

# Fine-Grained Trajectory Reconstruction by Microscopic Traffic Simulation With Dynamic Data-Driven Evolutionary Optimization

Htet Naing<sup>ID</sup>, Wentong Cai<sup>ID</sup>, Senior Member, IEEE, Jinqiang Yu, Jinghui Zhong<sup>ID</sup>, and Liang Yu

**Abstract**—Vehicle trajectory data are essential in smart mobility applications, yet often incomplete, necessitating systematic reconstruction for effective use. Existing methods often overlook traffic rules and vehicle interactions in their reconstruction process, a research gap that becomes critical for fine-grained reconstruction of incomplete and irregular microscopic traffic data. To address this limitation, this paper introduces a novel fine-grained trajectory reconstruction (FTR) framework, particularly for urban signalized intersections, considering both traffic rules and vehicle interactions through a microscopic traffic simulation (MTS) model. This is motivated by challenging missing patterns in real-world data from Alibaba City Brain Lab and limitations in existing reconstruction approaches. To this end, the FTR problem is first formulated as an MTS-based optimization problem. Then, to solve this problem effectively under a limited computing budget, an advanced dynamic data-driven evolutionary optimization technique, D3GA++, is proposed. Through the validation involving two real-world datasets, D3GA++ has demonstrated superior performance under various missing data scenarios consistently surpassing baselines such as brute-force random search and standard evolutionary algorithm in terms of reconstruction accuracy. Our work can have crucial implications for traffic management, urban planning, and autonomous vehicle technology development.

**Index Terms**—Trajectory reconstruction, microscopic traffic simulation, data-driven evolutionary optimization, surrogate-assisted evolutionary optimization.

## I. INTRODUCTION

RECENT advancements in sensing technology and mobile devices have facilitated the collection of massive urban trajectory data [1]. These data are important for a variety of applications including crowd behavior analysis, tourism

Received 28 December 2023; revised 24 June 2024 and 4 October 2024; accepted 11 November 2024. This work was supported in part by the RIE2025 Industry Alignment Fund – Industry Collaboration Projects (IAF-ICP), Administered by Agency for Science, Technology and Research (A\*STAR) under Award I2301E0026; and in part by Alibaba Group and Nanyang Technological University (NTU) Singapore through Alibaba-NTU Global e-Sustainability CorpLab (ANGEL). The Associate Editor for this article was F. Xia. (*Corresponding author: Htet Naing.*)

Htet Naing was with the College of Computing and Data Science, Nanyang Technological University, Singapore 639798. He is now with Centre for Climate Research Singapore, Singapore 537054 (e-mail: htetnaing001@e.ntu.edu.sg).

Wentong Cai is with the College of Computing and Data Science, Nanyang Technological University, Singapore 639798 (e-mail: aswtcai@ntu.edu.sg).

Jinqiang Yu and Liang Yu are with the City Brain Laboratory, Alibaba Cloud, Hangzhou 311121, China (e-mail: yujinqiang.yjq@alibaba-inc.com; liangyu.yl@alibaba-inc.com).

Jinghui Zhong is with the School of Computer Science and Engineering, South China University of Technology, Guangzhou 510641, China (e-mail: jinghuizhong@scut.edu.cn).

This article has supplementary downloadable material available at <https://doi.org/10.1109/TITS.2024.3502213>, provided by the authors.

Digital Object Identifier 10.1109/TITS.2024.3502213

planning, traffic simulation, and traffic monitoring among others [2]. However, collected trajectories are often noisy and incomplete due to various reasons such as data transmission issues, positioning error, limited area coverage, and object detection error [3], [4]. Therefore, it is crucial to reconstruct these trajectories systematically before they can be used for different applications. In literature, the notion of “*trajectory*” is viewed differently across different works. It is loosely referred to a wide variety of traffic data ranging from vehicle count, vehicle position, travel time to vehicle path, and others. Generally, trajectory data can be divided into two types: 1) coarse-grained macroscopic data that contain aggregated traffic states with a typical interval of 1-5 min (or more), and 2) fine-grained microscopic data that contain vehicle position information with the interval of 0.1-1 sec. Then, “*trajectory reconstruction*” can be defined as the process of reconstructing (i.e., predicting, estimating, or smoothing) one or more partially or fully incomplete states of interest at each sampling time between a pair of initial and final states.

The motivation behind this paper stems from two factors as follows. Firstly, in China, internet giant companies such as Alibaba, Baidu, and Tencent are leading the AI and big data innovation in the smart mobility industry [5]. Our research work has been developed to tackle the real-world challenge encountered by the Alibaba City Brain Lab. The objective of our collaboration is to perform *fine-grained trajectory reconstruction (FTR)* on incomplete trajectory data collected in a study area involving two signalized intersections in Kunming, China. The data suffer from both irregular vehicle appearance/disappearance patterns, and the relatively large spatiotemporal gap at the microscopic level. Performing FTR for incomplete vehicle trajectory data is attractive for many reasons such as improved traffic simulation modeling and analysis [6], [7], better policy and infrastructure planning [8], and promising potential in autonomous vehicle technology development [9] (elaborated further in Section V). Secondly, despite these benefits of FTR, about one-third of the existing works (See Section II-A) found in the literature only focus on coarse-grained reconstruction neglecting the influence of traffic rules and interactions among vehicles. As a result, they do not apply to our objective, which is to reconstruct vehicle trajectories at the microscopic level. Existing approaches that are suitable for this task are based on: 1) pure microscopic traffic simulation (MTS) that considers traffic rules and vehicle interactions [6], [7], [10], and 2) a combination of both MTS and learning-based models [11].

In the existing works using pure MTS, none of them considers the calibration of simulation model parameters based on incomplete trajectories. This can degrade the accuracy of MTS-based trajectory reconstruction. Furthermore, it also leads to the following technical challenge: “*How to quantify the goodness-of-fit based on incomplete (ground-truth) trajectories to realize the fine-grained reconstruction?*” Therefore, our work aims to tackle this challenge by proposing an FTR framework involving a novel *trajectory similarity evaluation* method to quantify the goodness-of-fit as described later in Section III-E.

Although the parameter calibration was performed in an existing work [11] using the combination of MTS and reinforcement learning, it is only meant for coarse-grained trajectory reconstruction because only one model parameter (i.e., the speed limit of the vehicles) was calibrated to fill missing vehicle count data in some road segments. Thus, its formulation is not ideal for reconstructing at the microscopic level. This is also reflected in our experiment results described in Section IV-C when adapting their reinforcement learning (RL)-based approach for FTR. This RL-based approach has shown poor performance in tackling the FTR problem under limited computing budget. This leads to the following technical challenge: “*How to design an effective optimization scheme to maximize the trajectory similarity between simulated trajectories and incomplete trajectories under limited computing budget?*”

Due to the shortcomings of the afore-mentioned existing works, it is desirable to develop an effective approach where trajectory reconstruction and MTS model calibration can be jointly performed based on fragmented trajectories across different intersections. Such an approach should also aim to maximize the usage of a limited computing budget (in terms of complete simulation runs) while achieving the best possible reconstruction accuracy measured in terms of trajectory similarity. To this end, a new FTR framework using MTS is proposed in this paper. Furthermore, to achieve accurate reconstruction performance with the limited computing budget, *dynamic data-driven evolutionary optimization (D3EO)* is employed to search near-optimal parameters that can help generate the simulated trajectories that are closely similar to the actual ones.

Our main contributions are:

- 1) A new MTS-based FTR framework involving a novel *trajectory similarity evaluation* method is proposed for performing trajectory reconstruction and parameter calibration jointly.
- 2) An innovative D3EO technique is proposed to achieve consistent and accurate trajectory reconstruction results with a limited computing budget.
- 3) An extensive set of controlled and actual reconstruction experiments was conducted by using two real-world urban intersection datasets to show the effectiveness of the proposed framework and the optimization technique.

The rest of the paper is organized as follows: related works are reviewed in Section II. Section III discusses the nature of the problem, and presents our proposed framework used

to address the problem. The experiment results are analyzed in Section IV. Finally, the paper concludes by discussing the implications of our research work and suggesting potential research directions in Section VI.

## II. RELATED WORKS

### A. Vehicle Trajectory Reconstruction

Existing literature on vehicle trajectory reconstruction is listed in Table I. In the table, about one-third of the works only focus on coarse-grained reconstruction, and hence, they do not apply to our objective of FTR.

Smoothing approaches were used for estimating both macroscopic traffic states [12] (such as vehicle count and average speed) and microscopic information [13], [14] (such as vehicle position). Although they can tackle noisy data issues relatively well in the case of macroscopic data, there are two limitations when reconstructing microscopic trajectories. First, their accuracy decreases as the size of the undetected (or occluded) area enlarges due to the increase in terms of uncertainty [13]. The second issue is their incapability of handling traffic situations involving urban intersections [10].

Probabilistic models were used in [15] and [16] to estimate vehicle trajectories collected by sparse GPS data in a freeway scenario [16] and an urban intersection scenario [15]. These models attempted to reconstruct vehicle trajectories by predicting in which driving modes (e.g., idle, acceleration, deceleration, cruising) a vehicle of interest was currently traveling during the missing period. Then, maximum likelihood estimation was used to predict the most likely driving mode sequence which could be used for vehicle trajectory reconstruction during each sampling interval. The limitation of such probabilistic models is that they do not consider vehicle interactions, or traffic rules, which are important for FTR. This could be due to the nature of GPS-based vehicle trajectories that are often independently reported.

Particle Filtering (PF) was also adopted for vehicle trajectory reconstruction for coarse-grained data (e.g., vehicle path) [17] and fine-grained data (e.g., vehicle position) [18]. The target issue addressed by [17] was to reconstruct the vehicle path under low Automatic Vehicle Identification (AVI) coverage. Their path reconstruction accuracy exceeded 80% when the AVI coverage was between 40-50% under the study area set up in the simulator VISSIM [23]. However, it did not utilize real-world data in their experiments. Closer to our objective, FTR using PF was proposed in [18]. However, their approach bypassed microscopic simulation models for reconstruction, thus ignoring detailed vehicle interactions and traffic rules governing these vehicles. Consequently, it is not applicable to oversaturated traffic due to its strong assumption that a certain vehicle could be only stopped by a traffic light near an intersection, but not by other causes.

