SUMO User Conference 2026

TIBconferencesession

TIB-OP will set DOI with \TIBdoi

Submitted: 1970-01-01

A Dynamic Traffic Calibration Framework for SUMO Based on Historical Real-World Speed Data

Finn Guist[1], Andreas Freymann[2], Sandro Lipinski [2], Emanuel Reichsöllner[1], and Mirko Sonntag[1]

[1]University of Applied Sciences, Esslingen, Germany

[2]Fraunhofer Institute for Industrial Engineering IAO, Stuttgart, Germany

Abstract: Microscopic traffic simulation calibration is often constrained by limited data availability and the inherent complexity of calibration processes. In some cases, only aggregated speed information is accessible, while detailed data such as vehicle trajectories are unavailable. To address this challenge, this work proposes an automated Dynamic Traffic Calibration Framework (DTC-Framework) based on SUMO calibrators. The framework relies exclusively on varying average speed data over daytime and transforms it into a dynamically calibrated SUMO simulation. Consequently, it remains applicable even in the absence of Origin-Destination (OD) matrices, turn counts, or trajectory data. In addition, this work evaluates both speed-based and flow-based SUMO calibration approaches and concludes that the speed-based approach is the more effective method. The required real-world speed data is obtained via TomTom's Traffic Application Programming Interface (API), which provides Vector Flow Tiles representing average speed data for processing. The DTC-Framework is validated using real-world traffic data from a German city. The results demonstrate that the speed-based calibration approach significantly improves the realism of the simulation.

Keywords: SUMO, traffic simulation, calibrators, data-driven calibration, speed-based calibration, data-flow-based calibration, Floating Car Data (FCD)

1 Introduction

The increasing complexity of urban mobility systems drives the need for high-fidelity traffic simulations [1]. Microscopic traffic models enable detailed analysis of vehicle interactions and congestion dynamics [1]. In this context, SUMO (Simulation of Urban Mobility) [2] has become one of the most widely used simulation environments for this purpose, especially in research, due to its open-source license, transparent modeling assumptions, and continuous research-driven development [2]–[4]. In contrast, commercial simulation frameworks such as PTV Vissim [5], Aimsun Next [6], or Paramics [7] are often associated with substantial licensing costs.

Accurately representing real-world traffic dynamics is essential for such applications. Purely synthetic demand models or static simulation scenarios are unable to adequately capture short-term disturbances such as congestion waves, accidents, construction sites, or temporary bottlenecks [1]. At the same time, available real-world traffic data are typically limited [8]. However, this does not imply that traffic calibration becomes easier when more detailed data are available: previous studies show that traffic calibration remains a significant challenge even in the presence of extensive datasets, such as trajectory data [9]. Commercial Floating-Car-Data (FCD) providers such as TomTom [10] typically offer aggregated information such as average speeds or probe counts at the segment level, but do not supply complete trajectories or Origin-Destination (OD) matrices [11]. This mismatch between aggregated data and microscopic simulation requirements creates a fundamental calibration challenge where different existing approaches attempt to bridge this gap. Parameter calibration using optimization methods has been demonstrated with genetic algorithms [12] while multi-source offline and online calibration workflows show how historical and real-time data can be fused to adjust traffic states dynamically [13], [14]. Other research proposes automated segment-matching and the generation of route distributions based on commercial probe counts [15]. Although effective in specific case studies, these techniques often require high-resolution data, substantial manual tuning, or computationally intensive optimization, limiting their scalability and transferability.

To address the aforementioned challenges, this paper proposes a Dynamic Traffic Calibration Framework (DTC-Framework), which provides a structured, stepwise calibration methodology and is implemented as a software-based solution. The framework dynamically calibrates SUMO simulations using aggregated real-world speed data that varies over the course of the day and reflects current, and therefore more realistic, traffic conditions. The implementation is publicly available via a GitHub repository [16]. It generates a randomized base traffic demand using standard SUMO tools and introduces a dynamic calibration layer based on SUMO calibrators [17], which iteratively adjust traffic conditions. For real-world data integration, the DTC-Framework describes the acquisition and processing of traffic speed data from a TomTom Traffic API [18]. Since SUMO calibrators support both speed-based and flow-based calibration approaches [17], this work investigates and compares both methods. While both approaches are evaluated, the remainder of the paper focuses on the speed-based approach, as it proves to be more robust and better suited for calibration using aggregated speed data, as demonstrated by a comparative analysis. Furthermore, a dedicated SUMO simulation scenario was defined to apply and evaluate the DTC-Framework. The framework targets scenarios with limited data and enables dynamic, data-driven calibration without relying on OD matrices, turn counts, or trajectory data. It relies exclusively on aggregated real-world speed data which change over daytime and does not model flows, densities or demands. Its objective is to reproduce observed speeds rather than to capture demand–supply feedback such as congestion-induced re-routing, thereby preserving consistency with the input data concerning the number of vehicles and the assigned routes.

The remainder of this paper is structured as follows: Section 2 reviews related work. Section 3 introduces the DTC-Framework, describing its steps from traffic data acquisition to metric generation. Section 4 presents the SUMO simulation scenario used to apply and evaluate the framework. Section 5 compares and validates the two main SUMO calibration approaches, speed-based and flow-based, to identify the most effective strategy. Section 6 describes the execution of the defined SUMO scenario by

using the speed-based approach. Finally, Section 7 presents the results and validation, and Section 8 concludes the paper and provides an outlook on future work.

2 Related Work

Research on data-driven microscopic traffic simulation covers frameworks, reference scenarios, calibration, and FCD-based modeling. This section positions the present work within this context.

Several traffic simulation tools are used in research and industry which differ in complexity, licensing, and methodological focus. SUMO's transparent model structure makes it well-suited for reproducible scientific workflows. Furthermore, SUMO's calibration functionality (using e.g., `calibrator`, `routeProbe`) provides great flexibility for experiments based on aggregated measurements [17], [19]. Commercial tools, e.g., Aimsun [6], PTV Vissim [5] or Paramics [7] offer comprehensive modeling environments but come with licensing constraints and closed-source model assumptions.

A number of large-scale SUMO scenarios have been developed to benchmark microscopic simulations and ensure reproducibility across studies. These include, e.g., the Luxembourg SUMO Traffic scenario [20], the Bologna scenario [21], or recent high-resolution models such as Berlin SUMO Traffic [22] or Monaco SUMO Traffic [20]. They provide well-calibrated environments to evaluate routing algorithms or simulation techniques. Table 1 shows publicly available SUMO scenarios and the input data they rely on. Across all scenarios, a consistent pattern emerges: each model relies on extensive empirical datasets, including full OD matrices, induction-loop measurements, or high-resolution FCD trajectories. This creates a high barrier to build comparable scenarios in cities with limited sensing infrastructure. This work explicitly addresses this gap by enabling a realistic microscopic traffic reconstruction using only aggregated speed data, a fundamentally lower-information setting than those used in established benchmarks.

Calibration is crucial to ensure consistency between simulated and real-world traffic states. Several studies propose optimized calibration approaches. [12] utilize vehicle speed data and genetic algorithms to tune model parameters. [13] integrate historical and real-time traffic data to incrementally calibrate SUMO simulations in the Intelligent Transportation Systems (ITS) Upper Austria project. [14] demonstrate online calibration for network-wide monitoring in the ITS Huainan test environment. These approaches rely on detailed speed and flow measurements, which are often unavailable in commercial FCD datasets. By contrast, this work examines if meaningful calibration is possible with only aggregated average-speed data using SUMO calibrators.

