SUMO User Conference 2026

TIBconferencesession

TIB-OP_will_set_DOI

Published: 1970-01-01

A Cycle-Level Distribution-Based Calibration of Microscopic Intersection Models Using UAV Trajectories

Charalambos Tsioutis, Konstantinos Pourgourides, and Stelios Timotheou

KIOS Research and Innovation Center of Excellence, and

Department of Electrical and Computer Engineering, University of Cyprus, Cyprus

*Correspondence: Charalambos Tsioutis, tsioutis.charalambos@ucy.ac.cy

Abstract. Accurate calibration of microscopic traffic simulation is important at signalized intersections, where errors in queue formation and turning behavior can propagate as models scale to larger urban networks. This paper presents a UAV-driven, distribution-based calibration framework for SUMO that targets cycle-level intersection dynamics rather than aggregate link counts. Using pNEUMA UAV trajectory data from Athens, we jointly calibrate key SUMO components that govern intersection behavior, including car-following (Krauss), lane-changing, and junction interaction parameters.

The observation horizon is segmented into signal cycles, and within each cycle vehicles are assigned to a signal phase and an exit direction. Cycle data are then aggregated by phase and direction to construct cumulative exit-count curves that act as distributional targets. Simulated outputs are processed in the same way, and calibration minimizes discrepancies between observed and simulated curves using the NGOpt optimizer from the Nevergrad library. To reflect UAV data gaps and avoid error accumulation across cycles, each cycle is reinitialized and simulated independently, enabling parallel evaluation of objective calls.

Results show stable convergence of NGOpt. For the dominant phase–direction movement, the normalized objective deviation (nABC) is as low as 0.46%, while the maximum cumulative deviation does not exceed two vehicles per cycle in any phase-direction pair. Compared to SUMO default parameters, median cycle-level MAPE decreases by approximately 70% in the higher-volume movements, even when both scenarios are initialized from the same UAV-derived vehicle states.

Keywords: UAV-based Traffic Analysis, Traffic Data-Analysis, Data-driven calibration and validation

1. Introduction

Microscopic traffic simulation is widely used to study traffic operations, evaluate control strategies, and support transportation planning and policy analysis. For these applications, it is essential that simulated traffic dynamics reproduce real-world behavior with sufficient realism. Calibration refers to the process of adjusting model inputs and behavioral parameters so that the simulation produces traffic patterns consistent with observed conditions. Established microsimulation guidelines emphasize calibration as a necessary step to ensure that simulation results are credible and suitable for operational use [1]–[3].

Signalized intersections are critical elements of urban traffic networks and are often locations where congestion first develops. Queues form at intersections and may spill back to upstream links, affecting adjacent junctions and corridors. Errors in representing intersection dynamics, such as queue discharge rates, lane utilization, or turning behavior, can therefore propagate to the network level [1]. Accurate modeling and calibration of signalized intersections is thus particularly important in urban microscopic simulation studies.

Calibration practice is strongly influenced by data availability. Many studies rely on stationary sensors, such as inductive loop detectors, which provide aggregated traffic counts at fixed locations [4]. Other data sources, including Bluetooth-based sensing, similarly observe traffic conditions at discrete points and may introduce detection or classification biases [5]. As a result, calibration objectives are often formulated using aggregate measures such as link flows or average speeds. While useful for capturing overall demand and flow patterns, these measures provide limited insight into phase-level dynamics and turning behavior at signalized intersections [1], [6].

Microscopic simulation platforms such as the *Simulation of Urban MObility* (SUMO) provide detailed representations of traffic processes through distinct modeling components [7]. In SUMO, car-following models describe longitudinal vehicle interactions, lane-changing models govern lateral maneuvers, and junction models control behavior at stop lines and conflict areas. These components jointly determine traffic dynamics at signalized intersections. In this study, they are calibrated together to capture their combined influence on intersection-level behavior.

Recent advances in unmanned aerial vehicle (UAV) sensing enable the observation of a top-down perspective, allowing the extraction of high-resolution vehicle trajectories across the full spatial extent of intersections. The pNEUMA experiment provides such trajectory data with high spatial and temporal resolution [8]. Compared to stationary sensors, UAV trajectories allow direct observation of turning movements, lane usage, and phase-dependent exit behavior. However, UAV data collection is often fragmented due to flight duration limits and repositioning.

This paper proposes a UAV-based calibration framework for signalized intersections in SUMO that leverages trajectory distributions rather than aggregate measurements. The observation period is segmented into individual signal cycles, and vehicle exit times are grouped by phase and direction to construct cumulative exit-count curves. Simulated outputs are processed identically, and calibration minimizes discrepancies between observed and simulated cumulative curves. To accommodate fragmented UAV observations and prevent error accumulation across cycles, each cycle is simulated independently with reinitialization. Parameter tuning is performed using

the NGOpt optimizer from the Nevergrad library. The framework jointly calibrates car-following, lane-changing, and junction-related parameters to improve representation of intersection-level traffic dynamics.

The remainder of this paper is organized as follows. Section 2 reviews relevant calibration studies. Section 3 presents the problem formulation and objective function. Section 5 describes the case study and data processing. Section 6 details the simulation and calibration setup. Section 7 reports the results, followed by conclusions and future work.

2. Related Work

Calibration of microscopic traffic simulation models has traditionally been driven by the availability of stationary sensing technologies that provide aggregated observations. Inductive loop detectors remain one of the most widely deployed sources of traffic data, supporting large-scale monitoring but imposing spatial rigidity because they observe traffic at fixed points [4]. As a result, many calibration studies formulate objectives over discrepancies between observed and simulated flows or speeds aggregated over time and space. Within this context, demand calibration has received particular attention. Bayesian approaches such as CaDyTS adjust individual-level travel demand so that simulated link flows match time-dependent count observations, using simulation-compatible fixed-point procedures rather than analytical gradients [9]. While effective for large-scale networks, these approaches primarily address demand consistency and do not explicitly calibrate microscopic behavioral parameters that govern intersection dynamics.

Because simulation-based calibration objectives are often noisy and computationally expensive, derivative-free optimization methods have been widely adopted. Simultaneous Perturbation Stochastic Approximation (SPSA) and its variants are attractive due to their low evaluation cost per iteration, which is independent of the number of calibrated parameters. Antoniou et al. demonstrated a weighted variant of SPSA (W-SPSA) that incorporates spatial-temporal correlations among measurements, enabling simultaneous calibration of demand and supply parameters in large-scale applications [10]. Lu et al. further applied SPSA-based calibration to dynamic traffic assignment models using aggregated loop-detector counts over an expressway network, minimizing normalized discrepancies across sensors and time intervals [6]. These works illustrate the scalability of such optimization methods but remain closely tied to aggregated stationary data.