With the rise of big data, more researchers have adopted deep learning-based methods to harness the power of data abundance and fast computing. Hence, they have become increasingly popular for trajectory reconstruction in terms of GPS data [4], [21] and vehicle path data [22]. In [4], an encoder-decoder network based on Gated Recurrent Unit

TABLE I  
SUMMARY OF EXISTING WORKS ON VEHICLE TRAJECTORY RECONSTRUCTION

Method	Year	Ref	Scenario	Data Source	Data Type	Target Data	Target Issues	Target Resolution	Controlled Experiment Settings
(1) Smoothing	2018	[12]	Expressway	Microwave Sensor, LPR Camera	Real-world	Vehicle Count, Average Speed	Incomplete Traffic States	5 min	Undetected Zone Range: 7.7-9.2 km
	2020	[13]	Instrumented Vehicles (IVs), Freeway	GPS, Video Camera	Naturalistic, Real world	Vehicle Position	Noisy, Occlusions	1 sec	Occlusion Length: 0-10 m
	2021	[14]	IVs	GPS	Real-world	Vehicle Position	Noisy, Sporadic Sampling	-	-
(2) Probabilistic Models	2014	[15]	Signalized Intersections	Video Camera / GPS	Real-world	Vehicle Position, Travel Time	Low Sampling Frequency	1 sec	Sampling Interval: 20 sec
	2016	[16]	Freeway	Video Camera / GPS	Real-world	Vehicle Position, Speed	Low Sampling Frequency	1 sec	Sampling Interval: 10 sec
(3) Particle Filters	2015	[17]	Traffic Network Level	Virtual AVI Detectors	Simulated	Vehicle Path	Low AVI Coverage	-	AVI Coverage: 40-50%
	2020	[18]	Signalized Intersections	Video Camera / Probe Vehicles	Real-world	Vehicle Position, Travel Time	Incomplete Vehicle Trajectories	1 sec	Undetected Duration: 9-69 sec Undetected Length: 122-365 m
(4) Microscopic Traffic Simulation	2018	[10]	Signalized Intersections	Virtual Detectors	Simulated	Vehicle Count, Travel Time	Noisy, Miss-and over-counts	1 sec	Sampling Interval: 3-9 min
	2020	[6]	Freeway	Connected & Automated Vehicles (CAVs)	Simulated	Vehicle Position	Unobserved Vehicle Trajectories	1 sec	Penetration Rate: 1-12%
	2022	[7]	Signalized Intersections	Video Camera / CAVs	Real-world	Vehicle Count, Vehicle Position	Unobserved Vehicle Trajectories	1 sec	Penetration Rate: 5-15%
(5) Variational Traffic Theory	2012	[19]	Signalized Intersections	AVI Detectors, Probe Vehicles	Real-world	Vehicle Position, Travel Time	Unobserved Vehicle Trajectories	1 sec	Sampling Interval: 1-5 min
	2021	[20]	Signalized Intersections	Video Camera / GPS	Real-world	Queuing Related Measures	Unobserved Vehicle Trajectories	1 sec	Sampling Interval: 3 sec Penetration Rate: 5-20%
(6) Learning-based Models	2019	[21]	Taxi-logged Trajectories	GPS	Real-world	Vehicle Position	Low Sampling Frequency	1-5 min	Sampling Rate: 30-70%
	2020	[4]	User-logged Trajectories	GPS	Real-world	Vehicle Position	Sporadic Sampling	5 sec	-
	2021	[22]	Traffic Network Level	Traffic Camera Images	Real-world	Vehicle Path	Incomplete Vehicle Trajectories	-	-
(4) + (6)	2019	[11]	City Level	GPS, Surveillance System	Real-world	Vehicle Count	Incomplete Vehicle Trajectories	1 min	-

In the *Data Source* column, *Video Camera / GPS*, for instance, indicates that the data were collected by *Video Camera*, but it was assumed to come from *GPS* for experiments.

(GRU) has been shown to outperform smoothing techniques such as Autoregressive Integrated Moving Average (ARIMA) and filtering techniques such as Kalman Filter. More interestingly, a hybrid model was proposed in [21] by combining an LSTM-based sequence-to-sequence model and Kalman Filter. By using GPS trajectory datasets, the proposed model has been shown to surpass classic search algorithms and other deep learning methods. Besides GPS trajectory reconstruction, another work that is based on image data has been proposed in [22] for large-scale vehicle path (trajectory) reconstruction. It proposed a vehicle path reconstruction system involving multiple deep learning models. An extensive set of experiments was carried out based on a real-world dataset that consists of more than 7 million vehicle snapshots captured by over a thousand traffic cameras.

The trajectory reconstruction problem is also tackled by existing works that rely more on domain knowledge in traffic engineering and transportation research. They can be divided into two reconstruction approaches that are fundamentally based on – 1) macroscopic traffic flow theory [19], [20], and 2) microscopic traffic flow theory [6], [7], [10]. The former is based on the Lighthill-Whitham-Richards (LWR)

model [24], and its variational formulation [25], [26] as its efficient solution method. The key idea behind this variational theory (VT) approach for trajectory reconstruction is that given a discretized time-space diagram (or coordinate system), the nodes having the same cumulative number of vehicles on this diagram can be connected to reconstruct vehicle trajectories that have appeared within the boundaries of the time-space diagram [20]. Although VT-based reconstruction is likely to work for macroscopic trajectory data (e.g., vehicle count), it is not realistic from the microscopic standpoint due to two issues. First, its reconstructed trajectories follow a piecewise linear pattern among connected nodes [6], meaning there is no acceleration/deceleration during that interval, which is often not the case in practice. Second, it disregards lane-changing behaviors during the reconstruction process as constrained by the VT formulation [20].

The only reconstruction approaches that jointly consider traffic rules and vehicle interactions during the reconstruction process are those that are based on microscopic traffic simulation (MTS) [6], [7], [10], and a combination of MTS and learning-based methods [11]. A generic data assimilation framework, that involves a microscopic traffic flow model with

PF was proposed in [10]. A car-following model was used to capture vehicle dynamics while PF was adopted to filter the noisy observation data and estimate vehicle states (such as vehicle count and travel time). The framework was validated based on simulated data generated under a signalized urban road setting. However, only longitudinal vehicle movement was considered in their simulation for simplicity. Similarly, although cut-in and cut-out behaviors were considered in [7], their proposed method was not evaluated under multi-lane scenarios. Likewise, in [6], their test environment was also simplistic since it assumed all the vehicles were traveling on a single-lane road only.

Another common issue with existing MTS-based approaches is that no simulation model (e.g., car-following model) parameter calibration was performed in their reconstruction processes. This can indeed undermine the accuracy of MTS-based trajectory reconstruction. This issue is addressed in [11] by using deep reinforcement learning (DRL) to calibrate the speed limit of different vehicle types and reconstruct missing vehicle volume data along the process. Since this is still a coarse-grained reconstruction targeting vehicle count, its formulation is not ideal for reconstructing at the microscopic level. It is also preferable to perform FTR under the constraint of a limited computing budget. Therefore, this serves as the motivation behind our paper to develop a new MTS-based reconstruction framework that meets those requirements.

### B. Evolutionary Optimization (EO)

Many real-world optimization problems not only involve non-convex or multi-modal optimization landscapes filled with many local optima but also contain a large number of decision variables accompanied by a high number of constraints [27]. As a result, conventional approaches such as mathematical optimization (e.g., linear or quadratic programming) or single solution-based local search algorithms (e.g., simulated annealing, tabu search) struggle to find optimal or near-optimal solutions in these problems [27], [28], [29], [30]. Evolutionary optimization (EO) approaches have emerged as effective tools to handle such problems. These EO algorithms are population-based metaheuristic algorithms (e.g., genetic algorithms, particle swarm optimization) that perform a global search by employing many candidate solutions sampled from the solution space. Compared with local search methods, the EO algorithms have relatively strong global search ability by maintaining diverse solutions in the population [30].

When the evaluation of objective function(s) and constraint(s) used in EO is based on data gathered from physical or numerical simulation experiments, such an EO is known as *data-driven evolutionary optimization* [27]. Although data-driven EO can handle optimization problems where the exact evaluation of the objective function is computationally inexpensive, it becomes inefficient when the evaluation takes a considerably longer duration. In fact, a single-function evaluation involving a high-resolution MTS model can take from a few seconds to hours. Therefore, it is desirable to adopt a computationally efficient mechanism that could best exploit

the limited number of function evaluations. For this purpose, *surrogated-assisted evolutionary optimization (SAEO)* is developed in the literature. In SAEO, surrogates are function approximators approximating expensive objective/constraint functions and are often pre-trained using offline training data (e.g., an input-output pair containing a solution as the input and its corresponding fitness as the label) before the start of the optimization process.

SAEO can be further divided into 1) *offline data-driven SAEO*, and 2) *online data-driven SAEO*; the terms “*online data-driven*” and “*dynamic data-driven*” are used interchangeably in this paper. The former allows an EO algorithm to completely replace exact function evaluations with the surrogates’ approximations while the latter still performs exact function evaluation for some new promising candidates per an arbitrary number of generations. As a result, no new training data for surrogates are generated in the former, but in the latter, online training data are generated from exact function evaluations. These online data can be used to update the surrogates’ parameters dynamically during the search process so that their approximation accuracy can be further improved. Because the offline variants are vulnerable to large approximation errors that could mislead the evolutionary search, online data-driven SAEO algorithms are preferable in most cases. Therefore, many new variants [31], [32], [33] of online data-driven SAEO have been developed in the EO community with successful applications in optimal water drainage system design [28], production optimization [34] as well as Intelligent Transportation Systems in smart cities [29].

Despite its success, to the best of our knowledge, none of the trajectory reconstruction works has adopted online SAEO in tackling the problem of FTR. Furthermore, existing SAEO methods could not be directly applied to FTR because these works focus more on saving computationally expensive function evaluation runs than ensuring the quality of the solutions found via *surrogate-assisted evolutionary sampling (SAES)*. As for FTR, it is desirable to place emphasis on both aspects to achieve high-quality reconstruction results. One viable approach is to execute both standard *evolutionary sampling (ES)* and SAES independently in parallel during the search. In this case, the former can serve as the baseline for producing reference candidates while the latter aims to find better promising solutions than these reference ones. This can ensure that new candidates found via SAES during the search process are at least as good as those of the standard ES. Thus, such an SAEO is proposed in this paper in order to achieve accurate and consistent trajectory reconstruction results under the limited computing budget.

## III. FINE-GRAINED TRAJECTORY RECONSTRUCTION

### A. Problem Formulation

1) *Trajectory Definition*: In this paper, a vehicle trajectory is defined with reference to [2] as:

$$P = \{p_1, p_2, \dots, p_l\}; \quad p_t \in \mathbb{R}^d \quad (1)$$

where  $P$  is the entire observed trajectory with length  $l$ ;  $p_t$  is the observed vehicle state represented as a vector with

dimension  $d$  at sampled time  $t$ ; when only vehicle position is concerned,  $p_t$  usually contains the 2D position of the vehicle. This 2D convention is adopted in this paper. The sampling interval  $\Delta t$  is chosen as 1 sec.