FCD has been used for traffic-flow analysis, congestion detection, and the reconstruction of microscopic states [11]. A further work outlines core use cases and limitations related to penetration rates and sampling variability [11]. Furthermore, [27] analyze TomTom trajectories and demonstrate the presence of synchronized flow. [15] provide an advanced pipeline for converting probe counts into high-resolution detector data for SUMO scenarios. Their approach, however, requires frequent probe updates and cannot be applied directly to datasets consisting only of mean speeds. Thus, this paper extends the applicability of SUMO calibration techniques to these more limited data environments. Moreover, the use of pure traffic count data represents a classical approach to addressing limited data availability. However, as shown in [28], such data are often affected by measurement errors or incompleteness. Against this background, the

Table 1. *SUMO Simulation Conference*

Scenario	Size [km^2]	Input data for demand
Bologna Scenario [21]	–	Induction loops, public mobility statistics, detector data from several junctions
Luxembourg SUMO Traffic [20]	156	Public statistics for `activitygen`, induction loops, floating-car validations
Berlin SUMO Traffic [22]	800	Demand extracted from MATSim scenario, iterative traffic assignment, population model
Dublin Scenario [23]	17.5	Induction loops (approx. 480, every 6 minutes), Bluetooth sensor flows
Ingolstadt Traffic Scenario [24]	51.54	Induction loops (24 junctions, every lane, 15-minute intervals), demographic statistics
Ingolstadt City Model (extended) [24]	100	Traffic-light loop detectors and socio-demographic data
Monaco SUMO Traffic [20]	22	Public statistics for `activitygen`, elevation model, induction loops
Tokyo SUMO Traffic Scenario [25]	33.22	Induction loops (7 junctions, 60-minute intervals), population statistics
Mannheim SUMO Traffic Model [26]	144	OD matrices, loop detectors, floating-car measurements; Covariance Matrix Adaptation Evolution Strategy (CMA-ES) calibration under data-scarce conditions

proposed framework deliberately focuses on modern aggregated speed data sources, which are now widely available and enable a more realistic representation of traffic conditions.

Compared to existing tools, scenarios, and calibration approaches, this work contributes a framework that does not rely on OD matrices or comprehensive detector networks. It uses aggregated speed data changing over daytime and representing a more realistic traffic situation. By combining baseline traffic generation with SUMO's calibration mechanisms, the approach examines a reproducible method for constructing realistic microscopic simulations from minimal data.

3 Dynamic Traffic Calibration Framework

This section introduces the DTC-Framework, an automated solution that transforms aggregated real-world speed data into a dynamically calibrated SUMO simulation. The framework constitutes a structured, stepwise calibration methodology, implemented as an automated pipeline [16]. The pipeline is organized into sequential steps including *Traffic Data Acquisition (1)*, *SUMO Network Generation (2)*, *Speed Mapping (3)*, *Base Traffic Generation (4)*, *Calibrator Generation (5)*, and *Metrics Generation (6)*, building on a defined SUMO network (Figure 1). All processing steps illustrated in Figure 1, including the associated scripts, were implemented by this work as part of the DTC-Framework. Several of these scripts internally make use of existing SUMO tools (e.g., `randomTrips`, `duarouter`, or `routeSampler`), which are explicitly referenced and described in the corresponding subsections.

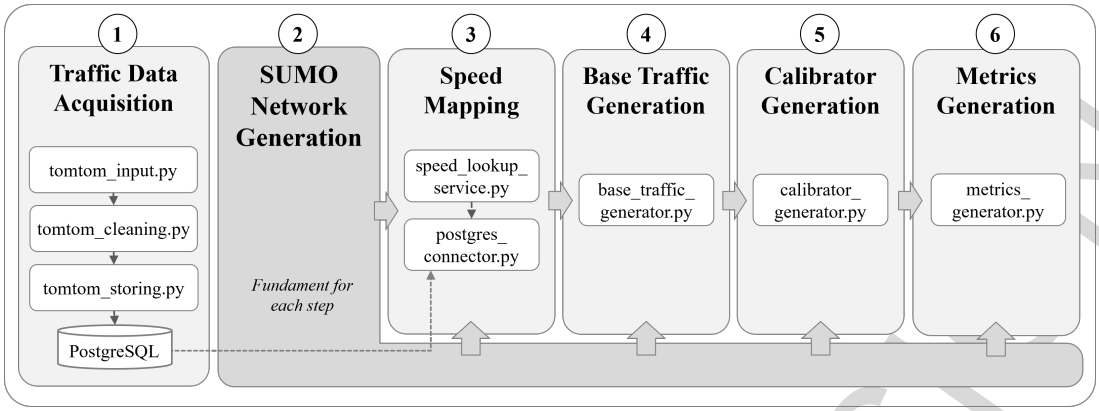

Figure 1. *Overview of the DTC Framework, its six-step process, and key implementation details*

3.1 Traffic Data Acquisition

This first step of the framework focuses on acquiring traffic speed data depending on the area covered by the simulation. The speed data are retrieved from a TomTom API [18], which provides Vector Flow Tiles containing speed information varying over daytime. Vector tiles are a vector-based representation of spatial data, in which the road network is partitioned into square sections; each tile contains a predefined set of road geometries enriched with associated traffic flow information [18]. Retrieving traffic data are decoded from TomTom's *Protocol Buffer Binary Format* and reconstructed as `LineString` representations in the *World Geodetic System 1984 (WGS84)* coordinate system. Each `LineString` represents a road segment and carries attributes, e.g., road type or traffic level. Relative tile coordinates are transformed into absolute geographic positions to ensure spatial consistency and interoperability. Each extracted `LineString` defined by its corresponding tile and timestamp is stored as an individual record in a PostGIS-enabled PostgreSQL database.

Furthermore, the traffic speed data is organized in geographically bounded tiles defined by a global *XYZ tiling scheme*, where X and Y denote the tile's position on a two-dimensional grid and Z represents the zoom level. The zoom levels range from $Z = 0$ to $Z = 22$, where at $Z = 0$ the entire Earth is represented by a single tile, and each subsequent level subdivides the grid into $2^Z \times 2^Z$ tiles [29]. The physical size of a tile can be derived from the Earth's circumference using the relation $\frac{40{,}075{,}017}{2^Z}$, yielding the tile edge length in meters [29]. At the highest zoom level ($Z = 22$), a tile corresponds to an area of approximately 9.5×9.5 meters. While the TomTom documentation does not explicitly define zoom levels in terms of spatial resolution or road-type filtering, the hierarchical structure of the tiling scheme implies a decreasing spatial extent with increasing Z. This relationship can also be empirically observed when requesting tiles at different zoom levels. For this work, a zoom level of $Z = 13$ was chosen as a compromise between spatial resolution and data volume.

The DTC-Framework supports both processing data dumps containing proper traffic data from TomTom and collecting data using dedicated scripts based on the *lambda architecture* for processing large amounts of data [30]. It includes `tomtom_input.py` for retrieving the data, `tomtom_cleaning.py` to transform `LineString` data objects, and `tomtom_storing.py`, which stores the data in a PostgreSQL database. These scripts share the data via the message broker RabbitMQ.

3.2 SUMO Network Generation

This step involves generating the SUMO simulation network. This step is primarily conceptual, highlighting the necessity of a SUMO network, which then serves as the basis for the later steps. The framework does not provide an implementation for this step. However, SUMO offers the OSMWebWizard (based on OpenStreetMap, OSM) [31], a well-suited tool for constructing simulation scenarios that can be used to create a SUMO scenario using OpenStreetMap (OSM) data defined by a bounding box.

3.3 Speed Mapping

The next step retrieves `LineString` geometries which are mapped to the SUMO network. This speed mapping is performed at the lane level, since lanes are the primary driving units where vehicles move and interact, whereas edges may contain multiple lanes, causing ambiguous speed assignments [3], [4]. The mapping uses the geometric centerline; if no speed data lies exactly on it, the nearest available data is assigned. This ensures a precise link between real-world speeds and the simulated network. Figure 2 shows the mapping process for a single lane by starting with the first step of requesting an average speed for a lane (1). For sets of SUMO lanes, each TomTom street segment is reconstructed using PostGIS `ST_Buffer()` in the next step (2) to create a polygon representing the road surface. The buffer radius corresponds to half of the SUMO road, and a buffer style parameter controls side-specific buffering behavior.