Calibration studies that explicitly target signalized intersections often rely on trajectory information derived from fixed-point sensing technologies such as roadside cameras. Arafat et al. calibrated microsimulation models at multiple signal-controlled intersections by minimizing discrepancies between observed and simulated saturation headways, focusing on a limited set of car-following parameters and using manual sensitivity analysis [11]. Ranpura et al. applied a genetic algorithm to calibrate mixed-traffic microsimulation models at signalized intersections using camera-derived capacity observations [12]. Other studies have adopted sensitivity-driven or data-driven approaches for intersection calibration in commercial simulators, using fixed-point video or GPS-tagged fleet data to reproduce queue lengths, travel times, or stop frequencies [13], [14]. While these works highlight the importance of intersection-specific cal-

ibration, they typically focus on a small subset of behavioral parameters and limited temporal coverage.

An alternative line of work considers distribution-based calibration, where the objective is formulated in terms of discrepancies between observed and simulated distributions rather than aggregate statistics. Antoniou et al. proposed a distribution-based calibration framework using acceleration data, demonstrating that matching distributions can better capture variability in driving behavior [15]. Markou et al. further illustrated this approach using high-resolution trajectory data and normalized error measures [16]. These studies show the potential of distribution-based objectives but generally focus on simplified networks or individual behavioral components, and do not explicitly address joint calibration of multiple microscopic models at signalized intersections.

Within SUMO, calibration benefits from the simulator's open-source architecture and rich behavioral model set [7]. Existing SUMO calibration studies have predominantly relied on stationary aggregate targets or demand adjustment techniques, while joint calibration of car-following, lane-changing, and junction interaction parameters at signalized intersections using trajectory distributions remains comparatively underexplored. This paper addresses this gap by leveraging UAV trajectories within SUMO, building on distribution-based calibration concepts [15], [16] and scalable gradient-free optimization methods [6], [10] to propose a cycle-level calibration framework that targets phase-resolved intersection dynamics.

3. Problem Formulation and Solution Approach

3.1 Calibration Problem Formulation

Problem setup. Consider a signalized intersection operating under a fixed-time signal plan. A *traffic signal cycle* is defined as the time interval between two consecutive repetitions of the same ordered sequence of signal phases. Each cycle consists of a finite set of phases $p \in \mathcal{P}$, where a phase corresponds to a time interval during which a predefined group of traffic movements receives right-of-way.

Let the observation horizon be partitioned into C consecutive cycles indexed by $c = 1, 2, \dots, C$. For each phase p, let T_p denote its fixed duration under the signal plan.

Within each cycle, vehicles are classified according to: (i) the phase $p \in \mathcal{P}$ during which they exit the intersection, and (ii) the exit direction $d \in \mathcal{D}$ corresponding to their movement.

Exit-time sets. Let $\mathcal{N}_{c,p,d}$ denote the set of vehicles exiting the intersection during cycle c, phase p, and direction d. For each observed vehicle $i \in \mathcal{N}_{c,p,d}$, let $t^{\text{obs}}_{i,c,p,d}$ denote its observed exit time, expressed relative to the start of the observation horizon. The corresponding phase-relative exit time is defined as

$$\tau^{\text{obs}}_{i,c,p,d} = t^{\text{obs}}_{i,c,p,d} - t^{\text{start}}_{c,p}, \qquad 0 \le \tau^{\text{obs}}_{i,c,p,d} \le T_p, \tag{1}$$

where $t^{\text{start}}_{c,p}$ denotes the start time of phase p within cycle c, and T_p is the fixed phase duration.

Aggregating phase-relative exit times across all cycles yields the observed exit-time set for phase–direction pair (p, d):

$$\mathcal{T}^{\text{obs}}_{p,d} = \left\{ \tau^{\text{obs}}_{i,c,p,d} \;\middle|\; i \in \mathcal{N}_{c,p,d},\ c = 1, \dots, C \right\}. \tag{2}$$

Let $\theta \in \mathbb{R}^P$ denote the vector of microscopic simulation parameters to be calibrated, and let

$$\Theta = \{\theta \mid \ell \le \theta \le u\}$$

denote the feasible parameter domain.

Given fixed network and control inputs $\mathcal{N}$, the microscopic simulator is represented as the mapping

$$\mathcal{S}(\theta; \mathcal{N}), \tag{3}$$

which generates simulated vehicle trajectories.

Applying the same grouping procedure to simulation output yields the simulated exit-time set

$$\mathcal{T}_{p,d}^{\text{sim}}(\theta). \tag{4}$$

Although the simulator is written as a deterministic mapping, it incorporates internal stochastic mechanisms. As a result, repeated evaluations at the same parameter vector θ may produce different realizations of $\mathcal{T}_{p,d}^{\text{sim}}(\theta)$, and the corresponding objective value should therefore be interpreted as a noisy performance estimate.

Cumulative Exit-Count Curves. For each phase-direction pair (p, d), we represent exit patterns through a cumulative exit-count curve. Let $\mathcal{T}_{p,d}^{\text{obs}}$ denote the set of phase-relative exit times for observed vehicles assigned to (p, d). Exit times are aggregated across all cycles as defined in $\mathcal{T}_{p,d}^{\text{obs}}$ and $\mathcal{T}_{p,d}^{\text{sim}}(\theta)$, so each curve summarizes the phase-level discharge pattern over the full observation horizon.

The observed cumulative curve of phase $p \in \mathcal{P}$ and direction $d \in \mathcal{D}$ is defined as

$$C_{p,d}^{\text{obs}}(t) = \sum_{\tau \in \mathcal{T}_{p,d}^{\text{obs}}} \mathbf{1}\{\tau \le t\}, \qquad t \in [0, T_p], \tag{5}$$

where $\mathbf{1}\{\cdot\}$ is the indicator function. The simulated curve $C_{p,d}^{\text{sim}}(t \mid \theta)$ is defined analogously using $\mathcal{T}_{p,d}^{\text{sim}}(\theta)$.

In implementation, curves are evaluated on a discrete grid $t \in \{0, 1, \dots, T_p\}$ to match the time resolution used in post-processing.

Objective Function. The calibration objective $z(\theta)$ considers the minimization of the weighted average absolute deviation between observed and simulated cumulative exit-count curves as

$$z(\theta) = \sum_{p \in \mathcal{P}} \sum_{d \in \mathcal{D}} w_{p,d}\, \Delta_{p,d}(\theta), \tag{6}$$

where $w_{p,d} \ge 0$ are the optimization weights with $\sum_{(p,d)} w_{p,d} = 1$.

Here $\Delta_{p,d}(\theta)$ denotes the deviation between observed and simulated cumulative exit-count curves defined as

$$\Delta_{p,d}(\theta) = \frac{1}{T_p + 1} \sum_{t=0}^{T_p} \left| C_{p,d}^{\text{obs}}(t) - C_{p,d}^{\text{sim}}(t \mid \theta) \right|. \tag{7}$$

The calibration problem is therefore formulated as

$$\min_{\theta \in \Theta} z(\theta) \tag{8a}$$

subject to:

$$\begin{cases} \mathcal{S}(\theta; \mathcal{N}) & \text{(8b)} \\ \ell \leq \theta \leq u. & \text{(8c)} \end{cases}$$

The objective function $z(\theta)$ is evaluated through microscopic simulation and does not admit a closed-form analytical expression, as the mapping $\mathcal{S}(\theta; \mathcal{N})$ is highly non-linear, and the resulting response surface may be non-convex with multiple local minima. Furthermore, stochastic components of the simulator introduce noise in objective evaluations [10], [17]. These characteristics motivate the use of derivative-free optimization methods suitable for noisy black-box problems.

