SUMO User Conference 2026

TIBconferencesession

TIB-OP_will_set_DOI

Towards Smarter Intersections Through Digital Twins - The Coupling of SUMO and SCENIMINI

Lars Klitzke[1], Yun-Pang Flötteröd[2]*, and
Peter Wagner[2]

[1]Institute of Transportation Systems, German Aerospace Center, Braunschweig, Germany

[2]Institute of Transportation Systems, German Aerospace Center, Berlin, Germany

*Correspondence: Yun-Pang Flötteröd, yun-pang.floetteroed@dlr.de

Abstract. This paper presents a framework for coupling SUMO with a Scenario Mining Platform (SCENIMINI) to enhance the early detection of critical traffic situations and improve vehicle behavior modeling at intersections. The proposed method enables high-resolution data integration from infrastructure sensors, followed by data transformation, synchronization, and bidirectional communication between simulation and visualization environments. A proof-of-concept demonstration shows the integration of real and simulated traffic data in SCENIMINI for qualitative validation of the coupling approach. Several challenges related to online system integration are identified as directions for future work.

Keywords: smart intersection, digital twins, SCENIMINI, SUMO

1. Introduction

Intersections are among the most dynamic and risk-prone elements within road network infrastructure. By integrating real-time trajectory data from sensors, environmental conditions, historical patterns and traffic simulation into a digital twin, potential critical locations can be easier identified. This includes conflict zones, near miss hot-spots/events that do not appear in crash data, or locations that exhibit recurring safety concerns. This level of digital insight provides a richer, data-driven understanding about real-world driving maneuvers at intersections, and enhances proactive interventions from redesigning geometry to adjusting signal timing or deploying targeted safety measures. In essence, such digital twins facilitate cities to move from reactive problem-solving to data driven prevention, enhancing intersection safety and resilience.

Various sensors and cameras have been deployed at DLR's Research Intersection (RI) in Braunschweig, Germany, and not only the trajectory data of road users, but also data related to weather, traffic signals and emissions have been collected [1]. The trajectory data are collected at 20 Hz, whereas the other data are recorded once per second. The RI has supported numerous research projects spanning multiple domains, including microscopic vehicle and traffic modeling, road users' behaviors, traffic safety, automated and autonomous vehicle systems, emission monitoring, and environmental impact studies. SCENIMINI (Scenario Mining Platform) was developed at the German Aerospace Center (DLR) with the objective of providing a systematic framework for transforming stationary collected traffic data into structured traffic scenarios and supporting the analysis of traffic dynamics within these scenarios. The platform establishes a technological foundation that enables the consistent representation, processing, and evaluation of infrastructure-based traffic measurements. To achieve this objective, SCENIMINI provides a comprehensive set of tools, interfaces, and data models. These components facilitate standardized data ingestion, storage, processing, and access, thereby enabling interoperable and extensible workflows. The modular design allows additional components to be flexibly developed or adapted, e.g., visualization modules tailored to specific research questions and use cases. Among the supported use-cases are the analysis of historical traffic data for the identification of atypical or potentially critical traffic scenarios [2], [3], [4]. Such analyses contribute to safety assessment, traffic management evaluation, and the derivation of representative scenarios for simulation-based studies. In addition, SCENIMINI supports the processing and analysis of live traffic data, enabling near real-time monitoring and scenario generation.

To facilitate effective real-time traffic management and the establishment of digital twins, SUMO[5], [6] has been integrated with a range of traffic signal controllers and simulation platforms. This integration allows dynamical, real-time optimization of traffic signal plans, supports emergency and on-demand vehicle dispatch planning, and serves as a backend system for providing network-wide, real-time traffic and emission conditions through the integration of diverse sensor data sources (e.g., [7], [8], [9], [10]). However, direct integration of high-resolution trajectory data collected at intersections for safety analysis remains unaddressed.

This paper focuses on the coupling of SUMO with SCENIMINI to support earlier detection of critical situations by leveraging simulation capabilities, as well as enhancing the accuracy of modeling vehicle behaviors when approaching and traveling through intersections. This paper proposes a structured process for coupling SCENIMINI with SUMO. The process encompasses data acquisition from stationary infrastructure sensors, transformation and synchronization of vehicle-related information, and runtime interaction with the simulation environment. Particular attention is given to the integration mechanisms that enable bidirectional communication and consistent state updates between both systems. As a proof of concept, a visualization of real and simulated traffic data within SCENIMINI is presented. This visualization enables a direct comparison of observed and simulated vehicle movements and serves as an initial qualitative validation of the coupling approach. It demonstrates the feasibility of embedding simulation results into the SCENIMINI platform and provides a basis for subsequent quantitative evaluation and calibration.

2. SCENIMINI

The processing of infrastructure-based traffic data poses specific technical and methodological challenges. In contrast to static datasets, such data are typically generated as a continuous stream and must be processed either with minimal latency or in real time to support live applications. This requirement calls for scalable, distributed architectures capable of handling high data throughput while ensuring robustness and extensibility. SCENIMINI was designed to achieve this goal. In the following, we will briefly describe the architecture of SCENIMINI and some key components for the sake of clarity (for more details, see [11, Chapter 6]).

2.1 Architecture

Scenario-based representations of traffic data enable structured analysis of complex traffic dynamics and provide an abstraction layer that facilitates downstream applications such as safety assessment, system validation, and long-term traffic analysis. For scenario-based traffic data analytics, the modular platform SCENIMINI was designed (see Figure 1). SCENIMINI follows a layered and modular architecture structured around three main components:

1. *Data Storage and Communication*
2. *Data Processing and Integration*
3. *Applications and Use cases*

In addition, a fourth cross-cutting component named *Infrastructure, Monitoring, and GitOps* extends across all three primary components.

The *Data Storage and Communication* layer comprises components whose primary responsibility is to manage data and make it accessible to the other layers of the system architecture. This includes persistently stored data in databases and network-attached storage, as well as transient data required for event-driven streaming applications. At the top of the architecture, the *Applications and Use-Cases* layer contains domain-specific applications that build upon the provided data services. These applications typically feature graphical user interfaces and are designed to support end users in monitoring, analysis, and decision-making tasks. They rely on standardized interfaces and processed data streams supplied by the underlying layers. Positioned between these two layers is the *Data Processing and Integration layer*. This layer hosts components responsible for transforming, aggregating, and enriching raw data before it is exposed to applications in a suitable format (e.g. defined in TASI). Typical tasks include data validation, synchronization, format conversion, and the integration of heterogeneous data sources.

Since SCENIMINI is designed as a modular and distributed system, all components across these layers are deployed as independent services within a Kubernetes environment. This containerized architecture supports scalability, resilience, and flexible orchestration of services. This is achieved by using Helm Charts for the configuration and parameterization of individual components and their integration. These charts are version-controlled in a GitLab instance and deployed to the respective Kubernetes clusters using ArgoCD. This architectural design ensures operational stability, reproducibility, and maintainability throughout the entire application life cycle, from development and deployment to monitoring and iterative refinement.