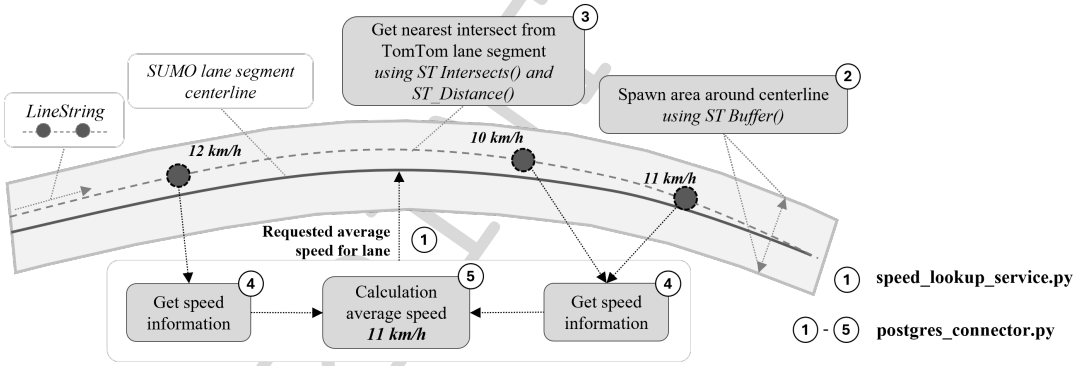

Figure 2. *Overview of the speed mapping functionality illustrated for a single lane.*

In the next step (3), the buffered polygon is then spatially compared with the TomTom `LineString` geometries using `ST_Intersects()` to get all intersecting segments. To ensure that only the most relevant speed information is assigned, a nearest-neighbor selection is performed. A subquery identifies the TomTom segments with the minimum spatial distance to the SUMO segment using `ST_Distance()`. Only segments matching this minimum distance are retained (4), ensuring that speed aggregation considers the closest corresponding road segments. Finally, the average speed is available (5). The implementation can be found in `speed_lookup_service.py` providing the service for speed requests, and `postgres_connector.py` retrieving speed data (see Figure 1).

A short excerpt of retrieved speed data for requested SUMO lanes at a given timestamp is shown below. Each record contains a timestamp and tuples `(LANE_ID, average speed)`, allowing efficient assignment of time-dependent speeds to lanes.

```
{'2026-01-27 06:05:00': [
```

```
    ('1035471353_1', Decimal('35.0000000000000000')),
    ('-105914659#1_1', Decimal('15.0000000000000000')),
...]}
```

3.4 Base Traffic Generation

This step ensures that the simulated traffic covers the entire network while preserving realistic vehicle interactions and route diversity. The initialization of base traffic in SUMO is done by the implemented `base_traffic_generator.py` script which uses several SUMO tools to generate trips and routes across the network (see Figure 3).

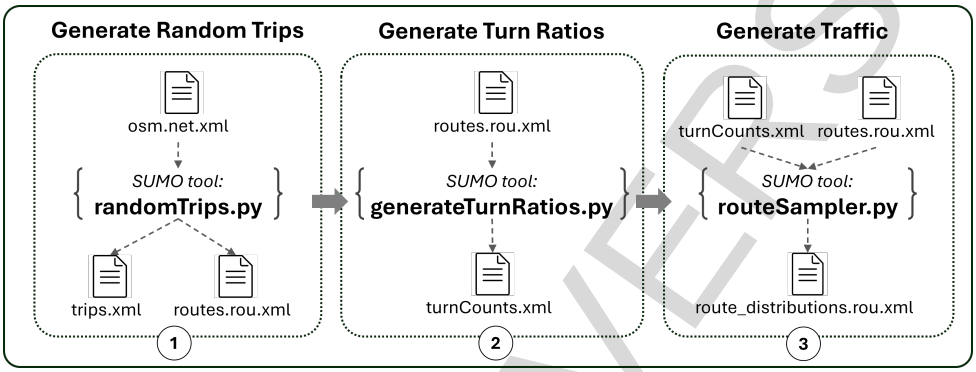

Figure 3. *Overview of the steps to generate trips and routes across the network*

In the first step, baseline trips are generated using the SUMO tool `randomTrips.py`, which takes the `osm.net.xml` file as input. This tool stochastically assigns trip origins and destinations based on the network topology [32]. Trip generation can either apply uniform sampling across the network or incorporate user-defined weighting schemes to reflect network structure, such as road hierarchy or spatial density. Temporal constraints allow the approximate representation of typical daily mobility patterns. The output consists of `trip.xml`, containing trip definitions, and `routes.rou.xml`, containing route definitions. Routes are then finalized using *duarouter* from SUMO, ensuring feasibility with respect to network connectivity and traffic rules [33]. Step two, turning ratios between consecutive edges are derived using the SUMO tool `generateTurnRatios.py` [34]. It takes `routes.rou.xml` as input and outputs `turnCounts.xml`, representing traffic volume distribution at network intersections. Finally, in the third step, which has the `turnCount.xml` and the `routes.rou.xml` as input, the SUMO tool `routeSampler.py` generates routed vehicles or flows based on a combination of turn ratios, edge counts or origin–destination relations [35]. The output is stored in `route_distributions.rou.xml`.

The output of this step is a plausible base traffic definition that populates the network but does not yet enforce conformity with empirically observed traffic speeds. In the current framework implementation, only the `routes.rou.xml` file is used as base traffic for the simulation. The additional artifacts are generated and provided by the framework but are currently not actively utilized in the simulation workflow.

3.5 Calibrator Generation

SUMO offers calibrator objects, which dynamically adjust traffic flows, speeds, and vehicle types based on measurements or location-specific behavior, adding or removing

vehicles as needed and preventing unrealistic congestion [17]. Applied on top of the previously generated base traffic, this step creates SUMO calibrators using the implemented `calibrator_generator.py` script (see Figure 1). The procedure is fully deterministic and automated. The script reads previously defined SUMO lanes that are relevant for calibration, i.e., those that serve primary traffic functions, such as highways or main roads. If observed speed data exists for a relevant lane, retrieved and processed by the speed mapping step (Section 3.3), a SUMO calibrator is generated, positioned at the middle of the lane, and written to an XML file. Only lanes with available speed information are calibrated. No additional filtering or clustering is applied at this stage. This design choice keeps the calibration layer simple and modular: any prior aggregation or clustering of lanes can be performed as a preprocessing step. The framework provides the option to thin the amount of lanes, e.g., to prevent potential performance issues, as studies have shown that memory consumption and runtime increase with the number of simulated agents and network size in SUMO [36].

To determine when and for how long a SUMO calibrator should calibrate a lane based on speed data, calibrators rely on SUMO flows, which generally specify traffic conditions over defined time intervals using parameters such as `vehsPerHour` or speed [17]. Depending on the calibration approach (i.e., speed-based or flow-based (Section 5)), the attributes of flows and the characteristics of the calibrator vary [17]. The DTC-Framework supports both approaches. However, in the remainder of this paper 5.5, only the speed-based calibration approach is considered. An excerpt of generated calibrators via an XML file using flows for the speed-based approach is shown in the following. Each calibrator is identified by a unique ID and its corresponding lanes. Within the speed-based approach, the flows specify the time periods (in seconds) during which the defined speed is applied to the calibrator. Using the speed-based approach, where only the speed attribute is specified for the calibrator flows, each calibrator acts as a variable speed sign [17]. Conceptually, this imposes a time-dependent speed limit on the connected edge, adjusting the permissible vehicle speed to a desired target value [37]. As the calibrators are placed on individual lanes within the DTC-Framework, this variable speed limit is applied at the lane level, preventing vehicles from exceeding the specified max. speed.

```
<?xml version="1.0" ?>
<additional><calibrator id="calib_-1035471353_0" lane="-1035471353_0" ...>
    <flow begin="0" end="54000" speed="13.88888888888889"/>
</calibrator>
<calibrator id="calib_-105914659#1_1" lane="-105914659#1_1" ...>
    <flow begin="300" end="600" speed="4.166666666666667"/>
    <flow begin="600" end="900" speed="4.166666666666667"/> ...
</calibrator></additional>
```

3.6 Metrics Generation

This step focuses on generating evaluation metrics from the executed simulation scenario. During runtime, the framework collects time-weighted, lane-specific speeds using the implemented `metrics_generator.py` script. These simulated values are then compared with corresponding empirical data (e.g., TomTom speed measurements) to assess the accuracy of the calibration. In addition, the script computes aggregated

global metrics over the entire simulation period. Thus, this evaluation step enables systematic testing of different simulation parameters, e.g., reducing the number of lanes or modifying traffic patterns in order to analyze their impact, e.g., on traffic flow dynamics.