3.2 Solution Algorithm

Optimization problem (8) is solved using the NGOpt optimizer from the Nevergrad library [18]. NGOpt is a derivative-free optimization framework designed for high-dimensional black-box objective functions.

At iteration k, the optimizer proposes a candidate θ_k. The simulator $\mathcal{S}(\theta_k; \mathcal{N})$ is executed, cumulative curves are constructed, and the objective value $z(\theta_k)$ is computed. The optimizer updates its internal search state based on observed objective values until predefined maximum iterations are exhausted.

To prevent error accumulation across cycles and reflect fragmented UAV observations, each signal cycle is simulated independently with reinitialization. Because cycles are independent under this design, simulations can be executed in parallel during each objective evaluation.

4. Sensitivity Analysis

The parameter vector θ includes a large number of microscopic behavioral parameters associated with car-following, lane-changing, and junction interaction models. Calibrating all parameters simultaneously increases computational cost and may introduce unnecessary variability in the search process. Therefore, a preliminary screening step was performed to identify parameters that exhibit measurable influence on the objective function defined in Eq. (6).

An experimental dataset was generated by sampling parameter combinations within their predefined bounds and evaluating the corresponding objective values. For each sampled parameter vector $\theta^{(k)}$, the objective value $z^{(k)}$ was computed using the procedure described in Section 3, resulting in

$$\left\{\theta^{(k)},\ z^{(k)}\right\}_{k=1}^{K}. \tag{9}$$

Here $k \in \{1, \ldots, K\}$ indexes independent parameter samples drawn within the predefined bounds.

To assess parameter influence, an ordinary least squares (OLS) regression model was fitted:

$$z^{(k)} = \beta_0 + \sum_{j=1}^{P} \beta_j \theta_j^{(k)} + \varepsilon^{(k)}, \tag{10}$$

where β_j denotes the regression coefficient associated with parameter θ_j and $\varepsilon^{(k)}$ is the residual term.

The regression model is used strictly as a statistical screening tool to detect associations between sampled parameters and objective values. No assumption is made that the true relationship between θ and z is linear.

A parameter θ_j was retained if

$$\begin{cases} p_j < 0.05, \\ |\beta_j| > \varepsilon, \end{cases} \tag{11}$$

where p_j is the p-value of coefficient β_j and $\varepsilon = 10^{-3}$ is a minimum magnitude threshold. Because raw OLS coefficient magnitudes depend on parameter scaling and sampling ranges, $|\beta_j|$ is not used to compare parameters. Instead, the threshold is used only as a practical filter to remove near-zero effects under the sampled bounds. Parameters not meeting these conditions were fixed at their default values and excluded from subsequent optimization.

No assumptions of global linearity are imposed, here the regression model is used solely as a local screening tool to identify parameters exhibiting meaningful variation with respect to the objective.

5. Case Study

The proposed calibration framework is evaluated on a real-world signalized urban intersection using high-resolution UAV trajectory data from the pNEUMA experiment [8]. This section describes the study area, data preparation, traffic signal reconstruction, and the definition of phase–direction groupings used for calibration.

5.1 Study Area and Network Representation

The case study focuses on a signalized intersection formed by Panepistimiou Avenue and Amerikis Street in central Athens. The main approach consists of five lanes along Panepistimiou Avenue, intersecting with a two-lane cross street along Amerikis. This location was selected due to the availability of high-resolution UAV trajectory data from the pNEUMA experiment [8] and its heterogeneous traffic composition.

Figure 1 presents the real-world layout (Left) and the corresponding SUMO network model (Right). The microscopic network reproduces the geometric layout, lane configuration, and signal control scheme of the physical intersection.

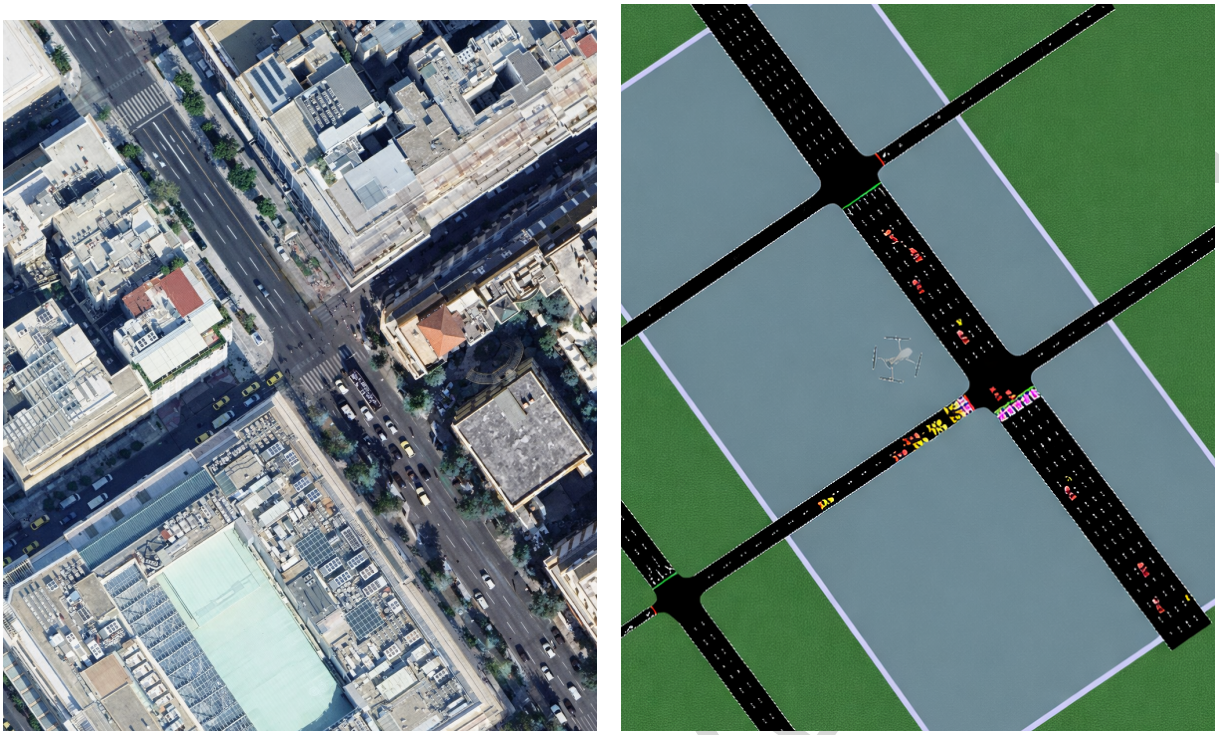

Figure 1. *Left: Satellite image of the Amerikis–Panepistimiou intersection. Right: Corresponding SUMO network including lane geometry, signalization, virtual detectors (pink color), and the UAV field of view (FoV). The modeled network extends beyond the intersection shown in the UAV FoV to capture interactions with nearby signalized intersections.*

5.2 UAV Trajectory Data

The pNEUMA dataset provides time-stamped vehicle trajectories extracted from aerial video recordings over the study intersection [8]. Each trajectory contains vehicle positions at discrete time steps, enabling reconstruction of vehicle paths and exit times.