2.2 Scenario Mining Pipeline

A core element of SCENIMINI is the concept of Scenario Mining Pipelines (SMPs). An SMP is a processing chain whose primary task is to transform raw traffic data into structured scenario representations. [11] Each SMP is implemented as a distributed application composed of multiple microservices. This microservice-based design promotes modularity, scalability, and independent deployment of functional components.

Within an SMP, individual services are responsible for well-defined processing tasks. These may include, for example, trajectory smoothing, map matching of traffic participants to a digital road network, computation of criticality metrics (TTC, PET, etc.), or other domain-specific analytical operations. By decomposing the overall processing

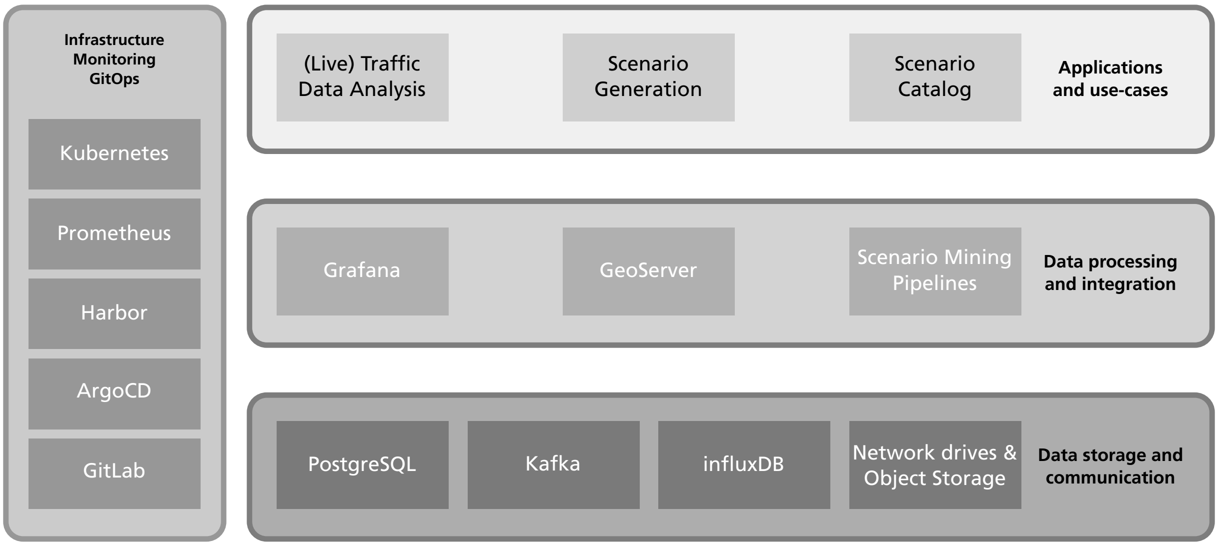

Figure 1. *Excerpt of SCENIMINI's architecture (adapted version based on [12]).*

workflow into specialized services, the system achieves high flexibility and adaptability to varying data sources and analytical requirements.

To enable flexible configuration of SMPs for specific use cases, data exchange between services is realized through a message-oriented middleware (MoM). In the current version, Apache Kafka is employed as the underlying communication backbone. The use of a MoM ensures loose coupling between services, supports horizontal scalability, and allows asynchronous communication patterns. Furthermore, it facilitates the efficient dissemination of enriched traffic data or even processed scenario data to downstream applications as illustrated in Figure 1. These applications include, among others, real-time visualization and analysis of live traffic data, automated generation of scenarios for scenario-based testing, and the creation of structured scenario catalogs for long-term studies.

2.3 TASI

Another central component of SCENIMINI, not depicted in Figure 1, is TASI [13]. TASI is an Open Source Python library designed for macroscopic traffic data analysis. While SCENIMINI primarily provides the distributed system architecture and processing infrastructure, TASI complements this framework by offering domain-specific analytical functionality and standardized data models.

TASI provides a comprehensive set of functions for the analysis of traffic data, particularly of trajectory data. These functions support tasks such as traffic flow analysis, statistical evaluation, and the derivation of performance indicators, such as Surrogate Measures of Safety, relevant to macroscopic traffic assessment. In addition, TASI defines structured data models for describing traffic-related information in Python. These models are implemented using pydantic [14], thereby ensuring type safety, validation, and consistent data serialization. An overview of relevant models defined in TASI is given in Figure 2. The `TrafficParticipant` contains information about a traffic participant such as a unique identifier, its dimension and object type. `Pose` represents the state of a traffic participant for a time instant, a `Trajectory` the history of a `TrafficParticipant` and the `PoseCollection` the poses of multiple traffic participants for a time instant.

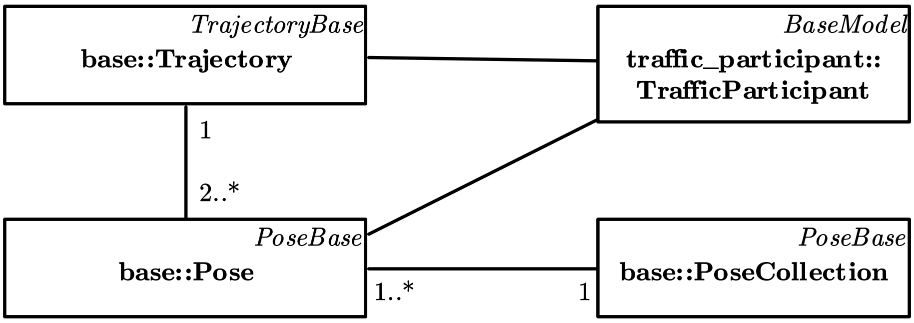

Figure 2. *Excerpt of relevant data models for representing traffic participants in TASI.*

The data models provided by TASI are also employed within SMPs. In this context, they serve as standardized message schemas for data exchange between microservices. By relying on shared, formally defined models, interoperability between services is enhanced, and ambiguities in data representation are minimized. This approach contributes to the robustness and maintainability of the overall system, particularly in distributed environments where loosely coupled services communicate asynchronously.

Beyond its analytical and modeling capabilities, TASI offers integrated access to publicly available traffic datasets. These include data from the RI [1] as well as from the Test Bed Lower Saxony [15]. By providing programmatic access to these datasets, TASI facilitates reproducible research and accelerates the development and evaluation of traffic analysis methods.

3. Coupling implementation

The modular structure of SCENIMINI enables the development and integration of additional components tailored to specific application requirements. For the coupling of SCENIMINI and SUMO, the existing components introduced in Figure 1 are reused and extended, which we will discuss in the following.