4 Simulation Scenario for DTC-Framework

For validating the DTC-Framework, we defined a suitable SUMO simulation scenario based on the following requirements. The traffic situation must include both congested and free-flow periods, and the area shall be defined as an urban region of moderate size, avoiding both small cities and large metropolitan areas. The area of Esslingen, a city in the southwest of Germany, southeast of Stuttgart, meets these requirements (Figure 4, left side).

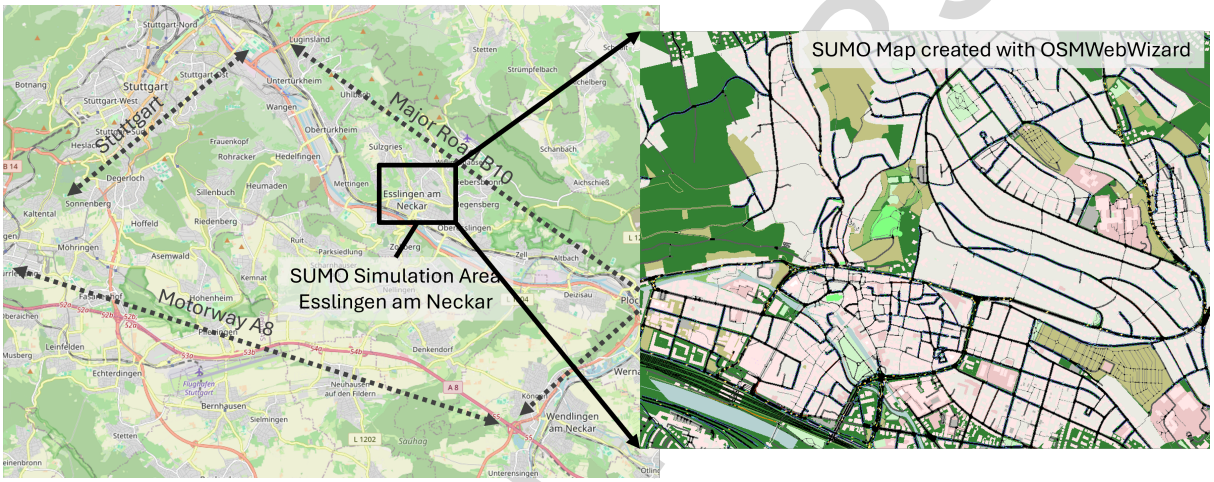

Figure 4. *Overview of the local conditions in Esslingen and the extracted SUMO Map*

With a population of around 95,000 [38], Esslingen represents a medium-sized urban area. It is located about 10 km from Stuttgart and is integrated into the regional transportation network, including major roads such as the B10 and connections to the A8 highway and regional rail and local bus services [39]. In addition, a mobility strategy aimed at promoting active travel (pedestrians, bicycle traffic) is currently being developed in Esslingen, with public participation to identify barriers and opportunities for network improvement [40]. This indicates notable traffic variability within the area.

For creating a SUMO simulation network, the OSMWebWizard [31] is used as described in step 2 of the DTC-Framework (Section 3.2). Figure 4 shows the SUMO network (right side) that has been sliced for the simulation scenario by using OSMWebWizard resulting in several XML files, including *osm.net.xml* and numerous intermediate files. The SUMO network of this scenario consists of 15.379 lanes and 14.383 edges. This work does not introduce a dedicated verification or correction step for the generated SUMO network. Basic plausibility checks, such as verifying network connectivity or identifying unrealistic geometries, are considered a necessary preprocessing step and should be performed after generating the network with the OSMWebWizard.

5 Experimental Comparison of SUMO Calibration Approaches

There are two complementary SUMO calibration approaches: (1) the *speed-based* calibration, in which target speeds are directly imposed on vehicles entering selected road

segments and (2) the *flow-based* calibration, in which traffic is adjusted such that the resulting traffic density reproduces observed speed patterns [17]. In general, when using SUMO calibrators, SUMO *routeProbes* are additional elements for traffic measurements [19]. These detectors track route distributions taken by vehicles on specific SUMO elements such as lanes over given time intervals, analogous to collecting route information through surveys or navigation provider data [19]. For choosing a suitable strategy for running and calibrating the defined SUMO simulation scenario of Esslingen (Section 4), this section experimentally compares these approaches.

5.1 Speed-Based Calibration

In the speed-based approach, when a calibrator is assigned to a SUMO lane l with a desired lane speed v_l^{obs}, all vehicles traversing this lane are limited to this max speed. Vehicles entering the calibrated SUMO lane are constrained to the target speed from the moment they spawn, analogous to a dynamic speed limit or a variable speed sign [17]. Since no additional vehicles are introduced, no *routeProbe* is required in this configuration. The resulting mean simulated speed v_l^{sim} emerges directly from the imposed constraint and vehicle interactions within the SUMO lane. This method provides a stable and computationally efficient way to enforce speed conformity. However, it does not explicitly model the relationship between traffic generation and speed and therefore bypasses fundamental traffic flow mechanisms.

5.2 Flow-Based Calibration (Exploratory)

Instead of constraining vehicle speeds explicitly, the flow-based calibration approach aims to reproduce observed average speeds as an emergent property of traffic density and vehicle interactions [17]. In classical traffic flow theory [1], the macroscopic variables traffic flow q, density k, and mean speed v are related by the equation:

$$q = k \cdot v. \tag{1}$$

This implies that changes in speed are inherently linked to changes in density and, consequently, to traffic generation. However, when only aggregated average speeds are available, as is the case in this work, the system is under-determined: for a given observed lane speed v_l^{obs}, infinitely many combinations of flow and density may satisfy the fundamental relation. As no direct measurements of density or flow are available, the traffic generation cannot be uniquely inferred from speed observations alone. Nevertheless, to investigate whether speed conformity can be achieved through demand adjustment, a simplifying assumption is introduced. Motivated by fundamental diagram models, which typically describe a monotonic decrease of speed with increasing density, a linear dependency between speed reduction and flow is postulated. For each calibrated SUMO lane l, a target flow q_l is defined as

$$q_l = q_0 + \alpha \cdot \left(v_l^{\text{free}} - v_l^{\text{obs}}\right), \tag{2}$$

where v_l^{free} denotes the free-flow speed derived from the network speed limit, q_0 represents the baseline traffic generation under free-flow conditions (in veh/h), and α is a global sensitivity parameter with unit $\frac{\text{veh/h}}{\text{km/h}}$ controlling how strongly deviations from free-flow speed translate into additional demand. Consequently, both q_0 and $\alpha \cdot (v_l^{\text{free}} - v_l^{\text{obs}})$ are expressed in vehicles per hour, consistent with SUMO's calibrator parameter `vehiclesPerHour`. Both parameters are kept constant across the network.

The linear relationship between speed deviation and induced flow is chosen for three reasons. First, classical fundamental diagram models [1] exhibit an approximately linear decrease of speed with increasing density in the free-flow regime. Thus, the proposed formulation represents a first-order approximation of the macroscopic speed-density relationship rather than a fully specified fundamental diagram. Second, only aggregated average speeds are available, while density and flow remain unobserved. Introducing nonlinear mappings would add parameters that cannot be empirically identified. The linear form thus provides the simplest identifiable model consistent with traffic theory. Third, a single global sensitivity parameter α is used due to the absence of lane-specific calibration data. Allowing lane-dependent parameters would increase flexibility but risk overfitting. A constant α ensures parsimony and interpretability.