For each vehicle, the exit time from the intersection is defined as the timestamp at which the trajectory crosses a predefined exit boundary. Vehicles are assigned to:

- a signal phase $p \in \mathcal{P}$ active at the time of exit, and
- an exit direction $d \in \mathcal{D}$ corresponding to the movement performed.

For each (p, d), phase-relative exit times are computed and aggregated across cycles as described in Section 3.

5.3 Traffic Signal Phase Identification

Traffic light phases are inferred directly from the UAV trajectory data following the methodology described in [19]. Virtual detectors are placed at the real positions of traffic light poles, and flow counts are measured at those locationsFigure 1(Right).

Because the trajectory data are discrete in time, flow measurements occur at distinct time steps. The registered count at a given time step corresponds to the number of vehicles that crossed the virtual traffic light pole between the previous and the current time step.

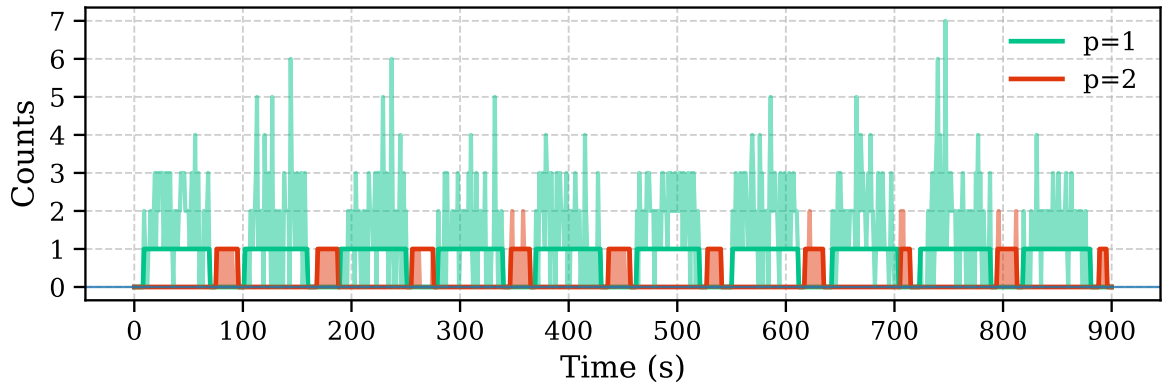

Figure 2. *Traffic light phase identification using virtual detector flow measurements at the study intersection. Vehicle counts are shown over time, and horizontal lines indicate grouped time windows corresponding to identified phases.*

For each approach, counts that belong to the same phase are registered within a specific time window, while no counts are observed for multiple consecutive time steps after the window ends, until the next cycle repetition.

Groups of registered counts therefore correspond to traffic light phases, and the duration of each phase equals the length of its associated time window. The observation horizon is segmented into consecutive cycles according to the repetition of the identified phase sequence. Only fully observed cycles are retained for calibration.

Figure 2 illustrates the phase identification procedure for the study intersection.

5.4 Cycle Segmentation

Two signal phases ($p = 1, 2$) and two exit directions ($d = 1, 2$) are considered. This yields four phase-direction pairs (p, d) used in Eq. (6). Figure 3 illustrates the adopted definition.

The pNEUMA trajectories span multiple observation periods. Following traffic signal reconstruction, the data are segmented into consecutive signal cycles. For each cycle, trajectories are assigned to a phase p active at exit time and to an exit direction d according to the movement performed. For each (p, d), phase-relative exit times are computed within each cycle and then aggregated across cycles to construct the observed cumulative exit-count curves. The same pair construction is applied to simulation outputs. Each cycle is later simulated independently using a cycle-specific route file and UAV-derived initial vehicle positions and speeds at cycle start.

6. Simulation and Calibration Setup

This section describes the calibrated parameter sets, the cycle-based simulation setup, and the optimization configuration used to solve Eq. (8). Figure 4 summarizes the end-to-end workflow of the proposed framework.

The pNEUMA dataset contains a substantial proportion of motorcycles, reflecting typical urban traffic composition in Athens [8]. To represent lateral positioning and overtaking behavior characteristic of motorcycles, the SUMO sublane model is activated.

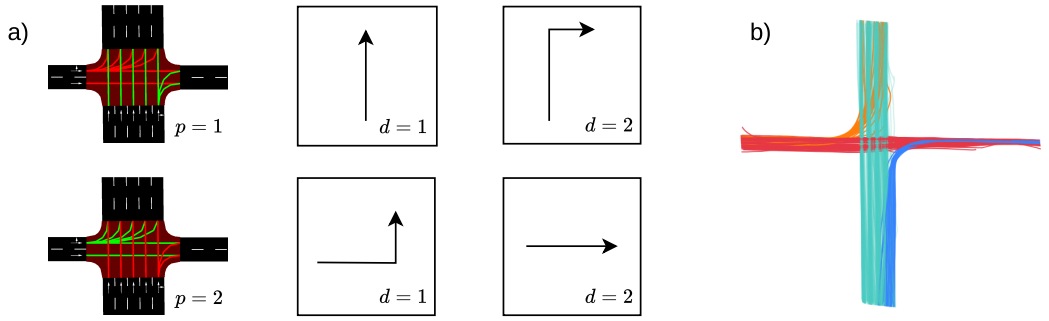

Figure 3.
a) Phase–direction grouping at the study intersection. Top row: Phase $p = 1$ with exit directions $d = 1$ and $d = 2$. Bottom row: Phase $p = 2$ with exit directions $d = 1$ and $d = 2$.
b) UAV-derived vehicle trajectories colored by direction.

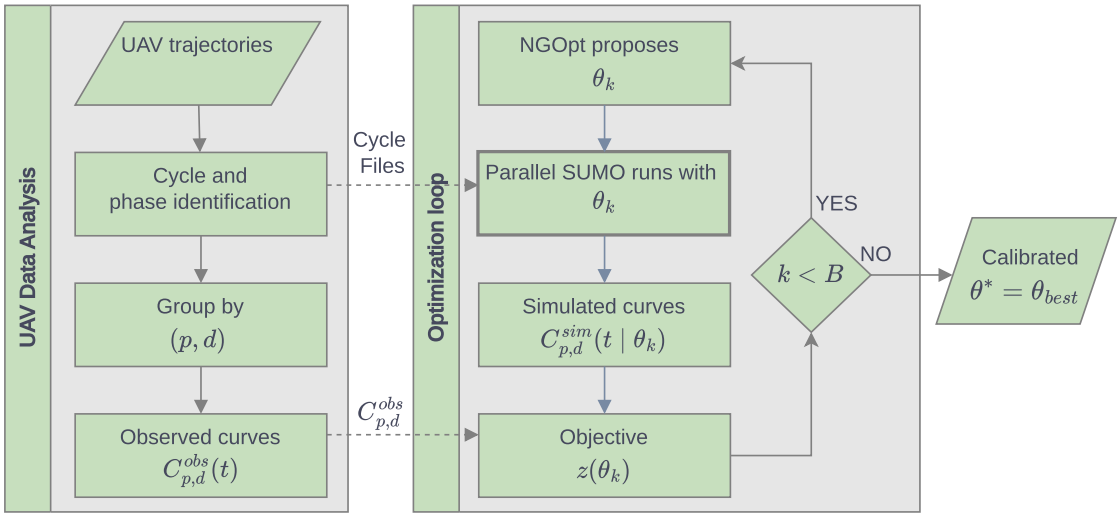

Figure 4. *Overview of the calibration workflow. UAV trajectories are processed to infer signal phases, segment cycles, and construct phase-direction cumulative exit-count curves. The NGOpt optimizer proposes parameters θ, SUMO is executed per cycle in parallel under fixed inputs $\mathcal{N}$, and the objective $z(\theta)$ is evaluated according to Eq.* (6)