3.1 Concept and requirements

The primary objective of the presented integration is to perform a visual comparison between real-world traffic participants and their counterparts simulated in SUMO. To this end, both real and simulated vehicles are displayed within a browser-based dashboard, enabling a direct, time-synchronized comparison of observed and simulated traffic states. Such visualization provides an intuitive means of validating the coupling mechanism and assessing the consistency between infrastructure-based measurements and simulation results.

Because the traffic information collected at the RI is continuously provided via Apache Kafka, the incoming data stream must be processed sequentially for simulation with SUMO. An overview of the overall process is illustrated in Figure 3 as a UML sequence diagram for two time steps. A Kafka message from the infrastructure represents the state of the detected traffic participants for a specific point in time encoded as `PoseCollection`. The *SUMO connector* ensures that vehicles are introduced into the simulation as soon as they are detected. Once inserted, control over their longitudinal and lateral motion is transferred to SUMO. From that point onward, the connector primarily supervises object type identification, driving/moving parameter extraction, edge/lane matching, route assignment and, if necessary, performs route corrections to ensure consistency with the observed trajectory. Regarding object positions, we use the `convertLonLat2XY` method of the SUMO `Net` class for coordinate transformation. The position data are originally encoded in EPSG:32632; we first convert them to EPSG:4326 (WGS 84) and then transform them into the coordinate reference system defined by the SUMO network using the `convertLonLat2XY` method. For every simulation step corresponding to the 0.05-second data collection interval, the connector gets the states of all simulated traffic participants, encodes it as `PoseCollection` and produces it into a dedicated Kafka topic for follow-up usage. Section 3.2 presents the implementation of a dedicated service within the SMP to realize this SUMO connector. This service encapsulates the described functionalities: consuming real-time traffic data, managing vehicle insertion and route assignment in SUMO, executing synchronized simulation steps, and publishing the updated simulation states for visualization.

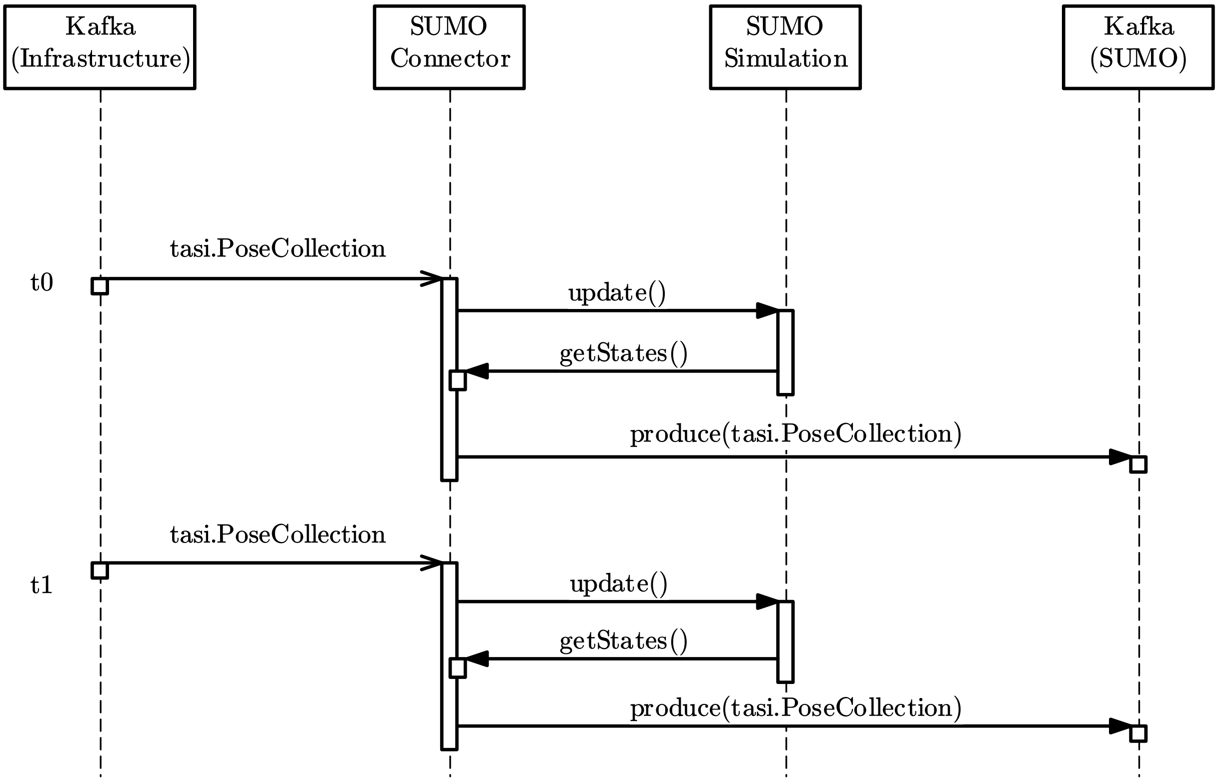

Figure 3. *Overview of the interactions between the SUMO connector service of SCENIMINI and the SUMO simulation.*

3.2 Service overview

The coupling between SUMO and SCENIMINI is technically realized through a dedicated SMP service that acts as an interface between streaming traffic data and microscopic simulation. The objective of this service is to continuously consume TASI-encoded traffic data using the models illustrated in Figure 2 from Apache Kafka and to communicate with a locally running SUMO instance via TraCI (Traffic Control Interface). The corresponding process flow is illustrated in Figure 4 as a UML activity diagram.

Upon initialization of the service, the SUMO simulation instance is started and configured. Moreover, a Kafka consumer and producer are instantiated to enable bidirectional message exchange. That is, the service will receive messages that include information about traffic participants and will send messages that includes that states of the vehicles within the simulation.