In this configuration, calibrators inject vehicles according to the derived flow rates, while vehicle speeds remain unconstrained and emerge solely from microscopic vehicle interactions [17]. To preserve realistic routing behavior when additional vehicles are introduced, each calibrator needs to be associated with a corresponding `routeProbe` [19]. It records the routes taken by vehicles traversing this location over a defined aggregation period. Based on these observations, an empirical route distribution is derived that captures the prevailing turning behavior and route choices induced by the baseline demand. Newly injected vehicles are subsequently assigned to routes according to this distribution, ensuring that the additional demand introduced by the calibrator remains consistent with the existing traffic flow structure.

It is emphasized that this formulation does not claim a unique or physically exact reconstruction of traffic generation. Rather, it constitutes an exploratory approach that explicitly acknowledges the under-determined nature of the problem and tests whether a simple flow–speed coupling can serve as a plausible surrogate in the absence of detailed traffic counts. The linear relationship represents a first-order approximation in the free-flow regime, while the congested regime is not explicitly modeled due to the lack of density or flow data. To the authors' knowledge, systematic investigations of such demand inference strategies using calibrators and aggregated speed data remain limited, motivating the inclusion of this approach as a conceptual contribution.

5.3 Experimental Conditions

The following sections evaluate the effectiveness of the proposed calibration approaches by comparing simulated mean speeds with empirical averages from TomTom.

5.3.1 General Conditions and Configurations

All simulation experiments were conducted on the same SUMO network from the defined SUMO scenario described in Section 4. Calibrators were placed in the middle of lanes. However, not all lanes are used, only those serving a primary traffic function, e.g., main streets or highways, and for which an average speed value from TomTom is available. To ensure comparability across the calibration approaches, the same base traffic configuration is used generated by SUMO's `base_traffic_generator.py` tool over the full simulation horizon with stochastic trip generation (Section 3.4). An initial warm-up phase of 400 seconds was excluded from the evaluation to eliminate transient initialization effects and ensure steady-state traffic conditions. For each calibrated lane, the simulated mean speed was computed as a time-weighted average over all simulation

steps in which at least one vehicle was present on the respective lane. Performance metrics were calculated on the set of calibrated lanes; uncalibrated lanes were excluded from the error analysis, as no direct target values were defined for these segments. All reported speed values are expressed in km/h.

5.3.2 Error Metrics

To assess the conformity between simulated and observed traffic, three complementary metrics are used: Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Mean Signed Error (Bias), computed exclusively over the calibrated SUMO lanes. For each calibrated SUMO lane l, the simulated mean speed v_l^{sim} is obtained as a time-weighted average over the evaluation period, while v_l^{obs} denotes the corresponding observed average speed derived from the TomTom data. The MAE, used as the primary evaluation metric, is defined as:

$$\text{MAE} = \frac{1}{N} \sum_{l=1}^{N} \left| v_l^{\text{sim}} - v_l^{\text{obs}} \right|, \tag{3}$$

where N is the number of calibrated SUMO lanes. The MAE provides a directly interpretable measure of the average deviation in physical units (km/h) and is therefore well-suited for comparing calibration strategies [41]. To complement the MAE, the RMSE is computed as:

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{l=1}^{N} \left(v_l^{\text{sim}} - v_l^{\text{obs}} \right)^2}. \tag{4}$$

Due to its quadratic weighting, the RMSE penalizes larger deviations more strongly and is particularly sensitive to pronounced local mismatches, such as congestion near junctions or spillback effects [41]. Finally, Bias is defined as:

$$\text{Bias} = \frac{1}{N} \sum_{l=1}^{N} \left(v_l^{\text{sim}} - v_l^{\text{obs}} \right). \tag{5}$$

It captures systematic over- or underestimation of speeds. A positive Bias indicates that simulated speeds exceed observed values on average, whereas a negative Bias reflects systematic underestimation [42]. Together, MAE, RMSE, and Bias provide a compact yet comprehensive characterization of calibration accuracy, robustness to outliers, and systematic tendencies in the experimental results.

5.4 Experimental Investigation

Three experimental cases are presented and compared: (1) a scenario without calibration, which relies solely on statically generated base traffic, (2) a speed-based calibration using calibrators with explicit target speeds, and (3) a flow-based calibration employing a linear flow-speed coupling with varying sensitivity parameters.

5.4.1 Experiment 1 - Without Calibration

As a reference, a baseline simulation using only statically generated base traffic was evaluated (Table 2, left column). The pronounced positive bias indicates a systematic overestimation of speeds in the absence of calibration. In combination with the high MAE, it confirms that base traffic generation alone is insufficient to reproduce realistic speed levels, as it primarily ensures structural connectivity and route diversity rather than quantitative traffic state accuracy.

5.4.2 Experiment 2 - Speed-Based Calibration

In this experiment, calibrators are configured with explicit target speeds corresponding to the observed average speeds v_l^{obs}. No additional vehicles are introduced, and traffic generation is solely determined by the pre-generated base traffic. With direct speed enforcement, the resulting deviations are listed in Table 2 (middle column). The small MAE indicates close conformity between simulated and observed speeds. The Bias reveals a slight systematic underestimation of approximately 1.97 km/h. This offset can be attributed to aggregation effects, junction interactions, and the fact that target speeds are assigned at vehicle insertion, while the subsequent motion along the edge is governed by microscopic car-following dynamics. To compensate for this systematic offset, an experimental second run was conducted with all calibrator target speeds increased by 2 km/h. The results are shown in Table 2 (right column). The Bias is reduced to 0.25 km/h, effectively eliminating the previously observed systematic underestimation. This demonstrates that, in this specific configuration, the global offset correction largely removes systematic underestimation and reduces the variance-sensitive error (RMSE). At the same time, the slight increase in MAE shows that absolute deviations do not uniformly improve. Consequently, the +2 km/h adjustment cannot be considered unequivocally superior in general, but rather represents a trade-off that depends on the chosen evaluation criterion. If statistical neutrality (Bias ≈ 0) is prioritized, the adjusted configuration is preferable. However, if minimizing average absolute deviation is the primary objective, the original speed-based calibration performs slightly better. In summary, the correction mainly redistributes the error structure rather than fundamentally improving predictive performance.

Overall, speed-based calibration provides an upper bound on achievable conformity with observed speed data, but the global offset correction does not fundamentally alter the underlying deviation structure.

Table 2. *Summary of evaluation metrics for different calibration scenarios*

Metric	Without calibration	Speed-based calibration	Speed-based calibration 2 km/h
MAE	18.99 km/h	1.97 km/h	2.24 km/h
RMSE	22.68 km/h	4.25 km/h	3.15 km/h
Bias	13.75 km/h	-1.97 km/h	0.25 km/h

5.4.3 Experiment 3 - Flow-Based Calibration with Linear Coupling (Exploratory)

Flow-based calibration adjusts traffic generation rather than explicitly constraining vehicle speeds [17]. For each calibrated lane l, the injected flow is determined by:

$$q_l = q_0 + \alpha \cdot \left(v_l^{\text{free}} - v_l^{\text{obs}}\right), \tag{6}$$

where q_0 represents a baseline flow and α controls the sensitivity of demand to deviations from free-flow speed. In all experiments, $q_0 = 10$ was fixed, while α was varied systematically. The calibrator period was set to 50 s for all simulations, meaning that the specified flow rates were applied in aggregation intervals of 50 seconds. Table 3 summarizes the error metrics for different sensitivity parameters. As α increases from 10 to 50, simulated speeds decrease due to rising traffic generation, resulting in a substantial reduction of MAE and Bias. For small values of α, the system remains close to free-flow conditions (MAE > 20 km/h) with a strong positive Bias, indicating systematic overestimation of speeds. Increasing the sensitivity to $\alpha = 50$ significantly reduces the MAE to 13.64 km/h and lowers the Bias to 4.18 km/h, representing the best configuration among the tested parameters.