At a high level, the sublane model allows vehicles to occupy fractional lane positions and perform lateral movements within a lane, enabling more realistic representation of mixed traffic conditions. A detailed analysis of the sublane model mechanics is beyond the scope of this study.

6.1 Calibrated Parameters Across Vehicle Types

The calibrated decision vector θ includes both (i) *global* parameters that apply across all vehicle types and (ii) *vehicle-type-specific* parameters that capture heterogeneity across modes. A detailed description of SUMO vehicle and model parameters is available in the official documentation[1].

Global parameters are used to tune lane-changing and junction-interaction behavior that affects all vehicles. Vehicle-type-specific parameters are used to model differ-

[1] https://sumo.dlr.de/docs/

ences in desired speed, acceleration capabilities, and car-following behavior across categories.

In this study, vehicle-type-specific parameters are calibrated for the set of types

$$\mathcal{V} = \{\texttt{Car}, \texttt{Bus}, \texttt{Taxi}, \texttt{Motorcycle}, \texttt{Medium}, \texttt{Heavy}\}.$$

For each vehicle type $v \in \mathcal{V}$, the calibrated subset includes parameters such as `speedDev`, `speedFactor`, `minGap`, `accel`, `decel`, `startupDelay`, `sigma`, and `tau`. Global parameters include lane-changing parameters (e.g., `lcSpeedGain`, `lcKeepRight`, `minGapLat`) and junction-model parameters (e.g., `jmSigmaMinor`, `jmStoplineGap`, `jmTimegapMinor`, `impatience`). Parameter bounds ℓ and u are selected based on physically meaningful ranges and prior modeling experience.

Following the screening procedure in Section 4, parameters that did not satisfy the selection criteria were fixed to their default values, and only the retained subset was optimized.

6.2 Cycle-Based Simulation with Reinitialization

To align simulation outputs with UAV observations and to avoid error accumulation across cycles, each signal cycle $c = 1, \ldots, C$ is simulated independently. For each cycle, the simulator is initialized using cycle-specific inputs derived from the UAV data, while network and control inputs remain fixed in $\mathcal{N}$.

Let t_c^{start} denote the start time of cycle c. From UAV trajectories, an initial set of vehicles present in the intersection vicinity at t_c^{start} is identified. For each such vehicle, the initial state includes at minimum position and speed at t_c^{start}. These states are used to initialize the simulation for cycle c so that the simulated system begins from a consistent snapshot of observed traffic conditions.

Each cycle simulation is executed over the full cycle duration under the reconstructed signal plan (Section 5). After simulation, vehicle exit times are extracted and mapped to (p, d), from which the cumulative exit-count curves $C_{p,d}^{\text{sim}}(t \mid \theta)$ are constructed. Objective evaluation then follows Eq. (6).

Because cycles are independent in this design, cycle simulations can be executed in parallel for each candidate θ. Figure 5 illustrates the cycle-based execution used to evaluate one objective function call.

6.3 Optimization Configuration

Calibration is performed using the NGOpt optimizer from the Nevergrad library [18]. Optimization starts from SUMO default parameter values and enforces bound constraints $\ell \le \theta \le u$.

The objective in Eq. (6) is evaluated with $B = 700$ objective calls. For each candidate parameter vector, all cycles are simulated in parallel and the weighted discrepancy between observed and simulated cumulative exit-count curves is computed.

Weights $w_{p,d}$ are chosen to emphasize movements with higher observed vehicle volumes. In this study, (p, d) with larger counts are assigned higher weights so that the optimization prioritizes the dominant exit patterns.

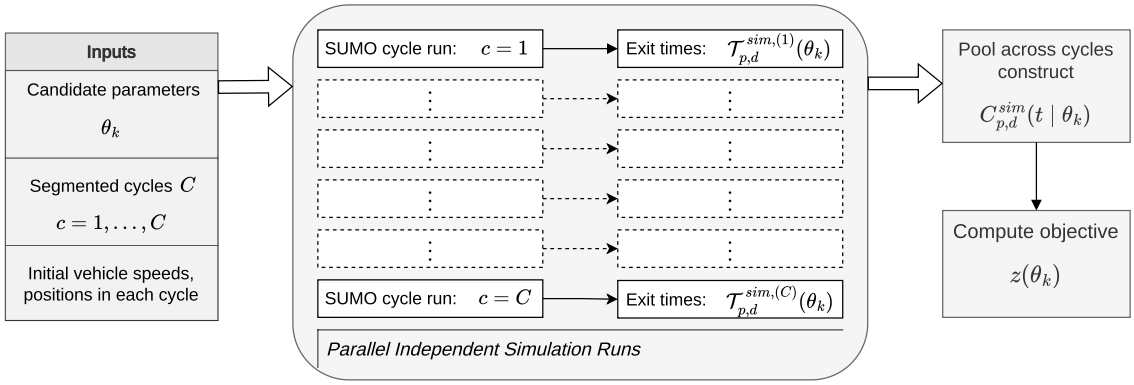

Figure 5. *Cycle-based objective evaluation. For a given parameter vector θ, each cycle c is initialized from UAV-derived states, simulated independently, and post-processed to obtain $C_{p,d}^{\text{sim}}(t \mid \theta)$. The objective $z(\theta)$ is computed after aggregating per-cycle outputs into the phase-direction cumulative curves.*

Algorithm 1 Cycle-Based UAV Calibration Procedure

Require: UAV trajectory data, network inputs $\mathcal{N}$, parameter bounds ℓ, u, maximum iteration number B
Ensure: Calibrated parameter vector θ^*
  Initialize θ_0 with SUMO default parameters
  Initialize NGOpt optimizer within bounds $\ell \le \theta \le u$
  for $k = 1$ to B **do**
    $\theta_k \leftarrow$ NGOpt.ask()
    for each cycle $c = 1, \ldots, C$ **in parallel do**
      Initialize SUMO using UAV-derived vehicle states at cycle start
      Run SUMO simulation for one signal cycle
      Extract simulated exit times
      Construct $C_{p,d}^{\text{sim}}(t \mid \theta_k)$
    end for
    Compute objective value $z(\theta_k)$ using Eq. (6)
    NGOpt.tell$(\theta_k, z(\theta_k))$
  end for
  $\theta^* \leftarrow$ NGOpt.recommend()
  return θ^*

7. Results

Sensitivity screening identified 16 parameters with statistically significant association with the objective function defined in Eq. (6). The screening dataset included $K = 75$ parameter samples used for the OLS regression fit.