If a trajectory contains missing points between its start  $p_1$  and end  $p_l$ , then  $p_t$  (where  $t \neq 1, t \neq l$ ) is subject to the condition whether the vehicle position is either captured or not by an arbitrary sensor at the sampled time  $t$ . Generally, there are two types of sensors considered in the literature: 1) fixed sensor (e.g., radar or video camera attached to a fixed location), and 2) mobile sensor (e.g., probe vehicles, connected and automated vehicles (CAVs) [6]). In this paper, we are only concerned with the former.

Trajectory data collected by fixed sensors are rarely independent. Often, there is a high correlation among these trajectories due to vehicle interactions, especially when they are traveling within close proximity. Thus, it is more natural to study these trajectories jointly as follows.

$$\zeta = \{P_1, P_2, \dots, P_V\} \quad (2)$$

where  $\zeta$  is a collection of trajectories that belong to a total of  $V$  vehicles, observed during an arbitrary study period. Each observed trajectory  $P_j$  may have different lengths  $l_j$ , and contain one or more missing data points at different sampled time steps.

2) *Major Assumptions:* Reconstruction of a given trajectory is possible if and only if:

- 1) Trajectory length is at least two ( $l_j \geq 2$ ).
- 2) Trajectory data points at the *origin* (the first time the position of a vehicle of interest is detected) and *destination* (the last time its position is detected) are known.

3) *Reconstruction as Optimization:* Given a microscopic traffic simulation (MTS) model, the trajectory reconstruction can be formulated as an optimization problem as follows.

$$\max_{\lambda \in [\lambda_{LB}, \lambda_{UB}]} J(\zeta, \hat{\zeta}) \quad (3)$$

$$J(\zeta, \hat{\zeta}) \approx \max_{i \in \{1, 2, \dots, r\}} J(\zeta, \hat{\zeta}_i) \quad (4)$$

$$\hat{\zeta} = MTS(\zeta | \lambda) \quad (5)$$

$$\text{s.t. } rN_{sim} \leq B \quad (6)$$

Given partially observed trajectories  $\zeta$  that contain missing data points collected during a study period, the objective is to maximize the similarity between the observed  $\zeta$  and simulated  $\hat{\zeta}$  trajectories, measured with a similarity scoring function  $J$  by searching the optimal simulation parameters  $\lambda^*$  subject to their valid ranges  $[\lambda_{LB}, \lambda_{UB}]$ , and available simulation budget  $B$ . This is an MTS-based stochastic optimization problem with various types of randomness present in the simulation (e.g., vehicle initialization, route choices, etc.). As a result, a simulation run configured with the same set of parameters may generate different trajectories each time the simulation is executed. Thus, with reference to [35], [36], the maximum similarity score resulting from  $r$  replication runs is selected as the estimated score of a given  $\lambda$ . Accordingly, the total number of simulation runs ( $rN_{sim}$ ) consumed for solving the optimization problem cannot exceed  $B$ ; where  $N_{sim}$  is the number of simulation runs executed with unique  $\lambda$  values.

Intuitively, the goal is to reconstruct vehicle trajectories by simulating them together based on their origin and destination points throughout the study period. In doing so, any missing points of a trajectory  $P_j$  are reconstructed along the process. Since the simulated vehicles not only interact with each other during simulation run-time but also react to different traffic rules (governed by traffic light signals or lane markings), the simulated  $\hat{\zeta}$  can also deviate from the partially observed  $\zeta$ . In other words, although the former helps reconstruct missing data points, it is also prone to deviation from the already observed data points that are not missing. Hence, the objective of the optimization is to reduce this deviation by maximizing the similarity between  $\zeta$  and  $\hat{\zeta}$  with  $\lambda^*$ .

The above formulation addresses two interrelated problems jointly which are fine-grained trajectory reconstruction with simulation-based optimization, and simulation model parameter calibration with partially observed trajectories. Hence, two major outcomes can be achieved after solving the problem: 1) partially observed trajectories are reconstructed with maximal accuracy, and 2) the simulation parameters are calibrated near-optimally with the given data, both of which can be useful for different purposes. However, solving this optimization problem is indeed difficult because the quality of its solution depends on many factors such as simulation randomness, number of replication runs, simulation budget, and the scoring function which should properly quantify the similarity between  $\zeta$  and  $\hat{\zeta}$ . This is exacerbated by challenging missing patterns altogether affecting  $\zeta$  rendering the reconstruction task even more difficult.

### B. Challenging Real-World Missing Patterns

In this section, different missing patterns that arise in the real-world data encountered by the Alibaba City Brain Lab are discussed. These patterns are indeed challenging from the perspective of microscopic trajectory reconstruction and are illustrated in Figure 1.

The figure shows a study area involving two signalized intersections with radars and traffic cameras deployed as fixed sensors. In this study area, the vehicles are traveling from the first junction (top) to the second (bottom). The total number of vehicles involved in each scenario is displayed in the *Total*. Trajectories are represented by their respective arrows with their respective colors. When a vehicle is detected by the sensor during the sampling interval, it is represented with a solid color. Otherwise, the color is not filled implying that it is missing during the sampling interval. When the trajectories belong to unknown vehicles whose IDs were failed to associate by the sensors, they are represented with a solid grey color. These unknown vehicles also suffer from missing data points, and hence, their colors are not filled in some steps. No trajectory arrow is used to represent unknown vehicles simply because the data points generated by them across two junctions could not be associated properly. The grey box with the *question mark* (?) indicates that there is no sensor deployment in that area and hence, all vehicle trajectory data are totally missing when they pass through this region. For the rest of the paper, this will be referred to as *hidden zone*.

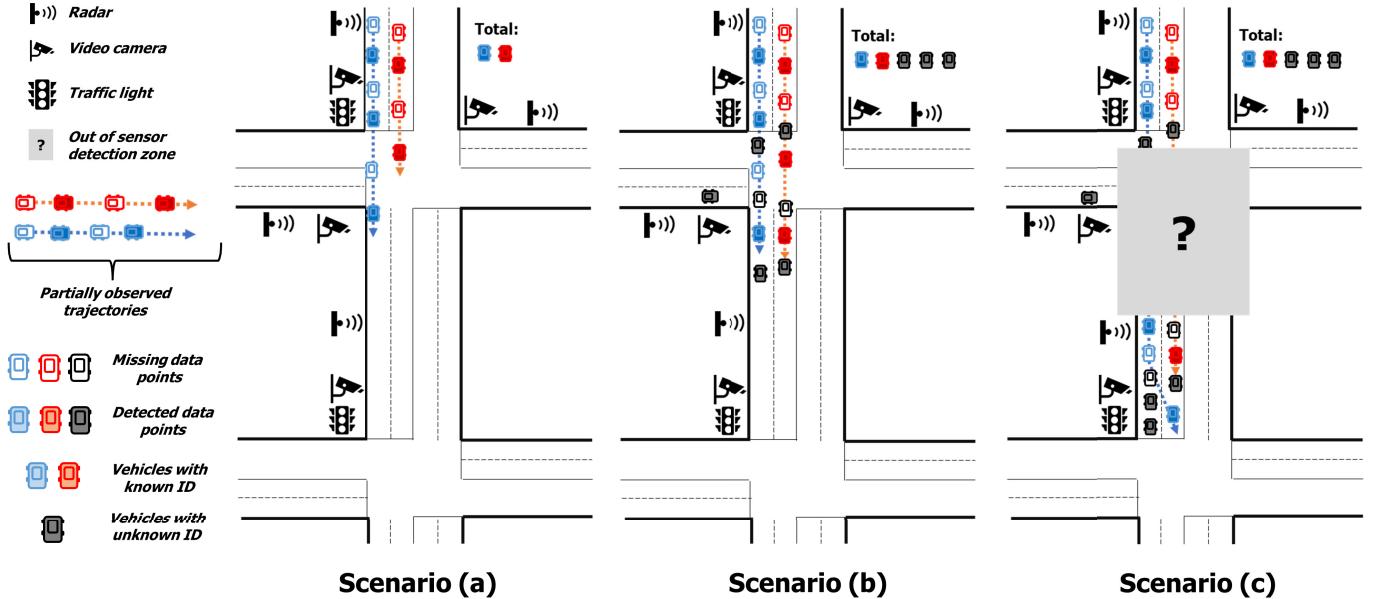


Fig. 1. Different missing patterns found in microscopic vehicle trajectory data – Scenario (a): Sporadic Sampling; (b) Co-existence of unknown vehicles with issue from (a); and (c) Large spatiotemporal gap with issues from both (a) and (b).

1) *Sporadic Sampling*: This missing pattern corresponds to the Scenario(a) in Figure 1. This is similar to sparsely sampled data issues as with GPS-based trajectories. This is the least difficult scenario among all since the data points are only missing at certain sampling intervals. It could even be reconstructed with interpolation and smoothing approaches if the sampled data points of a trajectory do not constitute a long consecutive missing period. If they undergo such a period, the interpolated/smoothed data points are often not realistic anymore since the interpolated trajectories can turn out to be overlapping each other (can be verified in the time-space diagram), falsely implying that vehicles crashed into each other during this period. Likewise, deep learning-based approaches can be employed to improve the reconstruction accuracy, but it is still questionable whether the deep learning-based reconstruction will also result in overlapped trajectories.

2) *Co-Existence of Unknown Vehicles*: In this scenario shown in Figure 1(b), vehicle trajectories not only suffer from sporadic sampling but are also surrounded by additional uncertainty introduced by unknown vehicles  $V_?$ . These  $V_?$  cannot be associated with a proper ID due to a number of reasons. First, the vehicle was detected by radar only and no video camera managed to capture its identity. In such a case, when they re-appeared again in another area of the road segment upon detection by another radar, it is uncertain whether the vehicle was a new vehicle entering from other directions, or it was the same vehicle that had been detected by the previous radar. Furthermore, when these  $V_?$  trajectories are also subject to sporadic sampling, it makes the task more challenging because there could be hidden interactions among vehicles during undetected sampling intervals. This scenario is similar to the type of reconstruction problem tackled by connected and automated vehicles (CAVs) with MTS [6], [7]. These approaches require that reliable CAVs' trajectories are available throughout the study period so that the surrounding

traffic near CAVs can be reconstructed. This assumption is inapplicable from the perspective of the trajectory data collected by fixed sensors since their coverage area is limited to only certain areas of the road due to sparse sensor deployment. It is also highly cost-intensive to install these fixed sensors in all areas of the road network. This leads to a large spatiotemporal gap in the trajectory data from the microscopic view.

3) *Large Spatiotemporal Gap*: This scenario belongs to Figure 1(c), and it is the most difficult among all for – 1) it suffers from both of the first two missing patterns and 2) lack of sensor deployment in a certain stretch of the study area between the two consecutive intersections which introduces the problem of completely missing trajectory data. Based on the size of the hidden zone, all vehicle trajectories are subject to incompleteness for a prolonged period affecting its continuity in terms of spatiotemporality. It also exacerbates the difficulty of associating  $V_?$  by means of other known vehicles because of the large uncertainty introduced by the hidden zone.

