A further enhancement would be to improve the usability of the mesoscopic simulation module by supporting the editing of public transport networks. Additionally, the gap between the hybrid scenario and reality could be quantified using measured data.

Data availability statement

The code of hybridPY is open sorce and available on github: https://github.com/TUMFTM/hybridPY-public. The application is provided in a containerized version to ensure easy usage.

Author contributions

Jörg Schweizer contributed to the paper in the following ways: conceptualization, methodology, data curation, formal analysis, visualization, software, and writing the original draft. Fabian Schuhmann contributed to the paper in the following ways: conceptualization, methodology, data curation, formal analysis, visualization, software, and writing the original draft. Cristian Poliziani contributed in the following ways: software and reviewing and editing the manuscript.

Competing interests

The authors declare that they have no competing interests.

References

[1] Meister, M. Balmer, F. ari, *et al.*, "Large-scale agent-based travel demand optimization applied to switzerland, including mode choice," Jul. 2010.

[2] "The multi-agent transport simulation matsim," in A. Horni, K. Nagel, and K. W. Axhausen, Eds. Gordon House, 29 Gordon Square, London WC1H 0: biquity Press Ltd., 2016, ch. The "Multi-Modal" Contribution, ISBN: 78-1-909188-76-1. DOI: http://dx.doi.org/10.5334/baw.

[3] M. Maciejewski and K. Nagel, "Towards multi-agent simulation of the dynamic vehicle routing problem in matsim," in *Parallel Processing and Applied Mathematics*, R. Wyrzykowski, J. Dongarra, K. Karczewski, and J. Wasniewski, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 551–560, ISBN: 978-3-642-31500-8.

[4] E. Nassar, "Integrating dynamic ride-sharing in matsim: Assessing the impacts of dynamic ride sharing on the environment. the case study of upper austria," English, M.S. thesis, Technical University of Munich, Jun. 2023.

[5] D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker, “Recent development and applications of SUMO - Simulation of Urban MObility,” *International Journal On Advances in Systems and Measurements*, vol. 5, no. 3&4, pp. 128–138, Dec. 2012.

[6] Y. Huang, K. M. Kockelman, V. Garikapati, L. Zhu, and S. Young, “Use of shared automated vehicles for first-mile last-mile service: Micro-simulation of rail-transit connections in austin, texas,” *Transportation Research Record*, vol. 2675, no. 2, pp. 135–149, 2021. DOI: 10.1177/0361198120962491. eprint: https://doi.org/10.1177/0361198120962491. [Online]. Available: https://doi.org/10.1177/0361198120962491.

[7] L. Koch, D. S. Buse, M. Wegener, *et al.*, “Accurate physics-based modeling of electric vehicle energy consumption in the sumo traffic microsimulator,” in *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, 2021, pp. 1650–1657. DOI: 10.1109/ITSC48978.2021.9564463.

[8] P. Pau, K.-H. Kastner, R. Keber, and M. Samal, “Real-time traffic conditions with sumo for its austria west,” May 2013, ISBN: 978-3-662-45078-9. DOI: 10.1007/978-3-662-45079-6_11.

[9] P. Pau and K.-H. Kastner, “Experiences with sumo in a real-life traffic monitoring system,” May 2015.

[10] C. A. Spurlock, M. A. Bouzaghrane, A. Brooker, *et al.*, “Behavior, energy, autonomy & mobility comprehensive regional evaluator: Overview, calibration and validation summary of an agent-based integrated regional transportation modeling workflow,” Berkeley, Tech. Rep., Feb. 2024. [Online]. Available: https://eta-publications.lbl.gov/publications/behavior-energy-autonomy-mobility.

[11] H. Laarabi, Z. Needell, R. Waraich, C. Poliziani, and T. P. Wenzel, “A modeling framework for behavior, energy, autonomy and mobility (beam),” Tech. Rep., Feb. 2024. [Online]. Available: https://eta-publications.lbl.gov/publications/behavior-energy-autonomy-mobility.

[12] R. Nazanin, T.-B. Annika, F. Sydny K., *et al.*, “At the nexus of equity and transportation modeling: Assessing accessibility through the individual experienced utility-based synthesis (inexus) metric,” *Journal of Transport Geography*, vol. 115, 2024. DOI: 10.1016/j.jtrangeo.2024.103824.

[13] W. Charlton and B. Sana, “Simwrapper, an open source web-based platform for interactive visualization of microsimulation outputs and transport data,” *Procedia Computer Science*, vol. 220, pp. 724–729, 2023, The 14th International Conference on Ambient Systems, Networks and Technologies Networks (ANT) and The 6th International Conference on Emerging Data and Industry 4.0 (EDI40), ISSN: 1877-0509. DOI: https://doi.org/10.1016/j.procs.2023.03.095. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050923006300.