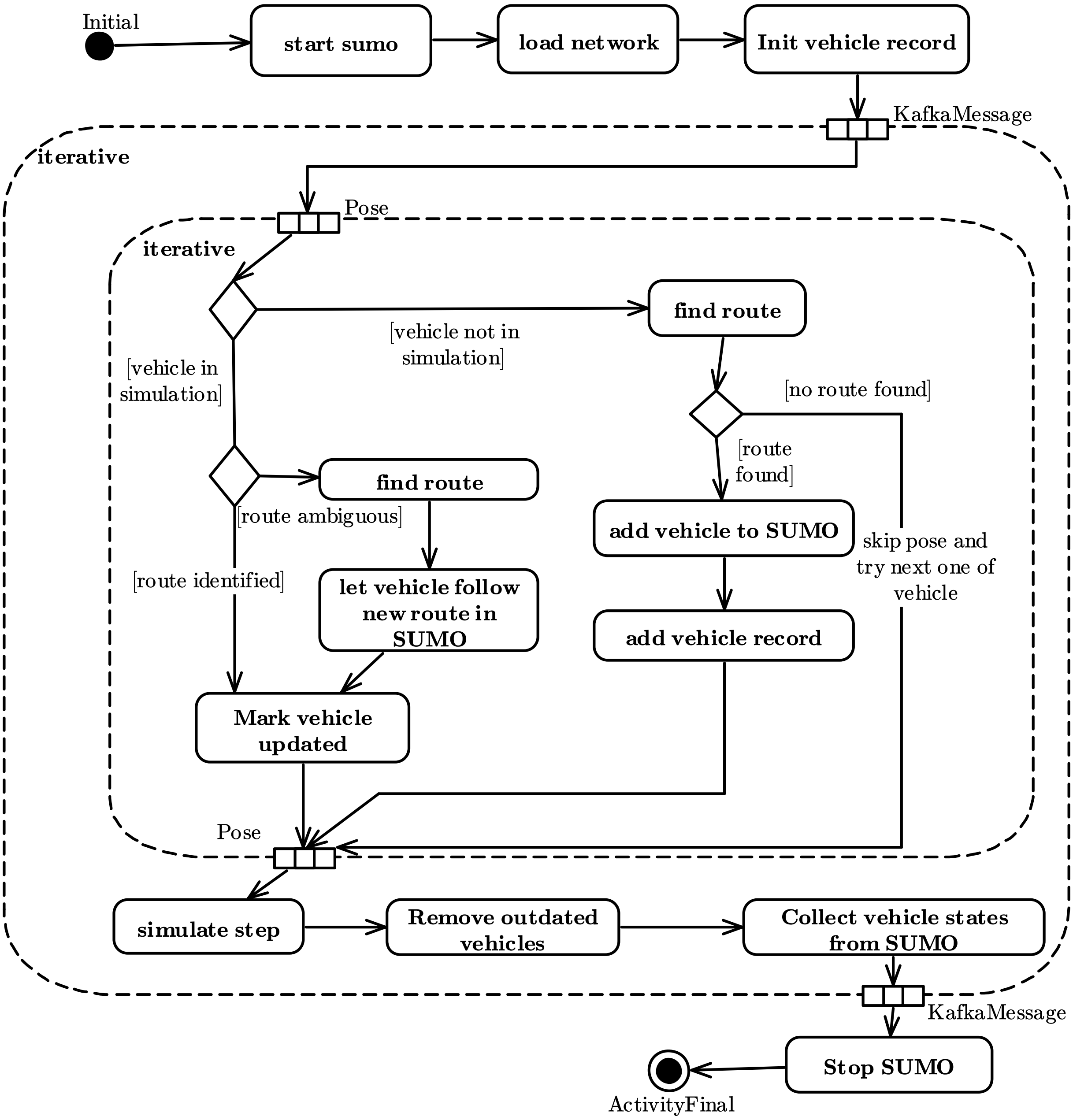

Figure 4. *The process of coupling SUMO with live traffic data in SCENIMINI.*

The service assumes that a Kafka message contains information of all traffic participants for a time instant encoded using the `PoseCollection` TASI model. For each received message, the service iteratively processes the contained poses and checks whether the corresponding vehicle is already present in the simulation. If a vehicle is not yet part of the simulation, its route is determined via TraCI based on its current position within the digital road network. Route determination is also triggered if a vehicle is already present in the simulation but cannot be uniquely associated with a route. Such ambiguity may occur, for example, when a vehicle enters an intersection where multiple potential downstream routes exist. Subsequently, route updates are performed based on the continuously collected poses. If a vehicle is not yet simulated and a valid route can be identified, the vehicle is inserted into SUMO at its current position with its current dynamic properties (e.g., speed) using TraCI. Conversely, if the vehicle is already part of the simulation and its route becomes uniquely identifiable, the service updates SUMO accordingly and assigns the vehicle to the determined route.

Since each `PoseCollection` represents the states of traffic participants at a discrete time step, the service performs a synchronized simulation cycle after all poses of the current message have been processed. This cycle comprises the four sequential steps:

1. Trigger SUMO via TraCI to perform a simulation step
2. Retrieve the updated vehicles states from SUMO via TraCI
3. Transformation of the states to TASI `PoseCollection`
4. Publication of SUMO-based `PoseCollection` to a designated Kafka topic

Through this closed-loop mechanism, the service establishes a continuous synchronization between real-world traffic observations and the microscopic simulation environment.