Table 3. *Flow-based calibration results for different sensitivity parameters α.*

α	MAE [km/h]	RMSE [km/h]	Bias [km/h]
10	24.30	26.34	24.25
20	21.48	23.97	21.41
30	17.82	21.13	17.31
50	13.64	17.87	4.18
100	17.61	20.91	-4.25
150	21.01	23.75	-7.53

However, further increasing α does not lead to continued improvement. At $\alpha = 100$, the MAE rises again to 17.61 km/h and the Bias becomes negative (-4.25 km/h), indicating a systematic underestimation of speeds due to excessive demand. For $\alpha = 150$, the deviations deteriorate even further, suggesting that overly aggressive demand adjustments destabilize the traffic state and amplify errors. These results indicate the existence of an optimal sensitivity range. While moderate α values effectively reduce systematic deviations, larger values overcompensate and degrade performance, demonstrating the nonlinear response of the network to linear demand scaling.

Compared to the baseline scenario without calibration (MAE = 25.09 km/h), the flow-based approach still yields noticeable improvements, particularly around $\alpha = 50$. Nevertheless, even in its best configuration, it remains considerably less accurate than the speed-based calibration, which achieves MAE values below 2 km/h.

5.5 Comparison of Speed-Based and Flow-Based Approaches

The comparison of the speed-based and flow-based approach highlights a clear distinction between direct speed enforcement and flow-based calibration. While speed-based calibration achieves very low deviations (MAE $\approx$ 1.97 km/h), the flow-based approach yields substantially larger errors throughout the sensitivity parameters. This difference reflects a fundamental property of traffic systems: average speed alone does not uniquely determine traffic generation.

From a traffic flow theory perspective, speed, flow, and density are related through the fundamental diagram. However, knowing only the average speed v does not uniquely determine the corresponding flow q or density k. Multiple traffic states may exhibit similar speeds but differ in density and flow. Consequently, reconstructing demand solely from speed observations constitutes an under-determined problem. The linear coupling from equation 6 therefore represents a heuristic approximation rather than a theoretically exact inversion of the fundamental relationship. The experiments confirm this lim-

itation. Increasing α initially reduces the positive Bias and lowers the MAE compared to the baseline scenario, indicating that higher traffic volume partially reproduces the observed congestion effects. However, beyond $\alpha \approx 50$, further increases in sensitivity lead to a deterioration rather than continued improvement. The MAE rises again and the Bias turns negative, suggesting overcompensation of demand and highlighting that demand scaling alone cannot resolve structural mismatches between simulated and observed traffic states. In contrast, direct speed enforcement effectively constrains the system to match observed mean speeds. However, this approach modifies vehicle dynamics explicitly and therefore does not reconstruct a physically consistent traffic volume. It should thus be interpreted as an upper bound on achievable calibration accuracy rather than as a demand-recovery method.

Overall, the findings suggest that aggregated speed data alone are insufficient to uniquely determine traffic generation. While a simple linear flow–speed coupling improves realism compared to purely synthetic base traffic, the flow-based calibration approach cannot fully replicate empirical conditions. Thus, this work uses the speed-based calibration approach for the following SUMO simulation scenario.

6 Running the Speed-Based Simulation Scenario

To simulate the defined SUMO simulation scenario of Esslingen with the speed-based approach using historical real-world speed data from TomTom and SUMO calibrators, the proposed DTC-Framework is applied. The simulation period was defined as January 27, 2026, a Tuesday, selected as a representative day with typical traffic conditions. The SUMO simulation scenario is defined to start at 5:00 am UTC and end at 8:00 pm UTC, covering the time range that includes morning and evening rush hours. This represents a total time horizon of 54,000 seconds (15 hours) with a simulation step length of 1 second. The following simulation configuration was done:

1. **Traffic Acquisition** (Section 3.1): TomTom data were collected within the specified period at a temporal resolution of between 5-15 minutes, varying with rush-hour conditions, resulting in 67 timestamps. It should be noted that, due to a minor data collection issue, data points at full-hour timestamps are missing, occasionally resulting in sampling intervals exceeding 15 minutes; however, this has no impact on the results.
2. **SUMO Network Generation** (Section 3.2): The OSMWebWizard was used for generating the SUMO network.
3. **Speed Mapping** (Section 3.3): 295 relevant lanes were identified out of a total of 15,379 lanes and subsequently reduced by 50% to improve computational efficiency, resulting in 147 lanes used for calibration. Furthermore, lane types which are not related to passenger cars, such as `highway.steps`, `railway.subway` or `highway.bus_guideway`, were excluded, whereas major road types, such as `highway.primary` or `highway.secondary`, were retained. For each selected lane, speed data were retrieved for the 67 timestamps.
4. **Base Traffic Generation** (Section 3.4): Trips were restricted to passenger vehicles, a min trip length of 100 m was enforced to avoid unrealistic short movements, and a random routing factor of 2 was applied to increase route diversity. A fringe factor of 100 was used to emphasize boundary traffic, and a period of 1 with fixed random seed ensured reproducibility across the experiments.

5. **Calibrator Generation** (Section 3.5): With the information of the relevant lanes and the output of the base traffic, the calibrators were generated and positioned at the midpoint of each relevant lane defined within an XML file. Finally, the corresponding retrieved time-dependent average speeds from TomTom were assigned to the calibrators used as dynamic input for the SUMO simulation.

6. **Metrics Generation** (Section 3.6): Several metrics were defined to evaluate the calibration performance. The MAE, RMSE, and Bias are computed over the entire simulation period. In addition, the speed difference between the average target speed and the average simulated speed across all calibrated lanes (147 lanes) were calculated at 60-second intervals.

7 Simulation Results and Validation

The results of the SUMO simulation scenario of Esslingen demonstrate that the proposed DTC-Framework using the speed-based calibration approach significantly improves the realism of the simulated traffic dynamics and enables the simulation to closely reflect real-world traffic behavior.

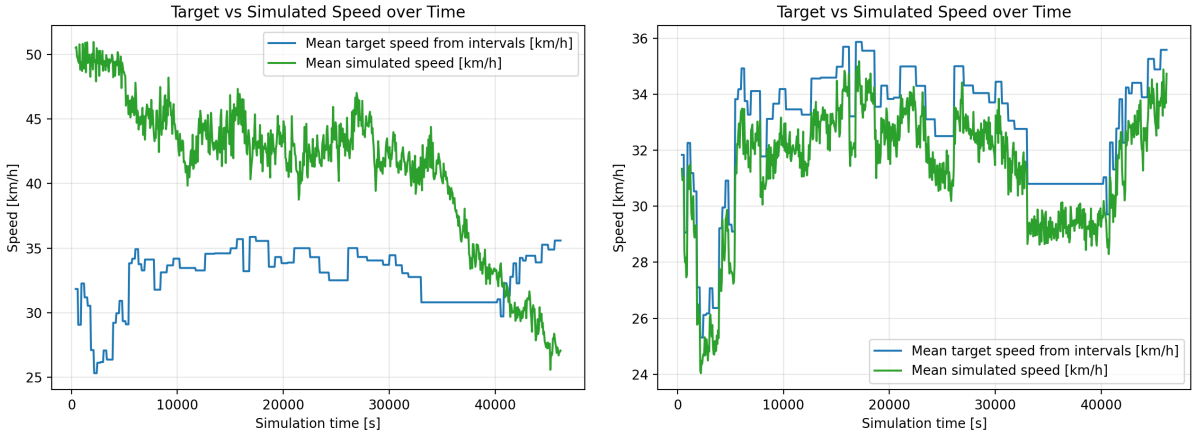

Figure 5. *Comparison of target and simulated mean lane speeds over time for uncalibrated (left) and calibrated (right) scenarios*

The evaluation is based on lane-specific speeds across all calibrated lanes as the primary performance metric, which are compared to the corresponding empirical TomTom data. Figure 5 shows the results of the target and simulated mean lane speeds over time for the uncalibrated (left) and for the calibrated (right) scenario. The right diagram in Figure 5 illustrates the time-dependent changes of the simulated average lane speed aggregated over all calibrated lanes in comparison to the target speed throughout the day. The target speed defined in the calibrators is dynamically updated during the simulation according to the TomTom speed data. The low MAE, together with the small RMSE and Bias in Table 4 (right column), indicates a strong agreement between simulated and observed speeds. This indicates that the approach is capable of closely reproducing the empirical TomTom speed data, even under varying traffic conditions over the day. Running the same scenario without any calibration further highlights the impact of the proposed approach. In the uncalibrated case (Figure 5, left diagram), substantially larger deviations from the empirical speed profiles are observed, along with more pronounced discrepancies in the temporal traffic patterns. This clearly underlines the effectiveness of the DTC-Framework in improving simulation accuracy.