### C. Target Reconstruction Issues

There are two groups of vehicles involved in the study area shown in Figure 1: 1) *ground-truth vehicles* ( $V_{truth}$ ) whose route choices are known when they travel across both intersections and 2) *unknown vehicles* ( $V_?$ ) whose route choices are unknown upon entering the road segment linking towards the second intersection. In this paper, we only focus on the reconstruction of  $V_{truth}$  trajectories. However, since  $V_?$  co-exists with  $V_{truth}$ , they cannot be simply neglected (by not simulating them) because it can severely affect the reconstruction accuracy. On one hand, we cannot ignore  $V_?$  since the simulation results will be unrealistic, and inaccurate. On the other hand, we also cannot perform the parameter estimation based on their information since they do not have

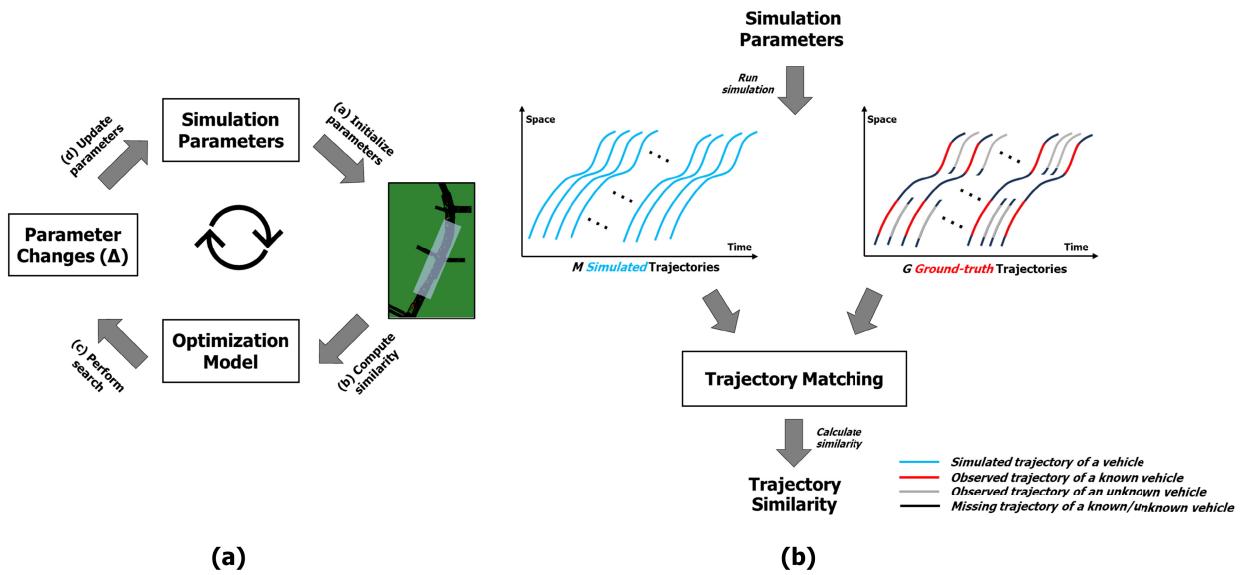


Fig. 2. (a) Trajectory reconstruction with MTS-based optimization; (b) Trajectory similarity evaluation.

proper origin and destination points across the two intersections. Although another set of unknown vehicle trajectories is detected by the sensors deployed in the second intersection, it is very challenging to perform their measurement-to-track association [37] only based on their position information due to the presence of a large hidden zone. Therefore, our proposed framework for FTR will take into account these aspects.

#### D. Proposed Reconstruction Framework

Our FTR framework using MTS-based optimization is shown in Figure 2(a). The framework aims to find the optimal  $\lambda^*$  via an iterative search process for the optimization problem formulated in Section III-A.3. With reference to Figure 2(a), in the beginning, a given MTS model is initialized with a given set of simulation parameters  $\lambda$  in Step (a). At each iteration, simulation runs are executed for  $r$  times using the same  $\lambda$  due to the MTS model's randomness. Then in Step (b), the quality (fitness) of  $\lambda$  is evaluated via the similarity scoring function  $J$  by calculating the similarity between partially observed and simulated trajectories collected after complete simulation runs. Based on this information, a chosen optimization model determines the favorable search direction that will guide towards  $\lambda^*$  in Step (c). Finally, in Step (d), current  $\lambda$  is updated before it continues to the next iteration. This iterative process will repeat until the available simulation budget  $B$  runs out. The best solution found after the search is used as the approximation to the true  $\lambda^*$ . In this framework, two crucial aspects that have a direct impact on the reconstruction accuracy are: 1) evaluation of  $J$  measuring the overall trajectory similarity, and 2) performance of optimization model searching  $\lambda^*$  under  $B$ . Thus, a new method for evaluating  $J$  and an advanced D3EO-based optimization technique are proposed in Sections III-E, and III-F, respectively.

#### E. Trajectory Similarity Evaluation

Since the evaluation of the objection function  $J$  involves running MTS, collecting simulated trajectories, performing trajectory matching, and finally computing overall trajectory similarity, the following sections elaborate on each major part involved in this evaluation process.

1) *Simulation Parameters*: The key components of an MTS model influencing the quality of its simulation outputs are its behavioral models governing the driving behavior of each individual vehicle running in the simulation. Generally, these behavioral models consist of *car-following models* (CFM) [38], [39], *lane-changing models* (LCM) [40], [41], and *intersection models* (IM) [42] as implemented in well-known MTS simulators such as SUMO [43], and VISSIM [23]. Each of these models has its own set of constant parameters that should be calibrated before running the simulation. However, most existing MTS-based reconstruction works [6], [7], [10] only used default car-following parameters reported in the literature with no additional calibration performed. In fact, the performance difference in terms of reconstruction accuracy between the calibrated MTS and default MTS models can be significant as observed in our experiment results (See Section IV-C). Since our framework focuses on the reconstruction of a 2D vehicle position trajectory, it is important to calibrate the above behavioral models jointly in order to achieve the best possible reconstruction accuracy. Thus, in this paper, without loss of generality, the set of simulation parameters  $\lambda$  includes both CFM and LCM parameters except those related to IM (because IM-related parameters have negligible impact on all our experiment scenarios).

2) *Simulation Randomness*: There are two main sources of simulation randomness considered in this paper, which are vehicle initialization randomness, and vehicle route choice randomness. As for vehicle initialization, two strategies are viable – 1) initializing randomly at any location (sufficient for vehicle insertion) on the nearest originated edge (based on

their start time and position), and 2) initializing at their exact start time and position. The former is the common practice in MTS and adopted in this paper because the latter often creates inconsistent simulation dynamics such as conflicts between a newly initialized vehicle and other running vehicles, and unnecessary emergency braking leading to vehicle pileups. As for vehicle route choices, the probability of each origin-destination (OD) route can be inferred from the information based on the available trajectory data such as vehicle count at each lane joining an intersection or the vehicle trajectories with known ID association. Then, based on these probabilities, a vehicle's destination route choice is determined via weighted sampling when they are first initialized in the simulation.

### 3) Trajectory Matching:

$$\min \sum_{i=1}^M \sum_{j=1}^G -c_{ij} x_{ij} \quad (7)$$

$$\text{s.t. } \sum_{j=1}^G x_{ij} = 1, \quad \text{for } i \in \{1, 2, \dots, M\} \quad (8)$$

$$\sum_{i=1}^M x_{ij} \leq 1, \quad \text{for } j \in \{1, 2, \dots, G\} \quad (9)$$

$$x_{ij} \in \{0, 1\}, \quad \text{for } i \in \{1, 2, \dots, M\} \quad (10)$$

$$\text{and } j \in \{1, 2, \dots, G\} \quad (11)$$

Since simulated vehicles are randomly initialized based on the OD route probability, it is required to find the right pair of trajectories to compute the similarity score. Therefore, as seen in Figure 2(b), each target  $V_{truth}$  trajectory needs to be paired with an appropriate simulated  $V_{sim}$  trajectory. This idea of finding a trajectory pair for calibration is inspired by an existing work [44] where trajectory matching was performed based on similar speed profiles, traffic conditions, and other factors. Although such an approach might be suitable for calibrating the simulator to perform microscopic traffic flow analysis, it is not ideal for our fine-grained reconstruction scenario. This is because we aim to reconstruct as much information as possible out of each incomplete ground-truth trajectory, rather than merely attempting to perform the calibration of the simulator. Adapting the practice of [44], the matching criterion is based on the spatiotemporal similarity in terms of start time and position as defined in Equation (12). However, instead of the sub-optimal greedy matching strategy used in [44], a more optimal matching strategy is adopted in this paper by casting this matching problem as a linear unbalanced assignment problem as formulated in Equation (7) and solving it using the Hungarian method [45]. The problem is unbalanced since we have a relatively higher number of  $M$  simulated trajectories than that of  $G$  ground-truth trajectories. To the best of our knowledge, in the domain of trajectory reconstruction, this is the first time in literature formulating the trajectory matching as an unbalanced linear assignment problem and solving it using the Hungarian method.

The optimal trajectory matching can be achieved by minimizing the objective function in Equation (7), that is the sum of matching costs ( $-c_{ij}$ ) multiplied by the assignment variables  $x_{ij}$ . Equation (8) ensures that each  $V_{truth}$  trajectory is assigned

to exactly one  $V_{sim}$  trajectory. Equation (9) allows for the possibility of unassigned simulated trajectories by relaxing the constraint to allow at most one assignment per column. Equations (10) and (11) require that the assignment variables  $x_{ij}$  are binary. The cost  $c_{ij}$  of a pair of trajectories  $i$  and  $j$  is defined as follows:

$$c_{ij} = \kappa(t_{i1}, t_{j1}) + \kappa(p_{i1}, p_{j1}); \quad c_{ij} \in [0, 2] \quad (12)$$

$$\kappa(y, z) = \exp\left(-\frac{\|y - z\|^2}{2\sigma^2}\right); \quad \sigma = 1 \quad (13)$$

where  $t_{i1}, p_{i1}$  and  $t_{j1}, p_{j1}$  are the start time and position of trajectories  $i$  and  $j$ , respectively. Their corresponding similarity is measured with Gaussian kernel  $\kappa(\cdot)$  and  $c_{ij}$  is the sum of spatial and temporal similarity. For ease of assignment, the trajectory matching is performed separately for each individual OD route.