Table 1 lists the retained parameters together with their estimated coefficients, standard errors, and t-statistics. Car-following parameters (`speedDev`, `speedFactor`, `tau`) exhibit the largest coefficients, indicating strong influence on phase-level exit-count dynamics. Several lane-changing parameters (`lcPushy`, `lcImpatience`, `lcAssertive`, `lcSigma`) and two junction parameters (`jmDriveAfterRedTime`, `jmDriveAfterYellowTime`) were also retained. All selected parameters satisfied the criteria $p < 0.05$ and $|\beta_j| > 0.001$, while the remaining parameters were fixed at their default SUMO values.

Table 1. *Parameters retained after sensitivity screening.*

Parameter	Coefficient	Std. Error	t-statistic
speedDev	1042.19	15.51	67.21
speedFactor	-552.74	7.02	-78.74
tau	304.22	4.63	65.73
lcPushy	-137.65	6.20	-22.19
sigma	59.73	12.04	4.96
lcImpatience	-38.83	3.10	-12.52
minGap	36.17	2.78	13.01
accel	-16.69	1.30	-12.88
lcCooperativeSpeed	-14.85	6.20	-2.39
minGapLat	-12.43	3.60	-3.45
lcAssertive	-11.91	0.72	-16.46
lcSigma	-8.89	0.62	-14.33
jmDriveAfterRedTime	5.22	0.54	9.67
jmDriveAfterYellowTime	2.41	0.54	4.47
lcLookaheadLeft	-1.98	0.85	-2.33
lcStrategic	-0.35	0.06	-5.57

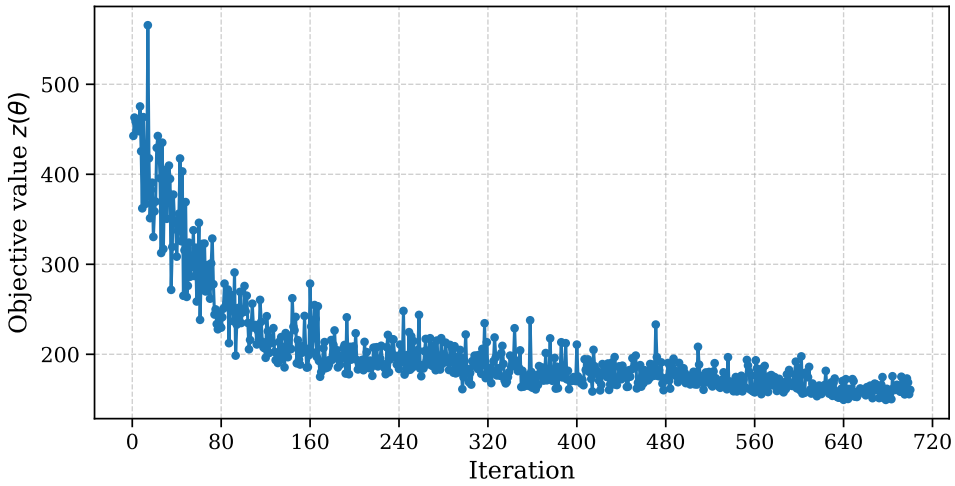

Figure 6. *Objective value as a function of optimization iteration.*

Figure 6 shows the evolution of the objective value $z(\theta)$ during optimization. The objective decreases rapidly in early iterations and gradually stabilizes toward the maximum number of objective evaluations, indicating stable convergence of the NGOpt algorithm without oscillatory behavior.

Calibration performance is evaluated using metrics derived from the objective function defined in Eq. (6). The phase–direction contribution $\Delta_{p,d}$ is defined in Eq. (7) and represents the average absolute deviation between observed and simulated cumulative exit-count curves.

To express this deviation relative to traffic volume, we define the normalized area between curves (nABC) as

$$\mathrm{nABC}_{p,d} = 100 \cdot \frac{1}{N_{p,d}} \frac{1}{T_p + 1} \sum_{t=0}^{T_p} \left| C_{p,d}^{\mathrm{obs}}(t) - C_{p,d}^{\mathrm{sim}}(t) \right| = \frac{100}{N_{p,d}} \Delta_{p,d}, \tag{12}$$

where $N_{p,d} = \frac{1}{C} C_{p,d}^{\text{obs}}(T_p)$ is the average number of observed vehicles per cycle. The nABC metric is therefore a normalized version of the objective contribution $\Delta_{p,d}$ expressed as a percentage of traffic volume.

As an additional diagnostic indicator, we report the maximum cumulative deviation per cycle,

$$\mathcal{D}_{p,d} = \frac{1}{C} \max_{t \in \{0,1,\ldots,T_p\}} \left| C_{p,d}^{\text{obs}}(t) - C_{p,d}^{\text{sim}}(t) \right|, \tag{13}$$

which represents the largest vertical deviation between cumulative curves in vehicles per cycle.

Figure 7 presents the observed and calibrated cumulative exit-count curves for each phase–direction pair. Across all pairs, nABC values indicate strong agreement between observed and simulated exit-count dynamics. The corresponding nABC values are 0.46% for $(1,1)$, 16.0% for $(1,2)$, 16.0% for $(2,1)$, and 15.0% for $(2,2)$. The maximum cumulative deviation per cycle does not exceed two vehicles in any phase–direction pair, with $\mathcal{D} = 1$ veh/cycle for $(1,1)$ and $(2,1)$ and $\mathcal{D} = 2$ veh/cycle for $(1,2)$ and $(2,2)$.

To assess performance at the individual cycle level, the Mean Absolute Percentage Error (MAPE) was computed per cycle and per phase–direction pair. MAPE was defined as

$$\text{MAPE}_{p,d}^{(c)} = 100 \cdot \frac{1}{T_p + 1} \sum_{t=0}^{T_p} \frac{\left| C_{p,d}^{\text{obs}}(t) - C_{p,d}^{\text{sim}}(t) \right|}{\max\left(C_{p,d}^{\text{obs}}(t), 1 \right)}. \tag{14}$$

Figure 8 summarizes the distribution of cycle-level MAPE values. Substantial improvements are observed in higher-volume phase–direction pairs $(1,1)$ and $(2,1)$, where median MAPE decreases by approximately 70% after calibration. Lower-volume pairs exhibit smaller relative improvements, consistent with the volume-weighted structure of the objective function.

Even when simulations are initialized from observed vehicle states, default SUMO parameters do not adequately reproduce phase-level exit-count dynamics, confirming the necessity of behavioral parameter calibration.