Table 4. *Summary of evaluation metrics for the DTC-Framework compared to the baseline scenario*

Metric	Baseline Scenario	DTC with Speed-based calibration
MAE	16.36 km/h	3.21 km/h
RMSE	19.20 km/h	5.46 km/h
Bias	6.50 km/h	1.42 km/h

Table 4 confirms that the speed-based calibration performs significantly better than the baseline scenario, as the metrics for the baseline show substantially higher deviations compared to those from the speed-based calibration. Calibration based solely on speed data ensures consistency with observed traffic states but does not guarantee a physically consistent reconstruction of traffic demand or microscopic behavior, and should therefore be interpreted as a state calibration approach.

8 Conclusion and Future Work

This work presented the Dynamic Traffic Calibration Framework (DTC-Framework), an automated pipeline for calibrating microscopic SUMO traffic simulations using aggregated empirical speed data which are changing over daytime from the TomTom Traffic API. The framework enables a systematic integration of real-world measurements through automated data acquisition, speed mapping and calibrator generation. Its structure ensures that the baseline traffic can be reproduced while allowing continuous, data-driven adjustments based only on average speeds, without the need for OD matrices, turn counts, or vehicle trajectories. Furthermore, this work investigates the differences between SUMO's speed-based and flow-based calibration approaches and demonstrates through controlled experiments that the speed-based calibration outperforms the flow-based approach, given that only average speed data is available. Finally, the framework demonstrates strong potential for improving the accuracy and applicability of microscopic traffic simulation in real-world contexts, particularly when using the speed-based calibration approach.

Simulation results for the Esslingen scenario show that the DTC-Framework using the speed-based approach makes the SUMO traffic simulation more realistic and closely matches real-world speeds, given that the number of the simulated vehicles is set constant by a randomized base traffic generation. The evaluation using key performance metrics shows that the calibration with the DTC-Framework in the presented simulation scenario reduces the MAE to the target speed from 16.36 km/h (baseline) to 3.21 km/h, achieving a closer match to the provided TomTom data.

The following potential improvements may be considered as part of future work. Using more accurate road widths can improve the realistic representation of lanes and spatial matching, since OSM data is often incomplete or inaccurate. Furthermore, a weighted speed assignment might be an improvement, instead of using simple averages, TomTom segments could be weighted based on their overlap with SUMO lanes, yielding a more realistic calculation of lane-average speeds. Additionally, using other aggregated data sources, such as vehicle densities or the number of vehicles per hour, could further improve the simulation and might enable a closer representation of real-world traffic by allowing flow-based calibration based on observed traffic volumes. These ideas also point to opportunities for future research on data use, calibration methods, and improving scenario realism.

Data availability statement

The code of the DTC-Framework's implementation is open source and available on Github [16]. The traffic data used in this study were obtained from the TomTom Traffic API and are subject to licensing restrictions, which prohibit redistribution of the raw data. Consequently, the original data cannot be shared publicly. However, the DTC-Framework provides scripts that allow users to independently collect the necessary traffic data via the TomTom API under their own license. Processed speed maps and the DTC-Framework implementation are available in the repository for use.

Funding and Acknowledgments

The work was funded by the Federal Ministry for Economic Affairs and Climate Action under grant number *01MV23020A*. Furthermore, this work was conducted within the research project *BiFlex-Industrie* [43], which investigates bidirectional electric vehicle integration and flexibility provision in industrial and prosumer energy systems. The authors gratefully acknowledge the support and collaboration provided within the project framework.

Author contributions

F. G.: Conceptualization, Methodology, Software, Writing – original draft. A. F.: Conceptualization, Methodology, Software, Writing. S. L.: Data Curation, Writing – review & editing. E. R.: Writing – review & editing. M. S.: Writing – review & editing. All authors approved the final manuscript.

Competing interests

The authors declare that they have no competing interests.

References

[1] M. Treiber and A. Kesting, *Traffic Flow Dynamics*. Berlin/Heidelberg: Springer, 2013. DOI: 10.1007/978-3-642-32460-4.

[2] P. A. Lopez, M. Behrisch, L. Bieker-Walz, *et al.*, "Microscopic traffic simulation using sumo," in *2018 21st Int. Conf. on Intelligent Transportation Systems (ITSC)*, 2018, pp. 2575–2582. DOI: 10.1109/ITSC.2018.8569938.

[3] D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker, "Recent development and applications of sumo – simulation of urban mobility," *Int. Journal On Advances in Syst. and Measurements*, vol. 5, pp. 128–138, 2012.

[4] SUMO Project. "Sumo user documentation." Accessed: 2026-02-11. (2025), [Online]. Available: https://sumo.dlr.de/docs/index.html.

[5] PTV Group. "Ptv vissim — microscopic traffic simulation software." Accessed: 2026-02-04. (2024), [Online]. Available: https://www.ptvgroup.com/de/produkte/ptv-vissim.

[6] Aimsun. "Aimsun next traffic modeling software." Accessed: 2026-02-04. (2024), [Online]. Available: https://www.aimsun.com.

[7] Siemens Mobility. “Paramics traffic simulation.” Accessed: 2026-02-04. (2024), [Online]. Available: https://www.systra.com/digital/solutions/transport-planning/paramics/.

[8] J. Barceló, *Fundamentals of Traffic Simulation*. Springer, 2010. [Online]. Available: https://link.springer.com/book/10.1007/978-1-4419-6142-6.

[9] Y.-P. Flötteröd and P. Wagner, “SUMO Simulation of DLR’s Research Intersection,” *SUMO Conference Proceedings*, vol. 6, 2025. DOI: 10.52825/scp.v6i.2633.

[10] TomTom International BV, *Location technology for developers*, Online, 2026. [Online]. Available: https://developer.tomtom.com.

[11] D.-I. M. Körner, *Applications of floating car data for traffic flow optimization*, Accessed: 2026-02-26, 2011. [Online]. Available: https://nbn-resolving.org/urn:nbn:de:bsz:14-qucosa-93546.

[12] S. Khaleghian, H. Neema, M. Sartipi, T. Tran, R. Sen, and A. Dubey, “Calibrating real-world city traffic simulation model using vehicle speed data,” in *2023 IEEE Int. Conf. on Smart Computing (SMARTCOMP)*, 2023, pp. 303–308. DOI: 10.1109/SMARTCOMP58114.2023.00076.

[13] C. Presinger, K.-H. Kastner, K. Bosa, and M. Neubauer, “Calibrating traffic simulation models in sumo based upon diverse historical real-time traffic data — lessons learned in its upper austria,” in *Proceedings of ITS Upper Austria*, 2018.

[14] Y.-P. Flötteröd and M. Behrisch, “Online calibration with sumo for network-wide traffic and emission monitoring — case study its huainan,” *SUMO Conf. Proc.*, vol. 2, pp. 1–11, 2022. DOI: 10.52825/scp.v2i.88.