3.3 Live traffic visualization

For the visual representation of vehicle states originating from both the simulation and the real-world measurements, an existing application was extended accordingly (see Figure 5). The dashboard provides a synchronized view of traffic participants, enabling a direct comparison between infrastructure-detected and simulation-controlled vehicles.

```
Logs(main-smp-core/smp-core-sumo-connector:sumo-connector)[tail]
smp-core-sumo-connector-6fbc84d8c7-mcld5 [2025-11-04 16:35:08 | app.py:process:193] > WARNING:  Failed to find routes for 1756825381936811
smp-core-sumo-connector-6fbc84d8c7-mcld5 [2025-11-04 16:35:08 | app.py:process:154] > INFO:  Process poses from 2025-09-02 15:03:19.360787
smp-core-sumo-connector-6fbc84d8c7-mcld5 [2025-11-04 16:35:08 | app.py:process:163] > INFO:  sim: 2025-09-02 15:03:19.310787, live: 2025-09-02 15:03:19.360787, offset: 0.05
smp-core-sumo-connector-6fbc84d8c7-mcld5 [2025-11-04 16:35:08 | app.py:process:193] > WARNING:  Failed to find routes for 1756825381936811
smp-core-sumo-connector-6fbc84d8c7-mcld5 [2025-11-04 16:35:08 | app.py:process:154] > INFO:  Process poses from 2025-09-02 15:03:19.410787
smp-core-sumo-connector-6fbc84d8c7-mcld5 [2025-11-04 16:35:08 | app.py:process:163] > INFO:  sim: 2025-09-02 15:03:19.360787, live: 2025-09-02 15:03:19.410787, offset: 0.05
smp-core-sumo-connector-6fbc84d8c7-mcld5 [2025-11-04 16:35:08 | app.py:process:193] > WARNING:  Failed to find routes for 1756825381936811
smp-core-sumo-connector-6fbc84d8c7-mcld5 [2025-11-04 16:35:08 | app.py:process:154] > INFO:  Process poses from 2025-09-02 15:03:19.460787
smp-core-sumo-connector-6fbc84d8c7-mcld5 [2025-11-04 16:35:08 | app.py:process:163] > INFO:  sim: 2025-09-02 15:03:19.410787, live: 2025-09-02 15:03:19.460787, offset: 0.05
smp-core-sumo-connector-6fbc84d8c7-mcld5 [2025-11-04 16:35:08 | app.py:process:193] > WARNING:  Failed to find routes for 1756825381936811
smp-core-sumo-connector-6fbc84d8c7-mcld5 [2025-11-04 16:35:08 | app.py:process:154] > INFO:  Process poses from 2025-09-02 15:03:19.510787
smp-core-sumo-connector-6fbc84d8c7-mcld5 [2025-11-04 16:35:08 | app.py:process:163] > INFO:  sim: 2025-09-02 15:03:19.460787, live: 2025-09-02 15:03:19.510787, offset: 0.05
smp-core-sumo-connector-6fbc84d8c7-mcld5 [2025-11-04 16:35:08 | app.py:process:193] > WARNING:  Failed to find routes for 1756825381936811
smp-core-sumo-connector-6fbc84d8c7-mcld5 [2025-11-04 16:35:08 | app.py:process:154] > INFO:  Process poses from 2025-09-02 15:03:19.560787
smp-core-sumo-connector-6fbc84d8c7-mcld5 [2025-11-04 16:35:08 | app.py:process:163] > INFO:  sim: 2025-09-02 15:03:19.510787, live: 2025-09-02 15:03:19.560787, offset: 0.05
smp-core-sumo-connector-6fbc84d8c7-mcld5 [2025-11-04 16:35:08 | app.py:process:193] > WARNING:  Failed to find routes for 1756825381936811
smp-core-sumo-connector-6fbc84d8c7-mcld5 [2025-11-04 16:35:08 | app.py:process:154] > INFO:  Process poses from 2025-09-02 15:03:19.610787
smp-core-sumo-connector-6fbc84d8c7-mcld5 [2025-11-04 16:35:08 | app.py:process:163] > INFO:  sim: 2025-09-02 15:03:19.560787, live: 2025-09-02 15:03:19.610787, offset: 0.05
smp-core-sumo-connector-6fbc84d8c7-mcld5 [2025-11-04 16:35:08 | app.py:process:193] > WARNING:  Failed to find routes for 1756825381936811
smp-core-sumo-connector-6fbc84d8c7-mcld5 [2025-11-04 16:35:08 | app.py:process:154] > INFO:  Process poses from 2025-09-02 15:03:19.660787
smp-core-sumo-connector-6fbc84d8c7-mcld5 [2025-11-04 16:35:08 | app.py:process:163] > INFO:  sim: 2025-09-02 15:03:19.610787, live: 2025-09-02 15:03:19.660787, offset: 0.05
smp-core-sumo-connector-6fbc84d8c7-mcld5 [2025-11-04 16:35:08 | app.py:process:193] > WARNING:  Failed to find routes for 1756825381936811
smp-core-sumo-connector-6fbc84d8c7-mcld5 [2025-11-04 16:35:08 | app.py:process:154] > INFO:  Process poses from 2025-09-02 15:03:19.710787
smp-core-sumo-connector-6fbc84d8c7-mcld5 [2025-11-04 16:35:08 | app.py:process:163] > INFO:  sim: 2025-09-02 15:03:19.660787, live: 2025-09-02 15:03:19.710787, offset: 0.05
smp-core-sumo-connector-6fbc84d8c7-mcld5 [2025-11-04 16:35:08 | app.py:process:193] > WARNING:  Failed to find routes for 1756825381936811
smp-core-sumo-connector-6fbc84d8c7-mcld5 [2025-11-04 16:35:08 | app.py:process:154] > INFO:  Process poses from 2025-09-02 15:03:19.760787
smp-core-sumo-connector-6fbc84d8c7-mcld5 [2025-11-04 16:35:08 | app.py:process:163] > INFO:  sim: 2025-09-02 15:03:19.710787, live: 2025-09-02 15:03:19.760787, offset: 0.05
smp-core-sumo-connector-6fbc84d8c7-mcld5 [2025-11-04 16:35:08 | app.py:process:193] > WARNING:  Failed to find routes for 1756825381936811
smp-core-sumo-connector-6fbc84d8c7-mcld5 [2025-11-04 16:35:08 | app.py:process:154] > INFO:  Process poses from 2025-09-02 15:03:19.810787
smp-core-sumo-connector-6fbc84d8c7-mcld5 [2025-11-04 16:35:08 | app.py:process:163] > INFO:  sim: 2025-09-02 15:03:19.760787, live: 2025-09-02 15:03:19.810787, offset: 0.05
smp-core-sumo-connector-6fbc84d8c7-mcld5 [2025-11-04 16:35:08 | app.py:process:193] > WARNING:  Failed to find routes for 1756825381936811
smp-core-sumo-connector-6fbc84d8c7-mcld5 [2025-11-04 16:35:08 | app.py:process:154] > INFO:  Process poses from 2025-09-02 15:03:19.860787
smp-core-sumo-connector-6fbc84d8c7-mcld5 [2025-11-04 16:35:08 | app.py:process:163] > INFO:  sim: 2025-09-02 15:03:19.810787, live: 2025-09-02 15:03:19.860787, offset: 0.05
smp-core-sumo-connector-6fbc84d8c7-mcld5 [2025-11-04 16:35:08 | app.py:process:193] > WARNING:  Failed to find routes for 1756825381936811
smp-core-sumo-connector-6fbc84d8c7-mcld5 Warning: Vehicle '1756825364389877' performs emergency braking on lane '-1.23_3' with decel=9.00, wished=4.00, severity=1.00, time=199.75.
smp-core-sumo-connector-6fbc84d8c7-mcld5 [2025-11-04 16:35:08 | app.py:process:154] > INFO:  Process poses from 2025-09-02 15:03:19.910787
smp-core-sumo-connector-6fbc84d8c7-mcld5 [2025-11-04 16:35:08 | app.py:process:163] > INFO:  sim: 2025-09-02 15:03:19.860787, live: 2025-09-02 15:03:19.910787, offset: 0.05
smp-core-sumo-connector-6fbc84d8c7-mcld5 [2025-11-04 16:35:08 | app.py:process:193] > WARNING:  Failed to find routes for 1756825381936811
```

Figure 5. *Snapshot of the dashboard for live traffic visualization (background map: GeoBasis-DE/LGLN (2024) CC-BY/4.0).*

Within the visualization, vehicles rendered in green are controlled by SUMO, whereas vehicles displayed in other colors correspond to those detected by the infrastructure. The association between a simulated vehicle and its real-world counterpart is illustrated by a red connecting line. This graphical linkage makes the assignment and tracking relationships immediately interpretable and supports qualitative validation of the coupling mechanism.