8. Conclusions and Future Work

This paper presented a cycle-level, distribution-based calibration approach for signalized intersections in SUMO using UAV trajectory data. The methodology segments observations into signal cycles and organizes vehicle trajectories by phase and direction. Cumulative exit-count curves are constructed and used directly within a simulation-based objective function. Calibration is performed using a derivative-free optimization algorithm under parameter bounds.

The results demonstrate that the proposed framework substantially improves the reproduction of intersection-level dynamics. For the dominant phase–direction movement, the normalized objective deviation (nABC) is reduced to 0.46%, and the maximum

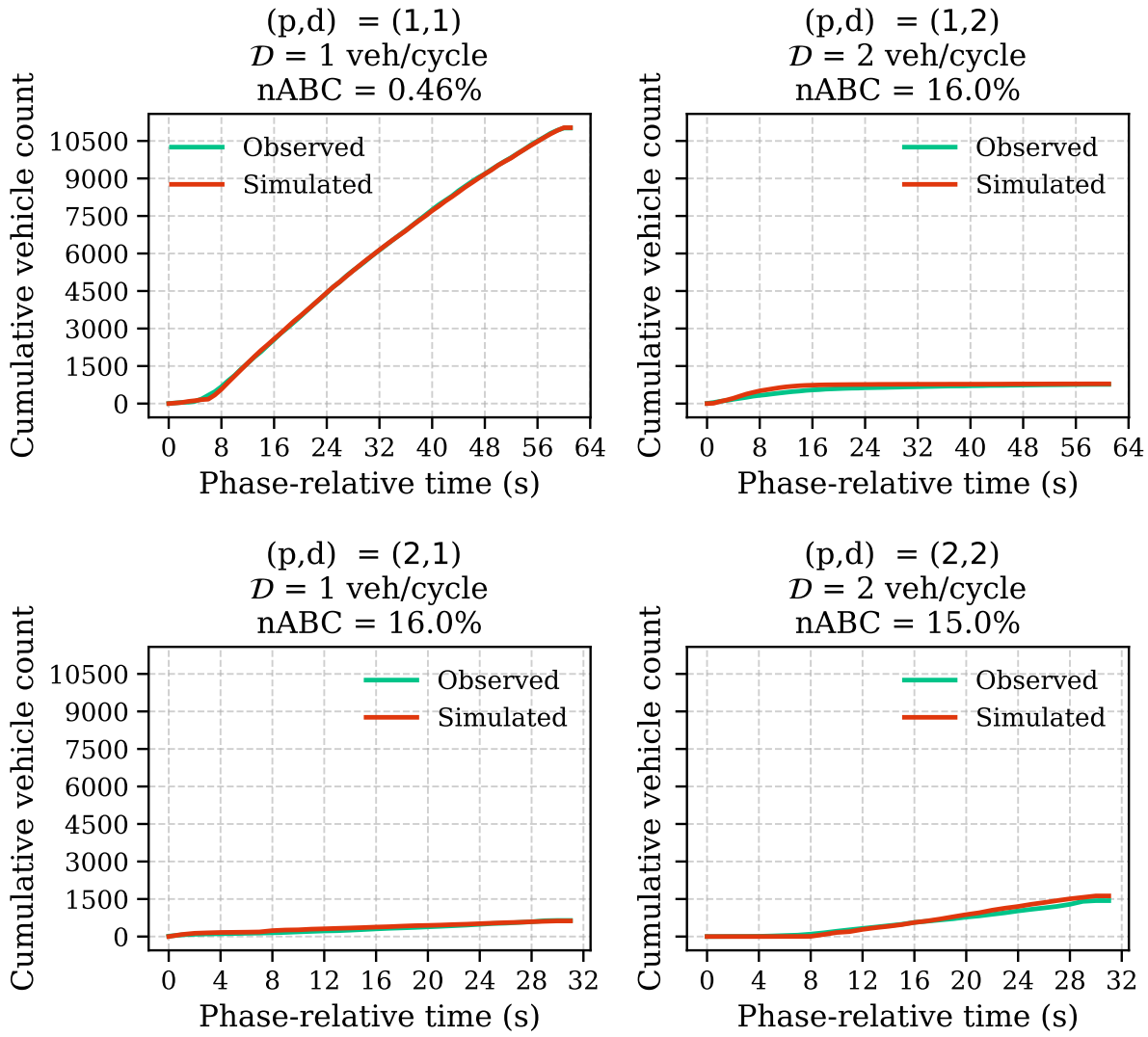

Figure 7. *Observed and calibrated cumulative exit-count curves per (p, d). $\mathcal{D}$ denotes the maximum cumulative deviation per cycle (veh/cycle), defined in Eq. (**??**). nABC denotes the normalized area between curves (percentage of observed vehicles per cycle).*

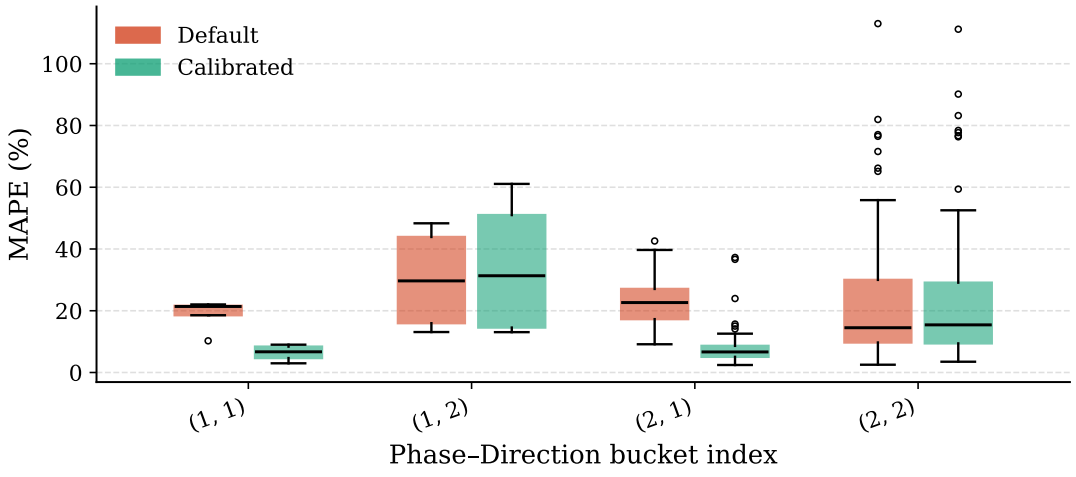

Figure 8. *Cycle-level MAPE comparison between default and calibrated configurations across (p, d).*

cumulative deviation does not exceed two vehicles per cycle in any phase–direction pair. Median cycle-level MAPE decreases by approximately 70% in higher-volume movements relative to default SUMO parameters. These findings show that, even when simulations are initialized with UAV-derived vehicle positions and speeds, default behavioral parameters are insufficient to reproduce realistic phase-level discharge patterns. Behavioral calibration is therefore necessary for accurate microscopic intersection modeling.

The framework jointly calibrates car-following, lane-changing, and junction interaction parameters, including vehicle-type–specific parameters. The complete calibrated parameter set is provided as supplementary material to support reproducibility [2].