**4) Trajectory Similarity:** Measuring the similarity between a pair of partially observed  $V_{truth}$  and complete  $V_{sim}$  trajectories is non-trivial. As shown in Figure 2(b), the former contains fragmented sub-trajectories with missing points while the latter is fully complete within the study time-space boundary. Among various trajectory distance/similarity measures discussed in [46], *Spatiotemporal Linear Combine Distance (STLC)* [47] is the only similarity measure that could potentially address this issue. Hence, STLC is extended for our task as follows.

$$J = \begin{cases} \hat{J} & ; \quad J \in [0, 2] \\ 0 & \text{if } G > M \end{cases} \quad (14)$$

$$\hat{J} = \frac{1}{|OD|} \sum_{i=1}^{|OD|} \left[ \frac{1}{|V_i|} \sum_{j=1}^{|V_i|} K_S(P_{ij}, P_{ij}^{sim}) + K_T(T_{ij}, T_{ij}^{sim}) \right] \quad (15)$$

$$K_S(P, P_{sim})$$

$$= \frac{1}{|P|} \sum_{s=1}^{|P|} \kappa(p_s, \hat{p}_t); \quad p_s \in P; \quad \hat{p}_t \in P_{sim}$$

$$t \approx \underset{t_{sim} \in T_{sim}}{\operatorname{argmin}} \|s - t_{sim}\|^2 \quad (16)$$

$$K_T(T, T^{sim})$$

$$= \frac{1}{|T|} \sum_{s=1}^{|T|} \kappa(t_s, \hat{t}_p); \quad t_s \in T; \quad \hat{t}_p \in T_{sim}$$

$$p \approx \underset{p_{sim} \in P_{sim}}{\operatorname{argmin}} \|p - p_{sim}\|^2 \quad (17)$$

Equation (14) defines the overall trajectory similarity measure  $J$  with its range  $[0, 2]$ . The measure outputs its minimum value 0 if  $G$  is larger than  $M$ ; where  $G$  and  $M$  are the numbers of target ground-truth and simulated trajectories, respectively. In other words, if there is an insufficient number of simulated trajectories for pairing with every target ground-truth trajectory, then such a simulation run is considered invalid and assigned the lowest similarity score. Otherwise, the proper similarity score  $\hat{J}$  can be calculated as defined in Equation (15).

Equation (15) defines  $\hat{J}$  as the mean similarity measure taking the average over the similarity scores obtained in each

OD route. For each OD route  $i$ , its similarity score is defined as the average spatiotemporal similarity score of all target ground-truth vehicles  $V_i$  within its route. The spatiotemporal similarity between a target vehicle and its paired simulated trajectory is defined as the sum of spatial and temporal similarity given by  $K_S(\cdot)$  and  $K_T(\cdot)$ , respectively.

Equation (16) defines  $K_S(\cdot)$  taking the observed sampled positions  $P$  of a target vehicle and its simulated counterpart  $P_{sim}$  as its inputs. Since the simulated trajectory is a complete trajectory, there are more sampled positions present in  $P_{sim}$  than in  $P$ . Thus, each sampled position  $p_s$  in  $P$  is required to pair with a relevant candidate position  $\hat{p}_t$  in  $P_{sim}$ . The candidate is found by finding the simulated position that has the closest sampled time  $t$  (i.e., the minimum squared distance between the observed time  $s$  and simulated time  $t_{sim}$ ). Then, the spatial similarity between  $p_s$  and  $\hat{p}_t$  is calculated using the same  $\kappa(\cdot)$  defined in Equation (13). Finally,  $K_S(\cdot)$  outputs the mean of the spatial similarity of all  $p_s$ .

Equation (17) can be explained similarly for measuring the mean temporal similarity as the above Equation (16).  $T$  and  $T_{sim}$  are the observed and simulated sampled time steps, respectively.  $t_s$  and  $\hat{t}_p$  are the observed sampled time and its paired candidate, respectively.  $p$  and  $p_{sim}$  are the observed and simulated positions, respectively.

#### F. Dynamic Data-Driven Evolutionary Optimization (D3EO)

Our proposed D3EO model is illustrated in Figure 3, and the corresponding algorithm is described in Algorithm 1. It starts with a pre-selected population of  $N$  candidate solutions and their corresponding fitness. This initial population goes through two evolutionary sampling branches involving the usual standard evolution, and our proposed *surrogate-assisted n-step evolution (SAnE)* (which are labeled as (2) and (3) in the figure, respectively). The goal of these evolutionary sampling strategies is to search for new promising candidate solutions for real fitness evaluation via either evolutionary operators or surrogate-assisted search methods [48]. As for the standard evolution, new candidate solutions are sampled based on a chosen EO algorithm (See Section III-F2), and after that, they are evaluated with the proposed MTS-based trajectory similarity evaluation procedure. As for the SAnE, it is designed to accelerate the EO by evolving the current population for  $n$  steps ahead using the same EO algorithm assisted with a surrogate model. This can make the overall EO more effective for finding the optimal solution under a limited computing budget. During this accelerated search, the evaluation of the solutions' fitness is based on the approximations of the surrogate. Due to the probabilistic nature of the EO algorithm, the same starting population may result in different evolved populations after  $n$  steps. Thus, SAnE is executed for  $S$  number of times, then the top-N solutions are selected based on the surrogate-approximated fitness. This complete SAnE process is implemented in Algorithm 3. As the solutions sampled via SAnE are prone to approximation error, they are also evaluated with the proposed evaluation method in order to ensure the quality of these solutions and avoid any issues arising from the approximation error. At this stage, since new

---

#### Algorithm 1 D3EO Algorithm

---

```

1 Input: LHS solution samples of size  $N_{LHS}$ ,  
pre-trained surrogate ANN;  
2 Output: The best solution found  $\hat{\lambda}^*$ ;  
3 Initialization:  
4 Set maximum evaluation budget  $B$ , number of  
replication runs  $r$ ;  
5 Set current budget spent  $b$  to  $rN_{LHS}$ ;  
6 Initialize population  $\Lambda$  and fitness  $F$  with top-N  
solutions from  $N_{LHS}$  samples; where  
 $\Lambda = \{\lambda_1, \dots, \lambda_N\}$ ,  $F = \{f_1, \dots, f_N\}$ ;  
7 while  $b \leq B$  do  
8    $\Lambda_E \leftarrow \text{GA}(\Lambda, F)$ ; where  $\Lambda_E = \{\lambda_1, \dots, \lambda_N\}_E$ ;  
9    $\Lambda_{SAnE} \leftarrow \text{SAnE}(\Lambda, F, \text{ANN})$ ; where  
 $\Lambda_{SAnE} = \{\lambda_1, \dots, \lambda_N\}_{SAnE}$ ;  
10  Evaluate  $F_E$  and  $F_{SAnE}$  using MTS-based  
trajectory similarity evaluation;  
11  Update ANN parameters with incremental batch  
learning (IBL) using online samples  $D$ ; where  $D$   
=  $\{(\Lambda_E, F_E), (\Lambda_{SAnE}, F_{SAnE})\}$ ;  
12  Update  $\Lambda, F$  with top-N solutions from  $D$ ;  
13  Replace worst-k solutions in  $\Lambda$  with k-elite  
solutions;  
14  Increment  $b$  by  $r|D|$ ;  
15 end

```

---

training samples become available, they are used to update the parameters of the surrogate model so that it does not suffer from increasing approximation errors as it progresses. Among the solutions found (via both sampling strategies) in the current generation, only top-N solutions are selected for the next generation based on their actual fitness. Through the practice of elitism, the k-worst solutions in the population are replaced with the k-elite solutions (found up to the current generation) before proceeding to the next generation. The elitism ensures that k-fittest solutions are always propagated to the next generation [30] so that they can contribute to the evolution process ( $k=1$  in this paper). This optimization process repeats until it has consumed the available simulation budget. Finally, the best solution  $\hat{\lambda}^*$  found after the optimization is used as the approximation to the true optimal solution  $\lambda^*$ . Accordingly, the simulated trajectories generated via the MTS initialized with  $\hat{\lambda}^*$  are used for reconstructing target ground-truth trajectories.

The above-proposed model consists of four major elements – 1) initial population, 2) evolutionary algorithm, 3) surrogate for SAnE, and 4) surrogate update. They are labeled accordingly in Figure 3 and are elaborated below.

1) *Initial Population*: The diversity of solutions in the initial population has a considerable impact on the performance of an EO model [49], [50]. To this end, *Latin Hypercube Sampling (LHS)* method [51] is often used to generate initial solution samples that are well distributed over the solution space [52]. The LHS usually helps EO models produce better optimization performance than uniform random sampling [49], [50]. Thus, in this paper,  $N_{LHS}$  samples are first generated via the LHS method, and their fitness are evaluated. Then, top-N solutions

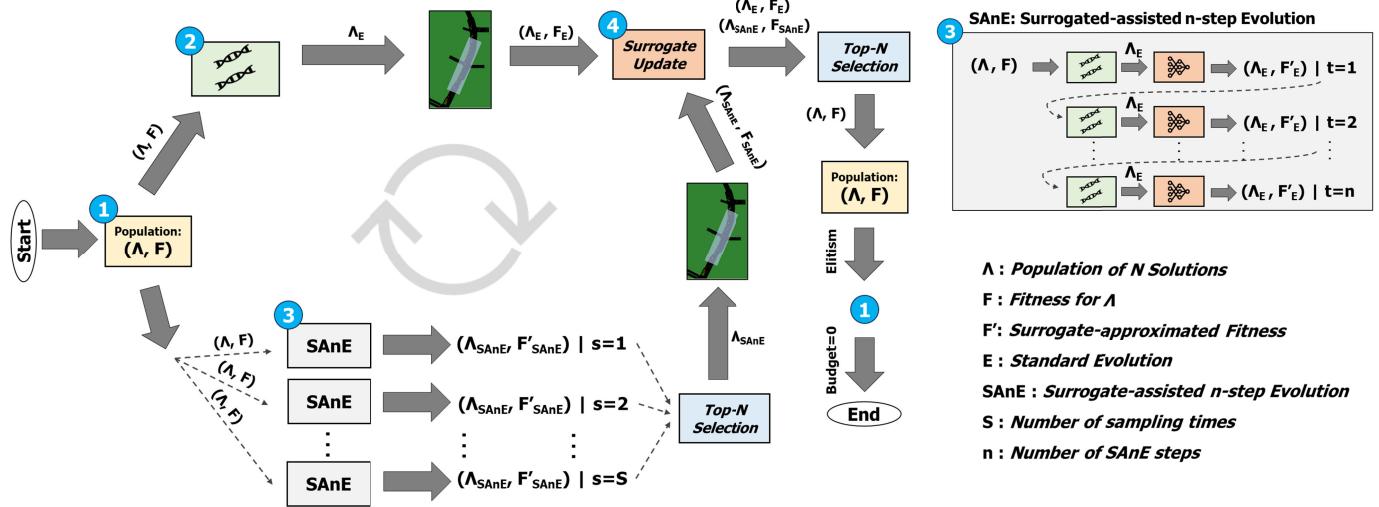


Fig. 3. Dynamic data-driven evolutionary optimization with surrogate-assisted n-step evolution.

(where  $N < N_{LHS}$ ) are chosen as the initial population of our model.