[14] A. Neumann and M. Zilske, *MATSim JOSMNetwork Editor*. Technische Universität Berlin, 2018.

[15] P. Barbecho Bautista, L. F. Urquiza-Aguiar, and M. Aguilar Igartua, “Stgt: Sumo-based traffic mobility generation tool for evaluation of vehicular networks,” in *Proceedings of the 18th ACM Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, & Ubiquitous Networks*, ser. PE-WASUN ’21, Alicante, Spain: Association for Computing Machinery, 2021, pp. 17–24, ISBN: 9781450390781. DOI: 10.1145/3479240.3488523. [Online]. Available: https://doi.org/10.1145/3479240.3488523.

[16] P. P. Garrido Abenza, M. Malumbres, P. Piñol, and O. López Granado, “A simulation tool for evaluating video streaming architectures in vehicular network scenarios,” *Electronics*, vol. 9, p. 1970, Nov. 2020. DOI: 10.3390/electronics9111970.

[17] L. Papaleondiou and M. Dikaiakos, “Trafficmodeler: A graphical tool for programming microscopic traffic simulators through high-level abstractions,” May 2009, pp. 1–5. DOI: 10.1109/VETECS.2009.5073891.

[18] W. Arellano and I. Mahgoub, “Trafficmodeler extensions: A case for rapid vanet simulation using, omnet++, sumo, and veins,” *2013 High Capacity Optical Networks and Emerging/Enabling Technologies*, pp. 109–115, 2013. [Online]. Available: https://api.semanticscholar.org/CorpusID:14090546.

[19] M. Gütlein and A. Djanatliev, “Coupled traffic simulation by detached translation federates: An hla-based approach,” in *2019 Winter Simulation Conference (WSC)*, 2019, pp. 1378–1389, ISBN: 1558-4305. DOI: 10.1109/WSC40007.2019.9004809.

[20] M. Gütlein, R. German, and A. Djanatliev, “Towards a hybrid co-simulation framework: Hla-based coupling of matsim and sumo,” in *2018 IEEE/ACM 22nd International Symposium on Distributed Simulation and Real Time Applications (DS-RT)*, 2018, pp. 1–9, ISBN: 1550-6525. DOI: 10.1109/DISTRA.2018.8601004.

[21] M. Gütlein and A. Djanatliev, “Coupled traffic simulation by detached translation federates: An hla-based approach,” in *2019 Winter Simulation Conference (WSC)*, 2019, pp. 1378–1389. DOI: 10.1109/WSC40007.2019.9004809.

[22] H. Triebke, *Calibrating spatio-temporal network states in microscopic traffic simulation on a global level: Youtube video - technische präsentation @ sumo user conference*. [Online]. Available: https://www.youtube.com/watch?v=QzrkIrEllTI.

[23] H. Triebke, M. Kromer, and P. Vortisch, “Pre-study and insights to a sequential matsim-sumo tool-coupling to deduce 24h driving profiles for saevs,” *SUMO Conference Proceedings*, vol. 1, pp. 93–112, 2022. DOI: 10.52825/scp.v1i.103.

[24] Henriette Triebke, Markus Kromer, and Peter Vortisch, “Bridging the gap between mesoscopic transport planning and microscopic traffic simulation: An analytical and numerical analysis of traffic dynamics,” *Transportation Research Record*, vol. 0, no. 0, p. 03 611 981 221 128 284, 0, ISSN: 0361-1981. DOI: 10.1177/03611981221128284.

[25] K. Schrab, R. Protzmann, and I. Radusch, “A large-scale traffic scenario of berlin for evaluating smart mobility applications,” in *Smart Energy for Smart Transport*, ser. Lecture Notes in Intelligent Transportation and Infrastructure, E. G. Nathanail, N. Gavanas, and G. Adamos, Eds., Cham: Springer Nature Switzerland, 2023, pp. 276–287, ISBN: 978-3-031-23720-1. DOI: 10.1007/978-3-031-23721-8{\textunderscore}24.

[26] Christian Rakow, Ihab Kaddoura, Chengqi Lu, Ronald Nippold, and Peter Wagner, “Investigation of the system-wide effects of intelligent infrastructure concepts with microscopic and mesoscopic traffic simulation,” 2021.

[27] S. F. M. T. Authority, *Gtfs of the san francisco public transportation*, Last accessed Feb 2024. [Online]. Available: https://transitfeeds.com/p/sfmta/60.

[28] R. Moeckel, N. Kuehnel, C. Llorca, A. T. Moreno, H. Rayaprolu, and F. Galante, “Agent-based simulation to improve policy sensitivity of trip-based models,” *Journal of Advanced Transportation*, vol. 2020, p. 1 902 162, 2020, ISSN: 0197-6729. DOI: 10.1155/2020/1902162.