The lower-right panel of Figure 5 displays a live excerpt of the logs produced by the running SMP service. The service itself operates as part of a distributed ensemble of microservices. Another service of this ensemble ensures that the information generated by the SUMO connector is available within the dashboard by forwarding messages from Kafka via a WebSocket server to connected web clients. The WebSocket-based communication ensures low-latency data delivery while maintaining a decoupled architecture between backend processing and frontend visualization. Each service is deployed via a Helm chart within a Kubernetes environment, ensuring reproducible deployment, scalability, and operational resilience orchestrated by an ArgoCD instance.

4. Online integration challenges

In the current version, historical data and the calibrated simulation result in [16] were used to demonstrate the technical feasibility of coupling SCENIMINI and SUMO. The RI operates under adaptive traffic control, for which the control logic is not available. Accordingly, a 24-hour signal plan extracted from recorded traffic light data on the same day was applied in the simulation. The primary objective at this stage was to validate that traffic data collected with infrastructure can, in principle, be transferred and represented within a microscopic simulation environment. The results confirm that the implemented interface enables the integration of real-world traffic data into SUMO, thereby establishing the foundation for a data-driven simulation workflow. Several issues have emerged at this stage, implying challenges that need to be addressed for future online integration.

At first, a number of simulated vehicles could not be matched to corresponding real-world vehicles. In fact, while the raw dataset used for testing contains 393 vehicles only 333 could be added into the SUMO simulation, as the raw data included invalid and fragmented trajectories as well as misclassifications. Furthermore, of these 333 vehicles, only 159 of real vehicles and their simulated counterparts were actually crossing the RI at the same time in the region, illustrated in the left panel of Figure 6. For the remaining real-simulation vehicle pairs, one vehicle in each pair hat either not yet entered or had already left the RI.

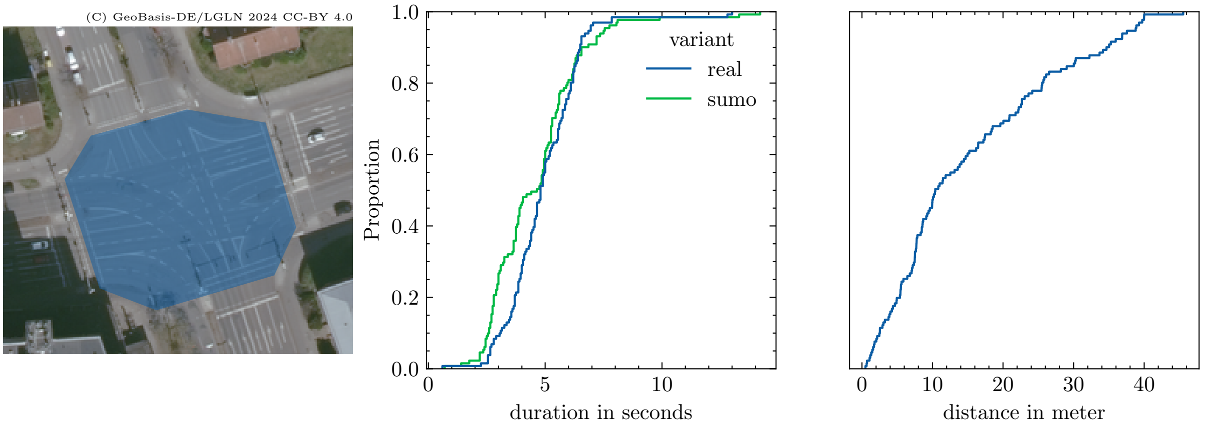

Figure 6. *Comparison between real and simulated traffic participants. Left: the investigation area. Middle: Duration of traffic participants. Right: Average distance between real and simulated trajectories.*

Moreover, in some cases, substantial deviations in vehicle positions were observed. This effect is also illustrated in the right panel of Figure 6 showing the empirical cumulative distribution of the average spatial distance between the real trajectories and their simulated counterparts within the RI area. These discrepancies indicate limitations in the current synchronization between simulation and real-world traffic data. An additional rationale is illustrated in the middle panel of Figure 6, which shows the cumulative distribution of the time required for vehicles to traverse the RI. The data reveal that real vehicles exhibit a lower velocity compared to their simulated counterparts, likely attributable to the more idealized simulation environment without pedestrians and cyclists and the less reduced traffic density in the simulation. The described deviations are also evident in Figure 7 which depicts the spatial distribution of the start and end points of both real-world and simulated vehicles within the RI area. In addition to the raw point distributions, the kernel density estimation (KDE) plots are provided to illustrate the underlying spatial probability densities. The KDE representations clearly highlight the differences between real and simulated trajectories. For the starting positions, both datasets exhibit clusters located along the four main incoming directions of the RI. However, notable deviations can be observed for the end positions. While the real-world data exhibit three dominant clusters corresponding to the primary outgoing directions, the simulated results show an additional cluster in the northern direction, indicating that simulated vehicles also leave the RI toward the north.