**2) Evolutionary Algorithm:** Among EO algorithms, the biological evolution-inspired Genetic Algorithm (GA) [30] is very well-known for calibrating microscopic traffic simulation models since it does not require any restrictive assumptions in the formulation of the underlying optimization problem. For its ease of parallelism and flexibility, it has become a standard calibration method widely used in many CFM calibration works [53], [54], [55]. Therefore, the standard GA is adopted as our core evolutionary search without loss of generality. It is expected that the proposed method could also be applied using other evolutionary search approaches such as *Particle Swarm Optimization* or *Differential Evolution* [27].

In this paper, the GA is implemented as shown in Algorithm 2. The algorithm first shuffles the solutions in the population to avoid any ordering bias affecting the subsequent GA operations. Then, new candidate solutions are generated by going through the following genetic operations in order – 1) selection, 2) crossover, and 3) mutation. The selection operation aims to select the parent solutions with good fitness scores so that more new high-quality solutions can be sampled based on their information. The crossover operation combines these good solutions (two or more parents) to generate potentially better solutions. The mutation operation helps create diverse solutions that are different from the current population. The selection operation favors exploitation while crossover and mutation lean towards exploration.

As recommended in [30], a *2-way tournament selection with replacement* is adopted here since it can help maintain population diversity when a small tournament size is used. As for the crossover operation, *1-point crossover* is adopted for its ease of implementation [30], and finally, the uniform random noise-based mutation is adopted to encourage the algorithm's exploration. Since after adding random noises, some solutions may result in an invalid solution range, such solutions are clipped to be within the range, after which the algorithm returns them as the new solution samples.

## Algorithm 2 Genetic Algorithm

- 1 **Input:** Population  $\Lambda$  and its fitness  $F$ ;
- 2 **Output:** New solution samples  $\Lambda_E$ ;
- 3 **Parameters:**
- 4 Number of parents  $N_{pr}$ , number of offsprings  $N_{op}$ , crossover rate  $R_{cx}$ , mutation rate  $R_m$ , valid solution range  $[\lambda_{LB}, \lambda_{UB}]$ ,  $\Delta\lambda = \gamma \left[ \frac{\lambda_{UB} - \lambda_{LB}}{2} \right]$ ;
- 5 Shuffle  $\Lambda$  to eliminate any bias due to ordering.
- 6 **Selection Operation:** Obtain parents  $\Lambda_{pr}$  of size  $N_{pr}$  using *k-way tournament selection with replacement* given  $(\Lambda, F)$ ;
- 7 **Crossover Operation:** Obtain crossover solutions  $\Lambda_{cx}$  of size  $N_{op}$  using *k-point crossover* given  $\Lambda_{pr}$  and  $R_{cx}$ ;
- 8 **Mutation Operation:** Obtain mutated solutions  $\Lambda_m$  of size  $N_{op}$  by adding *uniform noise*  $\mathcal{U}(-\Delta\lambda, \Delta\lambda)$  to  $\Lambda_{cx}$  given  $R_m$ ;
- 9 Output final  $\Lambda_E$  by clipping  $\Lambda_m$  for any out-of-bound values subject to  $[\lambda_{LB}, \lambda_{UB}]$ ;

**3) Surrogate for SAnE:** A variety of surrogate models that are commonly adopted in the literature [31], [32], [33] consists of Kriging model, artificial neural networks (ANNs), radial basis function networks (RBFNs), and so on [27]. The choice of the surrogate model should depend on the available computing budget and the nature of the problem at hand. For instance, when the objective function is simple enough to be approximated with low model complexity, then polynomial regression can be considered. For low dimensional problems, the Kriging model can provide relatively better performance [56] using fewer training samples while ANNs can be used for high dimensional problems requiring higher training samples [57]. Since our FTR problem can be considered complex and moderately high-dimensional, the ANN is adopted as our surrogate model. It is pre-trained using

**Algorithm 3** SAnE Algorithm

---

```

1 Input: Population  $\Lambda$  and its fitness  $F$ ,  

surrogate ANN;  

2 Output: New solution samples  $\Lambda_{SAnE}$ ;  

3 Parameters:  

4 Number of sampling times  $S$ ;  

5 Number of surrogate-assisted evolution steps  

 $n$ ;  

6 Surrogate-assisted sampling:  

7 Samples found  $D_S = \{\}$ ;  

8 for  $1, \dots, S$  do  

9    $\Lambda_E \leftarrow \Lambda$ ;  $F'_E \leftarrow F$ ;  

10   for  $1, \dots, n$  do  

11      $\Lambda_E \leftarrow GA(\Lambda_E, F'_E)$ ;  

12      $F'_E \leftarrow ANN(\Lambda_E)$ ;  

13   end  

14    $\Lambda_{SAnE} \leftarrow \Lambda_E$ ;  $F'_{SAnE} \leftarrow F'_E$ ;  

15   Add  $(\Lambda_{SAnE}, F'_{SAnE})$  to  $D_S$ ;  

16 end  

17 Output  $\Lambda_{SAnE}$  by selecting top-N solutions  

from  $D_S$ ;  $D_S =$   

 $\{(\Lambda_{SAnE}, F'_{SAnE})_1, \dots, (\Lambda_{SAnE}, F'_{SAnE})_S\}$ ;
```

---

$N_{LHS}$  training samples generated as part of the initialization procedure as described in Section III-F.1.

4) *Surrogate Update*: To update the parameters of the ANN surrogate, an online machine learning technique, namely *incremental batch learning (IBL)* is often adopted in literature [57], [58]. These works employed IBL by taking one (or more) backpropagation step(s) with a new batch of online samples to update the ANN's weight parameters dynamically at each generation. In this paper, the IBL with one backpropagation step is executed with all new online training samples at each generation. Following the practice of [58], these samples are then discarded afterward. The updated ANN surrogate is then used to assist SAnE in the next generation's evolution process.

Finally, the complete D3EO process is described in Algorithm 1.

## IV. EXPERIMENTS

### A. Experiment Setup

Two real-world signalized intersection datasets were used in our experiments, namely 1) *Lankershim Boulevard Dataset (Dataset<sub>LB</sub>)*, and 2) *Kunming Intersection Dataset (Dataset<sub>KM</sub>)*. The former is a publicly available dataset collected under the NGSIM project [59] while the latter is a proprietary dataset owned by the Alibaba City Brain Lab. The data were collected from their respective study areas as illustrated in Figure 4. On the top left of each sub-figure, the Google satellite view of the study area can be seen. A well-known microscopic traffic simulator, SUMO [43], was adopted in this paper as our MTS model. Therefore, each study area was set up in SUMO as shown on the top right. On the bottom left, the trajectory data are visualized using QGIS.<sup>1</sup>

<sup>1</sup><http://qgis.org>

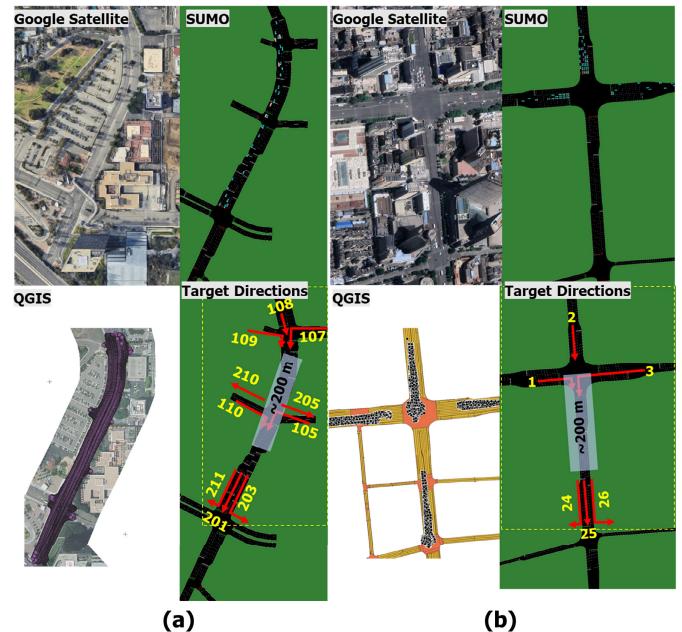


Fig. 4. Study area: (a) Lankershim Boulevard, (b) Kunming Intersection.

Finally, on the bottom right, each target reconstruction area is highlighted with the dotted yellow bound; our traffic flow directions of interest are displayed using solid red arrows with the corresponding origin and destination IDs labeled accordingly.

The choice of the two similar study areas for validation can be justified as follows. In most real-world situations, a target dataset that requires reconstruction (as in the case of *Dataset<sub>KM</sub>*) does not contain any complete ground-truth trajectories, thus posing a challenge to objectively assess the performance of different models. Therefore, setting up experiments in another similar study area where fully complete trajectories are available (such as *Dataset<sub>LB</sub>*) could be beneficial for both model validation and performance assessment.

1) *Lankershim Boulevard Dataset*: The *Dataset<sub>LB</sub>* contains arterial vehicle trajectory data collected from four signalized intersections in Los Angeles, California, United States. This study area is illustrated in Figure 4(a). The data were collected in a morning period between 8:30-8:45 AM. The study area was first imported into SUMO using *Open Street Map*<sup>2</sup> (*OSM*), then adjusted further using SUMO's network editor (*netedit*). Although the traffic light (TL) signal data were not directly available, they were inferred based on the vehicle movements with the trajectory visualization using SUMO. The origin and destination (OD) IDs shown in Figure 4(a) follow the same original IDs provided in the dataset. Since all the trajectories contained in the dataset are complete, this allows us to conduct controlled experiments for assessing the trajectory reconstruction performance under different missing patterns using different optimization models. Detailed information about the data involved in each controlled experiment is given in Section IV-A.3.

<sup>2</sup><https://www.openstreetmap.org/>

2) *Kunming Intersection Dataset*: The  $Dataset_{KM}$  contains the trajectory data collected in a study area (shown in Figure 4(b)) that involves two signalized intersections in Kunming, China in a morning period from 9:20-9:40 AM. Since importing the study area from OSM produces inaccurate road networks, it was manually constructed based on the high-definition (HD) map data. The HD map data allow us to implement very accurate road structures including lanes, edges, lane-to-lane connections, etc. The dataset also includes TL signal timings with each TL cycle having variable lengths. The OD IDs in Figure 4(b) were labeled arbitrarily. The data suffer from missing patterns discussed in Section III-B, and the trajectories are largely incomplete as shown in the figure. The trajectory data were sampled at 1-sec intervals. If a vehicle was not detected during that period, its position was assumed to be missing for that particular time step. Our reconstruction target is 2D vehicle positions at every second. The raw data containing noisy and incomplete trajectories were visualized in SUMO for pre-processing and identifying any anomalies present in the trajectories. After data pre-processing, a total of 481 vehicles appeared in the OD routes of our interest. Among them, 94 valid  $V_{truth}$  trajectories with known IDs were extracted from the (O, D) pairs: (2, 24), (2, 25), and (3, 25). The missing data points in these trajectories are our reconstruction targets. The rest of the vehicle trajectories were assumed to belong to  $V_?$ .