Future work will extend the methodology to intersections with more complex geometries and to sites with different traffic compositions, particularly locations with lower motorcycle penetration. The parallel cycle-based structure can support more advanced optimization algorithms with larger computational cost. In addition, automated hyperparameter tuning strategies, including machine learning–based approaches, may further improve calibration efficiency and robustness.

Data availability statement

The codes used for this paper is open source and available on github: https://github.com/TsioutisCh/phase-relative-sumo-calibration. The calibrated parameter file is also included.

Author contributions

Charalambos Tsioutis: Conceptualization, Writing – original draft, Data curation, Software, Methodology. Konstantinos Pourgourides: Conceptualization, Data curation, Software. Stelios Timotheou: Conceptualization, Writing – review & editing, Supervision.

Competing interests

The authors declare that they have no competing interests.

Funding

This work is supported by the European Union (i. ERC, URANUS, No. 101088124 and ii. Horizon 2020 Teaming, KIOS CoE, No. 739551), and the Government of the Republic of Cyprus through the Deputy Ministry of Research, Innovation, and Digital Strategy. Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council Executive Agency. Neither the European Union nor the granting authority can be held responsible for them.

[2] https://github.com/TsioutisCh/phase-relative-sumo-calibration

References

[1] C. Antoniou, J. Barcelo, M. Brackstone, *et al.*, "Traffic simulation: Case for guidelines," 2014.

[2] Federal Highway Administration, "Traffic analysis toolbox volume iii: Guidelines for applying traffic microsimulation modeling software," U.S. Department of Transportation, Federal Highway Administration, Office of Operations, Tech. Rep. FHWA-HOP-18-036, Apr. 2019. [Online]. Available: https://ops.fhwa.dot.gov/publications/fhwahop18036/index.htm.

[3] R. Dowling, A. Skabardonis, J. Halkias, G. McHale, and G. Zammit, "Guidelines for calibration of microsimulation models: Framework and applications," *Transportation Research Record*, vol. 1876, no. 1, pp. 1–9, 2004. DOI: 10.3141/1876-01. [Online]. Available: https://doi.org/10.3141/1876-01.

[4] I. Tsapakis, P. Anderson, D. Florence, *et al.*, "Leveraging existing traffic signal assets to obtain quality traffic counts: A guide," Tech. Rep., 2025.

[5] S. Carrese, E. Cipriani, U. Crisalli, A. Gemma, and L. Mannini, "Bluetooth traffic data for urban travel time forecast," *Transportation Research Procedia*, vol. 52, pp. 236–243, 2021, 23rd EURO Working Group on Transportation Meeting, EWGT 2020, 16-18 September 2020, Paphos, Cyprus, ISSN: 2352-1465. DOI: https://doi.org/10.1016/j.trpro.2021.01.027.

[6] L. Lu, Y. Xu, C. Antoniou, and M. Ben-Akiva, "An enhanced spsa algorithm for the calibration of dynamic traffic assignment models," *Transportation Research Part C: Emerging Technologies*, vol. 51, pp. 149–166, 2015, ISSN: 0968-090X. DOI: https://doi.org/10.1016/j.trc.2014.11.006.

[7] P. Alvarez Lopez, M. Behrisch, L. Bieker-Walz, *et al.*, "Microscopic traffic simulation using sumo," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, IEEE, Nov. 2018, pp. 2575–2582. DOI: 10.1109/ITSC.2018.8569938. [Online]. Available: https://elib.dlr.de/127994/.

[8] E. Barmpounakis and N. Geroliminis, "On the new era of urban traffic monitoring with massive drone data: The pneuma large-scale field experiment," *Transportation Research Part C: Emerging Technologies*, vol. 111, pp. 50–71, 2020, ISSN: 0968-090X. DOI: https://doi.org/10.1016/j.trc.2019.11.023.

[9] G. Flötteröd, M. Bierlaire, and K. Nagel, "Bayesian demand calibration for dynamic traffic simulations," *Transportation Science*, vol. 45, no. 4, pp. 541–561, 2011. DOI: 10.1287/trsc.1100.0367.

[10] C. Antoniou, C. L. Azevedo, L. Lu, F. Pereira, and M. Ben-Akiva, "W–spsa in practice: Approximation of weight matrices and calibration of traffic simulation models," *Transportation Research Procedia*, vol. 7, pp. 233–253, 2015, 21st International Symposium on Transportation and Traffic Theory Kobe, Japan, 5-7 August, 2015, ISSN: 2352-1465. DOI: https://doi.org/10.1016/j.trpro.2015.06.013.

[11] M. Arafat, S. R. Nafis, E. Sadeghvaziri, and F. Tousif, "A data-driven approach to calibrate microsimulation models based on the degree of saturation at signalized intersections," *Transportation Research Interdisciplinary Perspectives*, vol. 8, p. 100 231, 2020, ISSN: 2590-1982. DOI: https://doi.org/10.1016/j.trip.2020.100231.

[12] P. Ranpura, R. Gujar, and S. Singh, "Development of mixed traffic microsimulation model calibration for signalized intersections," *Transportation Research Procedia*, vol. 82, pp. 2898–2910, 2025, World Conference on Transport Research - WCTR 2023 Montreal 17-21 July 2023, ISSN: 2352-1465. DOI: https://doi.org/10.1016/j.trpro.2024.12.226.

[13] N. Jayasooriya and S. Bandara, "Calibrating and validating vissim microscopic simulation software for the context of sri lanka," in *2018 Moratuwa Engineering Research Conference (MERCon)*, 2018, pp. 494–499. DOI: 10.1109/MERCon.2018.8421918.

[14] N. Li and Y. Sun, "Vissim parameter calibration based on traffic characteristics distribution at signalized intersections," in *2018 IEEE 7th Data Driven Control and Learning Systems Conference (DDCLS)*, 2018, pp. 150–153. DOI: 10.1109/DDCLS.2018.8515913.

[15] C. Antoniou, V. Gikas, V. Papathanasopoulou, T. Mpimis, I. Markou, and H. Perakis, “Towards distribution-based calibration for traffic simulation,” in *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, 2014, pp. 786–791. DOI: 10.1109/ITSC.2014.6957785.

[16] I. Markou, V. Papathanasopoulou, and C. Antoniou, “A demonstration of distribution-based calibration,” in *2015 International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS)*, 2015, pp. 109–115. DOI: 10.1109/MTITS.2015.7223244.

[17] Y. Liu, B. Zou, A. Ni, L. Gao, and C. Zhang, “Calibrating microscopic traffic simulators using machine learning and particle swarm optimization,” *Transportation Letters*, vol. 13, no. 4, pp. 295–307, 2021, ISSN: 1942-7867. DOI: https://doi.org/10.1080/19427867.2020.1728037.

[18] J. Rapin and O. Teytaud, *Nevergrad - A gradient-free optimization platform*, https://GitHub.com/FacebookResearch/Nevergrad, 2018.

[19] K.Pourgourides, *UAVTrafficPy: Package for analysis & visualization of uav-based traffic data in python*, https://github.com/KPourgourides/UAVTrafficPy, GitHub repository, 2026.