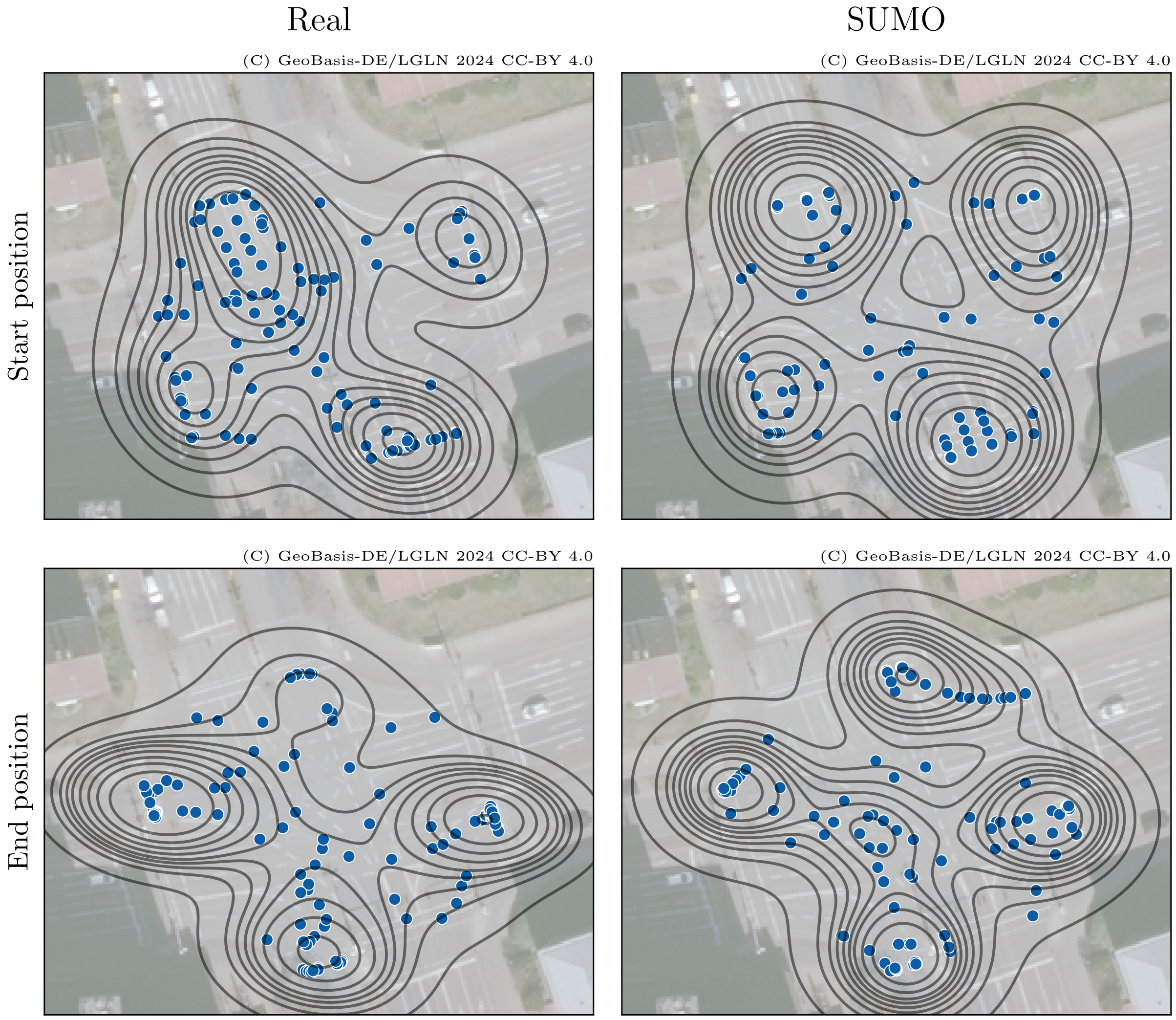

Figure 7. *An overview of the start and end positions for real vehicles and their simulated counterparts within the RI area.*

Furthermore, it can be observed that a non-negligible proportion of real-world trajectories terminate within the RI area itself. Such terminations may result from temporary occlusions or tracking losses by the infrastructure-based detection system. These cases lead to fragmented trajectories and therefore pose further challenges for online integration. These discrepancies may also be indicative of the observed differences in the mean spatial distance between real and simulated traffic participants. Other possible explanations for the observed deviations include synchronization issues between the simulated and real-world signal control as well as driver decision-making behavior during the yellow phase (i.e., stop-or-go decisions). These factors can lead to temporal mismatches, influencing vehicle trajectories and queue dynamics within the simulated environment.

Besides, log data revealed recurring difficulties when inserting vehicles into the simulation. In several cases, vehicles were detected by the infrastructure too late, meaning that they had already entered the RI area at the time of detection. Under these conditions, a clear assignment to specific routes was no longer feasible robustly, and some vehicles were delayed or even prevented from entering the simulation due to SUMO's vehicle inserting constraints. Consequently, these vehicles could not be integrated into the simulation as originally scheduled. Such resulting delays or exclusions may influence overall traffic dynamics to some extent.

Additionally, it remains to be examined to what extent real-time simulation is achievable with the current implementation. A potential bottleneck may arise from the repeated invocation of TraCI commands for data exchange and vehicle state updates. This polling-based communication can introduce latency and computational overhead, particularly under higher traffic volumes.

5. Conclusion and Future Work

This paper presents a modular and scalable approach for coupling infrastructure-based traffic observations with microscopic simulation in SUMO via SCENIMINI. To achieve this, a dedicated service was introduced to realize the technical coupling between real-world traffic data and SUMO. While the results confirm the integration of real-world traffic data into SUMO, thereby establishing the foundation for a data-driven simulation workflow, several issues have emerged at this stage, providing a solid foundation for further research and system refinement.

In particular, the findings highlight that working with real-world data still involves several challenges, some of which discovered in the context of this simulation coupling work, including fragmented trajectories, misclassifications and late object detection. In [17], several solutions were proposed that at least circumvent or even solve part of these problems, which can be considered in the next step. Further efforts should be devoted to data cleaning and refinement in online integration, as well as incorporating pedestrian and cyclist trajectories, which can influence right-turning vehicle behavior. To minimize the influence of data quality and focus purely on simulation digitalization, a second mode should be considered for development and implementation, in which the simulation lags behind real time, advancing only after all vehicles in a scene have left the RI and all their trajectories are either collected or repaired for simulation. Such a delayed execution facilitate to reduce route uncertainty and fragmented trajectories, which in turn makes the remaining issues more clearly identifiable.

Moreover, future work is required to address runtime traffic light control via TraCI for online applications. Corresponding models are already available within TASI as `TrafficLight` and `TrafficLightState`, enabling the integration of adaptive or externally synchronized signal control strategies. By aligning simulated signal phases more closely with real-world operations, it is expected that the temporal consistency between simulated and observed vehicle movements can be significantly improved. In addition, further efforts will be devoted to enabling the mapping of vehicles that are first detected within the RI onto the corresponding internal links in SUMO, and on utilizing heading information from the initial observations to infer the respective route. Moreover, the relevant vehicle insertion rules will be systematically examined to ensure that the observed traffic states are accurately reproduced in the simulation.

While the visual comparison already highlights discrepancies between simulation and reality, it is limited to individual situations. Nevertheless, the implemented coupling now enables the collection of long-term comparative data. This provides the basis for a systematic evaluation of deviations between simulated and real traffic processes and supports future calibration and validation efforts.

Beyond spatial consistency, future studies may investigate the runtime behavior of the coupled system for real-time processing. A central aspect is the latency introduced by the simulation cycle between real-world observations. This end-to-end latency results from data transmission via Apache Kafka, processing within the SMP service, communication through TraCI, and execution of the simulation step. Excessive delays may result in impaired temporal synchronization and comprised the validity of comparative analyses. A further key consideration is the selection of an appropriate sampling rate and corresponding simulation step size. It is evident that a reduction in the sampling rate reduce system load; however, this is accompanied by a reduction in temporal resolution. Identifying an optimal configuration therefore requires balancing synchronization accuracy with computational efficiency.

Finally, the libsumo Python library is already used to reduce overhead. However, repeated execution of SUMO queries through TraCI may still represent a potential bottleneck for real-time processing. This aspect must be systematically investigated in subsequent studies. One promising direction is the use of TraCI subscriptions to retrieve relevant state information of traffic participants. By subscribing to selected variables rather than issuing frequent polling requests, communication overhead could be reduced, and data exchange made more efficient. Such an approach may further enhance the feasibility of live simulation and improve overall system performance under higher traffic loads. In addition, investigating appropriate latency, i.e. the optimal simulation step size, would further support live simulation, instead of using data from every available frequency step.