3) *Controlled Experiment Data*: To perform different controlled experiments using  $Dataset_{LB}$ , three main factors influencing the missing patterns found in  $Dataset_{KM}$  can be identified as follows:

- 1) *Undetected intervals (UI)*: This occurs due to the *sporadic sampling*. Based on our observation, the issue occurred mainly when a vehicle first entered (or exited) the study area as well as the hidden zone. This missing pattern can also be seen in the illustration of ground-truth trajectories in Figure 2. In the figure, undetected (missing) points are present at the beginning and end of a partially observed trajectory. The undetected interval level of  $Dataset_{KM}$  was estimated to be about 50%.
- 2) *Unknown ID association (UA)*: The  $Dataset_{KM}$  suffers from this issue severely with about 80% of the vehicles not being associated with a proper unique ID. Again with reference to Figure 2, it can be seen that unknown trajectories found before and after the hidden zone were left unassociated, rendering their reconstruction impracticable.
- 3) *Hidden zone (HZ)*: No ground-truth data were available in the hidden zone. This causes a large spatiotemporal gap in all vehicle trajectories. The length of the hidden zone affecting  $Dataset_{KM}$  was about 200m.

The above factors were used as control variables in our experiments. The routes joining the (O, D) pairs (See Figure 4(a)): (108, 201), (108, 203), and (108, 211), were chosen as our target reconstruction routes. A total of 480 vehicles traveled via these routes, and their trajectories were subject to different control experiments as configured in Table II. The first row of Table II describes information about

TABLE II  
CONTROLLED EXPERIMENT DATA PROFILE

EXP ID	$V_{truth}$	UA (%)	UI (%)	HZ (m)	Missing
1	96	80	50	200	2006
2	192	60	50	200	3985
3	288	40	50	200	5891
4	96	80	30	200	1877
5	96	80	0	0	0

Experiment ID 1. In this experiment, 96 (out of 480) trajectories were treated as our reconstruction target, having a similar missing data profile as  $Dataset_{KM}$ . Based on the chosen sampling interval of 1 sec, any missing data points between the first time and last time of a vehicle's detection within the boundary of the study area is considered *missing*. This definition applies to both datasets. The total number of missing data points to be reconstructed in these 96 trajectories is 2006. The rest of the trajectories were assumed to belong to  $V_?$ . The missing pattern involving (UA, UI, HZ) was set to (80, 50, 200), respectively. Other experiment information in Table II can be explained similarly.

4) *Simulation Setup*: Since SUMO [43] was used as our MTS model, the parameters of its behavioral models – namely *Intelligent Driver Model (IDM)* [38] for car-following, and *LC2013* [40] for lane-changing, each having 5 and 11 constant parameters – are to be calibrated for trajectory reconstruction. Thus, this defines the simulation parameters  $\lambda$  with 16 dimensions. These models were chosen without loss of generality because of their widespread adoption in the MTS literature. It is expected that the proposed approach could also function with other behavioral models without affecting its validity. The measurement units used were primarily in second(s) and meter(m). The simulation step size was chosen as 1 second. Initially, the simulation model was configured with a given  $\lambda$ . At each simulation step, the TL signals were controlled using the actual TL data. Then, new vehicles were placed randomly at a vacant location in their originated edge, with their destination routes chosen with route-probability-weighted sampling. Existing vehicles were controlled by the above behavioral models. All the vehicles that had already arrived at their destination in the current simulation step were removed. Finally, the simulation advanced to the next step. This simulation loop was repeated until the end of the simulation study period. Due to the simulation randomness, five replication runs ( $r$ ) were executed for each  $\lambda$ . The simulation took 965 steps for  $Dataset_{LB}$ -related experiments and 1200 steps for  $Dataset_{KM}$ -related experiments for simulating the real-world duration of about 16 min and 20 min, respectively.

5) *Optimization Setting*: The available simulation budget  $B$  (in terms of the maximum number of simulation runs) was set to 100,000 ( $= r N_{sim} = 5 \times 20,000$ ). For optimization models, 50,000 initial budget was spent for evaluating 10,000 LHS samples and selecting top-N samples as the initial population. These samples were also used to train the ANN surrogate model. The other 50,000 was used for the optimization

process. The optimization budget per generation/iteration was set to 500. This altogether amounts to the optimization duration of 100 generations. The new online samples generated at each generation were used to update the ANN's parameters dynamically. Each model was run five times independently using five different random seeds.

6) *Evaluation Metrics*: The trajectory reconstruction performance is evaluated based on the similarity measure  $J$  defined in Equation (14). However, two types of  $J$  are used to assess the performance: 1)  $J_{OBS}$  that measures the observed similarity between a set of simulated trajectories and a set of partially observed trajectories, and 2)  $J_{TRUE}$  that measures the true similarity between a set of simulated trajectories and a set of complete ground-truth trajectories.

### B. Models for Comparison

The following models were considered for our MTS-based trajectory reconstruction framework.

1) *DEFAULT*: The MTS model parameters were set with default values – the CFM's parameters were set accordingly with reference to [60], and as for the LCM, default parameters in SUMO were used. This serves as the baseline for which no optimization was performed.

2) *RANDOM*: A total of 20,000  $\lambda$  were sampled uniform randomly from the solution space. Then, the best solution (with the maximum fitness score) was used as the MTS model's parameters. This serves as a random brute force baseline where no particular optimization mechanism was involved.

3) *GA*: This is the optimization model using only GA without the SAnE-based sampling branch of our proposed D3EO model in Section III-F. To adhere to the allowed budget spent per generation, a population size of 100 was used, and the population was initialized with top-100 LHS samples.

4) *DDPG*: This is an optimization model based on deep reinforcement learning (DRL), and selected here for comparison because an existing trajectory reconstruction work [11] adopted it for calibrating one parameter of the MTS model. However, unlike [11], a more advanced DRL variant, DDPG [61], was used here to calibrate 16 parameters, with the same population setting as GA.

5) *D3GA*: This is the first variant of our proposed D3EO model, where the number of surrogate-assisted evolution steps ( $n$ ) was set to 10, and SAnE-based sampling was performed only once at each generation (See Figure 3). Since standard evolutionary and SAnE-based sampling requires a total of  $2Nr$  evaluations per generation, a smaller population size of 50 was used to adhere to the allowed budget spent per generation, and the population was initialized with top-50 LHS samples.

6) *D3GA++*: This is the second variant of our proposed D3EO model, with the same parameter setting as D3GA except that SAnE-based sampling was performed 30 times. The name D3GA stands for “*Dynamic Data-driven Genetic Algorithm*” since it leverages a data-driven surrogate model trained using both offline and online data to accelerate the GA-based evolutionary search process.

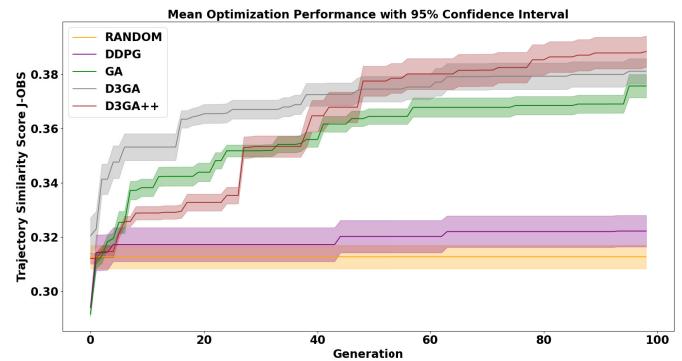


Fig. 5. Optimization performance comparison for EXP 1.

TABLE III  
TRAJECTORY RECONSTRUCTION PERFORMANCE  
UNDER THE DEFAULT SETTING

EXP 1	$J_{OBS}$		$J_{TRUE}$	
	MAX	MEAN	MAX	MEAN
<b>DEFAULT</b>	0.1176	0.1058	0.1275	0.1135
<b>RANDOM</b>	0.3505	0.3126	0.2935	0.2666
<b>GA</b>	0.4041	0.3758	0.3505	0.3281
<b>DDPG</b>	0.377	0.3219	0.2403	0.2475
<b>D3GA</b>	0.4158	0.3811	<b>0.3892</b>	<b>0.3426</b>
<b>D3GA++</b>	<b>0.418</b>	<b>0.3882</b>	0.3689	0.3378

More complete information about each model's training parameters and other experimental details are thoroughly reported in our supplementary materials which are provided online via the GitHub page.<sup>3</sup>

### C. Performance Assessment Under Default Setting

Based on EXP 1 data from Table II, trajectory reconstruction performance of different optimization models are compared in this section. This will be referred to as *default setting* in this paper. The goal is to identify which optimization method achieves the best performance under this default setting where the data contain similar missing patterns as the actual missing data. Figure 5 shows the mean optimization performance in terms of  $J_{OBS}$  by different models across 100 generations. The complete results in terms of both  $J_{OBS}$  and  $J_{TRUE}$  are tabulated in Table III. MAX in the table indicates the maximum similarity score achieved by a model among different runs whereas MEAN indicates the mean similarity score averaged over different runs' results. MAX can be unstable since it relies on a particular random sequence generated while MEAN is a more stable measure since it averages out the effect of different random sequences.

DEFAULT is excluded from the figure since it achieved extremely lower performance than other models. RANDOM achieves the second lowest performance in terms of mean  $J_{OBS}$  in Table III. This implies that a random brute-force search is ineffective for a difficult optimization problem where the solution space is large. This demands the need for an optimization method with a more systematic search strategy.

<sup>3</sup><https://github.com/Javelin1991/FTR-MTS-EO>

When using DDPG, there is a performance improvement over RANDOM in terms of mean  $J_{JOBS}$  by 2.98%. However, its mean  $J_{TRUE}$  is much worse than that of RANDOM by 7.16%. This reveals the challenging nature of our optimization problem encompassing FTR. There exists a gap between the observed reconstruction performance  $J_{JOBS}$  (measured based on the similarity with partially observed trajectories) and the true reconstruction performance  $J_{TRUE}$  (measured based on the similarity with complete trajectories). This gap will be referred to as the *reality gap* for the rest of the paper. In short, higher  $J_{JOBS}$  may not always translate into higher  $J_{TRUE}$ .

In Table 5, the final optimization performance of GA is much superior to that of DDPG with a 16.74% relative increase in terms of mean  $J_{JOBS}$ , which corresponds to an even higher 32.57% relative increase in terms of mean  $J_{TRUE}$ . This significant performance gain is also noticeable in Figure 5. It shows that DRL-based methods may not be suitable for parameter calibration when the parameters are high-dimensional and the cost of acquiring training samples is very high, leading to inefficient and ineffective exploration due to large state and action space.