[15] T. Veihelmann, V. Shatov, M. Lübke, and N. Franchi, “Using probe counts to provide high-resolution detector data for a microscopic traffic simulation,” *Vehicles*, vol. 6, pp. 747–764, 2024. DOI: 10.3390/vehicles6020035.

[16] A. Freymann, F. Guist, and S. Lipinski. “Dynamictrafficmanagement.” (2026), [Online]. Available: https://github.com/keim-hs-esslingen/dynamicTrafficManagement.

[17] Eclipse SUMO Development Team. “Calibrator — sumo documentation.” Accessed: 2026-01-22, German Aerospace Center (DLR) / Eclipse SUMO. (2025), [Online]. Available: https://sumo.dlr.de/docs/Simulation/Calibrator.html.

[18] TomTom International BV, *Vector flow tiles*, Online, 2022. [Online]. Available: https://developer.tomtom.com/traffic-api/documentation/tomtom-maps/traffic-flow/vector-flow-tiles.

[19] Eclipse SUMO Development Team, *Simulation/output/routeprobes — sumo documentation*, Accessed: 2026-01-22, German Aerospace Center (DLR) / Eclipse SUMO, 2024. [Online]. Available: https://sumo.dlr.de/docs/Simulation/Output/RouteProbe.html.

[20] L. Codecá, R. Frank, S. Faye, *et al.*, “Luxembourg sumo traffic (lust) scenario: Traffic demand evaluation,” *IEEE Intell. Transp. Syst. Mag.*, vol. 9, pp. 52–63, 2017.

[21] L. Bieker, D. Krajzewicz, A. P. Morra, C. Michelacci, and F. Cartolano, “Traffic simulation for all: A real-world traffic scenario from the city of bologna,” in *SUMO 2014*, 2014. DOI: 10.1007/978-3-319-15024-6_4.

[22] 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*, E. G. Nathanail, N. Gavanas, and G. Adamos, Eds., Springer Nature Switzerland, 2023, pp. 276–287.

[23] M. Gueriau and I. Dusparic, “Quantifying the impact of connected and autonomous vehicles on traffic efficiency and safety in mixed traffic,” in *23rd IEEE Int. Conf. on Intelligent Transportation Systems*, 2020.

[24] S. Lobo, S. Neumeier, E. Fernandez, and C. Facchi, “Intas – the ingolstadt traffic scenario for sumo,” in *SUMO User Conf. Proc.*, 2020. DOI: https://doi.org/10.52825/scp.v1i.

[25] Y. Yamazaki, Y. Tamura, X. Défago, E. Javanmardi, and M. Tsukada, “Tost: Tokyo sumo traffic scenario,” in *The 26th IEEE Int. Conf. on Intelligent Transportation Systems*, 2023. DOI: 10.1109/ITSC57777.2023.10422517..

[26] J. Kappenberger and H. Stuckenschmidt, “Overcoming data scarcity in calibrating sumo scenarios with evolutionary algorithms,” *SUMO Conf. Proc.*, vol. 6, pp. 133–148, 2025. DOI: 10.52825/scp.v6i.2590.

[27] G. Hermanns, I. N. Kulkov, P. Hemmerle, *et al.*, “Simulations of synchronized flow in tomtom vehicle data in urban traffic with the kerner-klenov model in the framework of the three-phase traffic theory,” in *Traffic and Granular Flow ’13*, M. Chraibi, M. Boltes, A. Schadschneider, and A. Seyfried, Eds., Springer Int. Publishing, 2015, ISBN: 978-3-319-10629-8.

[28] D. A. Guastella, A. Morales-Hernández, B. Cornelis, and G. Bontempi, “Calibration of Vehicular Traffic Simulation Models by Local Optimization,” *arXiv preprint arXiv:2502.11585*, 2025. DOI: 0.1007/s11116-025-10593-x.

[29] TomTom International B.V. “Zoom levels and tile grid.” Accessed: 2026-04-28, TomTom. (2026), [Online]. Available: https://developer.tomtom.com/map-display-api/documentation/tomtom-maps/zoom-levels/zoom-levels-and-tile-grid.

[30] A. Freymann, F. Maier, K. Schaefer, and T. Böhnel, “Tackling the six fundamental challenges of big data in research projects by utilizing a scalable and modular architecture,” in *5th Int. Conf. on IoT, Big Data & Security (IoTBDS 2020)*, 2020, pp. 249–256. DOI: 10.5220/0009388602490256.

[31] Eclipse SUMO Development Team. “Osmwebwizard — sumo documentation.” Accessed: 2026-01-22, German Aerospace Center (DLR) / Eclipse SUMO. (2025), [Online]. Available: https://sumo.dlr.de/docs/Tutorials/OSMWebWizard.html.

[32] Eclipse SUMO Development Team. “Trip — sumo documentation.” Accessed: 2026-02-03, German Aerospace Center (DLR) / Eclipse SUMO. (2025), [Online]. Available: https://sumo.dlr.de/docs/Tools/Trip.html.

[33] Eclipse SUMO Development Team. “Duarouter — sumo documentation.” Accessed: 2026-02-03, German Aerospace Center (DLR) / Eclipse SUMO. (2026), [Online]. Available: https://sumo.dlr.de/docs/duarouter.html.

[34] Eclipse SUMO Development Team. “Turn — sumo documentation.” Accessed: 2026-02-03, German Aerospace Center (DLR) / Eclipse SUMO. (2025), [Online]. Available: https://sumo.dlr.de/docs/Tools/Turns.html#generateturnratiospy.

[35] Eclipse SUMO Development Team. “Route sampler — sumo documentation.” Accessed: 2026-02-03, German Aerospace Center (DLR) / Eclipse SUMO. (2025), [Online]. Available: https://sumo.dlr.de/docs/Tools/Turns.html#routesamplerpy.

[36] P. Anyin, D. Wittenberg, and J. Pannek, “Quantitative methodology for comparing microscopic traffic simulators,” *Future Transportation*, vol. 5, 2025. DOI: 10.3390/futuretransp5040201.

[37] Eclipse SUMO Development Team. “Vehiclespeed/variable speed signs— sumo documentation.” Accessed: 2026-05-04, German Aerospace Center (DLR) / Eclipse SUMO. (2025), [Online]. Available: https://sumo.dlr.de/docs/Simulation/VehicleSpeed.html#variable_speed_signs.

[38] City Population. “Esslingen am neckar population statistics.” Accessed: 2026-01-22. (2024), [Online]. Available: https://www.citypopulation.de/en/germany/badenwurttemberg/esslingen/08116019__esslingen_am_neckar/.

[39] V. Betz, *Traffic in esslingen: Everything about infrastructure in the district*, Accessed: 2026-10-22, 2020. [Online]. Available: https://www.bw24.de/baden-wuerttemberg/esslingen/esslingen-verkehr-infrastruktur-bahnhof-s-bahn-bus-landkreis-region-stuttgart-90005107.html?utm_source=chatgpt.com.

[40] AGFK-BW. "Active mobility concept esslingen: City invites participation." Accessed: 2026-01-22. (2025), [Online]. Available: https://www.agfk-bw.de/news/details/nahmobilitaetskonzept-esslingen-stadt-ruft-zur-beteiligung-auf-5275.

[41] R. J. Hyndman and A. B. Koehler, "Another look at measures of forecast accuracy," *Int. Journal of Forecasting*, vol. 22, pp. 679–688, 2006. DOI: https://doi.org/10.1016/j.ijforecast.2006.03.001.

[42] J. Hora and P. Campos, "A review of performance criteria to validate simulation models," *Expert Systems*, vol. 32, pp. 578–595, 2015. DOI: https://doi.org/10.1111/exsy.12111.

[43] BiFlex-Industrie Konsortium. "Biflex-industrie — bidirektionale flexibilität durch flottenkraftwerke in und um unternehmen." Accessed: 2026-05-08. (2026), [Online]. Available: https://www.biflexindustrie.de/.