Data availability statement

The data used in this paper is for internal purpose. But a data set collected at DLR's Research Intersection in Braunschweig that includes information of traffic participants, the infrastructure via an OpenDRIVE map and weather is available in [1].

Author contributions

- Conceptualization: Y.-P. Flötteröd, L. Klitzke, P. Wagner
- Data curation: L. Klitzke
- Formal Analysis: Y.-P. Flötteröd
- Investigation: L. Klitzke, Y.-P. Flötteröd
- Methodology: Y.-P. Flötteröd, L. Klitzke, P. Wagner
- Software: L. Klitzke
- Validation: L. Klitzke, Y.-P. Flötteröd
- Visualization: L. Klitzke
- Writing – original draft: L. Klitzke, Y.-P. Flötteröd, P. Wagner
- Writing – review and editing: L. Klitzke, Y.-P. Flötteröd, P. Wagner

Competing interests

The authors declare that they have no competing interests.

Funding

This work is funded by the mFUND project KoFeMo [18], and therefore by the German Ministry of Digitalization and Transport. In addition, it used some basic funding of the DLR.

References

[1] C. Schicktanz et al., "The DLR Urban Traffic Dataset (DLR-UT): A Comprehensive Traffic Dataset from an Urban Research Intersection," in *IEEE IV 2025 - 36th IEEE Intelligent Vehicles Symposium*, Piscataway, NJ: IEEE, 2025, pp. 398–405, ISBN: 979-8-3315-3803-3. DOI: 10.1109/IV64158.2025.11097400.

[2] L. Klitzke, K. Gimm, C. Koch, and F. Köster, "Extraction and analysis of highway on-ramp merging scenarios from naturalistic trajectory data," in *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*, 2022, pp. 654–660. DOI: 10.1109/ITSC55140.2022.9922191.

[3] C. Schicktanz and K. Gimm, "Detection and Analysis of Critical Interactions in Illegal U-Turns at an Urban Signalized Intersection," *Data Science for Transportation*, vol. 7, no. 1, 2025, ISSN: 2948-135X. DOI: 10.1007/s42421-025-00117-5.

[4] L. Klitzke, C. Leschik, R. Lüdtke, and K. Gimm, "Investigation on road traffic safety in rural areas using trajectory data: Case studies at two measurement sites," *Traffic Safety Research*, vol. 9, e000120, Dec. 2025.

[5] P. A. Lopez et al., "Microscopic traffic simulation using sumo," in *The 21st IEEE International Conference on Intelligent Transportation Systems*, IEEE, 2018. [Online]. Available: https://elib.dlr.de/124092/.

[6] P. Alvarez Lopez et al., *Simulation of urban mobility (sumo)*, version 1.26.0, Jan. 2026. DOI: 10.5281/zenodo.18406080. [Online]. Available: https://doi.org/10.5281/zenodo.18406080.

[7] Z. Liang et al., "Real-time microscopic traffic simulation and optimization at intersections with video traffic detection," *International Journal of Electrical and Electronic Engineering & Telecommunications*, vol. 7, no. 1, pp. 30–37, 2018, ISSN: 2319-2518. DOI: 10.18178/ijeetc.7.1.30-37.

[8] R. Oertel, J. Erdmann, R. Markowski, W. Schmidt, J. Trumpold, and P. Wagner, "Vital - verkehrsabhängig intelligente steuerung von lichtsignalanlagen," *Strassenverkehrstechnik*, pp. 631–638, 2018, ISSN: 0039-2219. [Online]. Available: https://elib.dlr.de/119877/.

[9] Y.-P. Flötteröd, M. Behrisch, M. Hendriks, J.-B. Bonne, E. Vullings, and R. Bruining, "Co-simulation of vehicles and crowds for rescue trials," in *SUMO User Conference 2019*, M. Weber, L. Bieker-Walz, R. Hilbrich, and M. Behrisch, Eds., ser. EPiC Series in Computing, vol. 62, EasyChair, 2019, pp. 56–67. DOI: 10.29007/9lnm. [Online]. Available: /publications/paper/Mwlj.

[10] Y.-P. Flötteröd and M. Behrisch, "Online calibration with sumo for network-wide traffic and emission monitoring – case study its huainan," *SUMO Conference Proceedings*, vol. 2, pp. 1–11, Jun. 2022. DOI: 10.52825/scp.v2i.88. [Online]. Available: https://www.tib-op.org/ojs/index.php/scp/article/view/88.

[11] L. Klitzke, "From Real-World Traffic Data to Scenarios in the Context of Automated Vehicles," dissertation, University of Oldenburg, 2026. DOI: 10.5281/zenodo.18636638.

[12] C. Schicktanz, L. Klitzke, A. R. Uschok, K. Gimm, K. Liesner, and H. H. Mosebach, "Offene Verkehrsdaten aus dem Testfeld Niedersachsen für den sicheren Verkehr der

Zukunft," *ATZextra: Sonderheft*, Fortschritte in Energie, Sicherheit und Fahrzeugtechnik, vol. 2026, no. 01, pp. 10–15, 2026, ISSN: 2509-4610. [Online]. Available: https://elib.dlr.de/224026/.

[13] L. Klitzke and C. Schicktanz, *TASI: A Python Library for Traffic Data Analysis and Situation Interpretation*, 2024. DOI: 10.5281/zenodo.14034644.

[14] S. Colvin et al., *Pydantic Validation*, version v2.13.0a0+dev, Oct. 2025. [Online]. Available: https://github.com/pydantic/pydantic.

[15] C. Schicktanz et al., "The DLR Highway Traffic Dataset (DLR-HT): Longest Road User Trajectories from a German Highway," in *IEEE International Automated Vehicle Validation Conference*, Piscataway, NJ, USA: IEEE, 2025, pp. 1–8, ISBN: 979-8-3315-2526-2. DOI: 10.1109/IAVVC61942.2025.11219599.

[16] Y.-P. Flötteröd and P. Wagner, "Leveraging intersection trajectory data for microscopic traffic simulation," in *Transport Systems Development – Methods and Solutions*, G. Sierpiński, S. Naumann, and E. Macioszek, Eds., Cham: Springer Nature Switzerland, 2026, pp. 12–24, ISBN: 978-3-032-14826-1.

[17] Y.-P. Flötteröd and P. Wagner, "Sumo simulation of DLR's research intersection," *SUMO Conference Proceedings*, vol. 6, pp. 25–32, Jul. 2025. DOI: 10.52825/scp.v6i.2633. [Online]. Available: https://www.tib-op.org/ojs/index.php/scp/article/view/2633.

[18] KoFeMo. "Kombinierte Untersuchung von Feinstaub und Mobilität – KoFeMo," Accessed: Feb. 24, 2026. [Online]. Available: https://bmdv.bund.de/SharedDocs/DE/Artikel/DG/mfund-projekte/kofemo.html.