Since GA has outperformed other baseline models, it serves as a strong baseline to compare against the variants of our proposed model. In terms of mean  $J_{JOBS}$ , both D3GA and D3GA++ surpassed GA by about 1.41% and 3.3%, respectively. It is also evident in the mean optimization curves illustrated in Figure 5. This translates into 4.42% and 2.96% increase in terms of mean  $J_{TRUE}$ , respectively. This performance gain could be attributed to the combination of both standard and surrogate-assisted evolutionary sampling, effectively accelerating the search process to find more optimal solutions under a limited budget. It should be noted that they have attained this performance in spite of their population size being half that of GA while using the same budget spent per generation.

Among the two variants, D3GA++ has achieved the best mean  $J_{JOBS}$  with a 1.86% improvement over D3GA. This improvement could be attributed to the higher number of times SAnE-based sampling is performed in D3GA++. This averages out the accumulated approximation error introduced by the underlying surrogate model (during n-step evolution), ensuring a more consistent search process. However, this could also mean that a potentially good solution explored via a single SAnE-based sampling is not quickly promoted to participate in the overall evolution process. As a result, in Figure 5, the convergence rate of D3GA++ in the earlier generations is much slower than that of D3GA; nevertheless, it eventually surpassed D3GA around 45<sup>th</sup> generation. Furthermore, observing the convergence pattern of GA could also help convey information about how our proposed model would turn out without SAnE-based sampling. Due to the reality gap, the performance gain of D3GA++ over D3GA is reversed when translated into mean  $J_{TRUE}$  with D3GA having a 1.42% increase over D3GA++.

Finally, when comparing the best, D3GA, and the worst, DEFAULT, based on mean  $J_{TRUE}$ , the former has more than 200% performance increase over the latter. This stresses the crucial need for MTS model calibration when it is to be used for FTR.

#### D. Performance Assessment Under Varying Missing Patterns

In this section, the performance of three best-performing EO models: GA, D3GA, and D3GA++ from the previous section is studied under varying missing patterns. There are five different experiment settings listed in Table II giving a total of 10 performance measures across both  $J_{JOBS}$  and  $J_{TRUE}$ . In Figure 6, their mean performances under different missing patterns are shown on the left, and the performance increase/decrease of D3GA and D3GA++ relative to GA is on the right.

According to the figures on the left, one common trend that can be observed is the reality gap between  $J_{JOBS}$  and  $J_{TRUE}$ . The  $J_{JOBS}$  of a given model is almost always higher than  $J_{TRUE}$ . For the figures on the right, it can be seen that D3GA++ outperformed GA in the majority of performance measures (8 out of 10) whereas D3GA is on par with GA (5 out of 10). This supports our hypothesis in the previous section that a higher number of SAnE-based sampling should be favored in our proposed model so that it can average out the accumulated approximation error of the surrogate, and ensure a consistent search process. Thus, when subject to varying missing patterns, D3GA++ is able to maintain its performance consistently unlike D3GA which only performed SAnE-based sampling once. Moreover, in the case where D3GA++ underperformed GA, the performance decrease is merely 0.03%. These results show that D3GA++ outperforms both the baseline GA and its counterpart D3GA consistently under varying missing patterns. More specific analysis for each setting is given in the following sections.

1) *Varying UA Level*: Figure 6.1(a) displays the results when varying the UA level from 80% to 60% and 40% while controlling the UI and HZ at 50% and 200m, respectively. A clear decreasing trend is observed in the figure for all models. This is because when reducing the unknown association level, the number of target reconstruction vehicles increases, and hence, more missing data points (increased from 2006 to 5891) are to be reconstructed in the optimization process. Despite this increase, since the same computing budget was allocated for all experiments, there is a decline in terms of performance measures. Technically, another implication is that the trajectory matching process (See Section III-E.3) becomes harder to solve optimally since more target  $V_{truth}$  trajectories need to be matched with those of  $V_{sim}$  trajectories.

In Figure 6.1(b), when comparing D3GA++ against GA, there is also a marginal 0.36% increase in terms of  $J_{JOBS}$  when the UA level is reduced to 40%, and no difference when the UA level is at 60%. Due to the more challenging nature of the optimization problem, the SAnE-based sampling becomes much less effective under the same limited budget. This might suggest two implications: 1) a higher population size is required to keep solution diversity for harder problems, and 2) a higher number of offline/online samples are required to train the surrogate for more accurate approximation. Lastly, again due to the reality gap, the performance increase is reversed with D3GA++ having a 3.33% mean  $J_{TRUE}$  decrease at 40% UA level. At 60% UA level, there is a highly non-linear translation from 0% mean  $J_{JOBS}$  to 13.57%

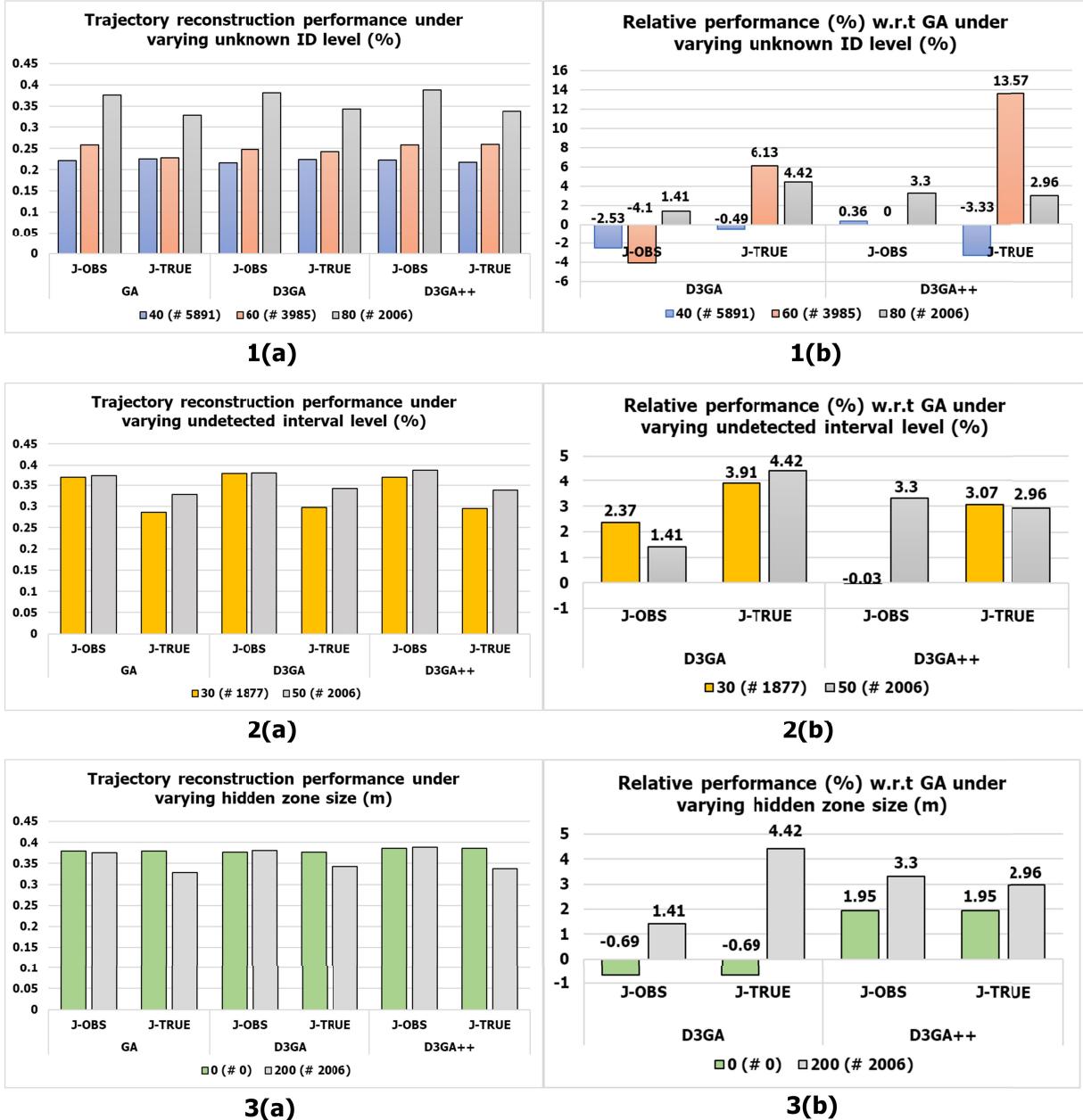


Fig. 6. Reconstruction performance comparison of different EO models under varying missing patterns.

mean  $J_{TRUE}$  increase w.r.t GA. One possible explanation is that there are many local optima with the same  $J_{OBS}$  value in the optimization landscape, but each of these local optima may produce highly different results when evaluated under  $J_{TRUE}$ . This phenomenon needs to be further investigated more carefully in future works.

2) *Varying UI Level:* Figure 6.2(a) plots the results when varying the UI level from 50% to 30% while controlling the UA and HZ at 80% and 200m, respectively. Varying the UI level does not change the number of missing data points a lot, unlike in the case of UA. There is merely a 6.43% decrease from 2006 to 1877 missing data points for a 20% reduction. Therefore, no noticeable difference is observed in the mean  $J_{OBS}$  for all models between 30% and 50%.

However, there are some performance differences in terms of  $J_{TRUE}$  for all models between the two levels. Having no other plausible causes, this could only be attributed to the reality gap induced by having no clear monotonic relationship between  $J_{OBS}$  and  $J_{TRUE}$ .

In Figure 6.2(b), when comparing the two variants of our model against GA, they outperform GA in almost all measures (except one case with only -0.03%). One interesting observation is that the performance increase in terms of  $J_{OBS}$  usually becomes larger when translated into  $J_{TRUE}$ . Again, this may be associated with the issue of the reality gap.

3) *Varying HZ Size:* Figure 6.3(a) shows the results when varying the HZ size from 200m to 0m, and the UI from 50% to 0% while controlling the UA at 80%.

This experiment setup results in zero missing data points needed for reconstruction. This reduces the FTR problem to the MTS parameter calibration problem amidst the co-existence of  $V_g$  and  $V_{truth}$ . In other words, it is the same as calibrating the simulation model by only using fully observed trajectories with known ID associations across different intersections. One crucial observation is that with no missing data points involved, the reality gap no longer exists. Therefore, for all models, their performances in terms of both  $J_{OBS}$  and  $J_{TRUE}$  are exactly the same. As a result, all the models have achieved higher true similarity scores  $J_{TRUE}$  in this experiment setting than in the default setting.

In Figure 6.3(b), D3GA has a very slight performance decrease of 0.69% compared to GA whereas D3GA++ still consistently outperforms GA by 1.95%. When the uncertainty only comes from unknown ID associations, the optimization problem becomes relatively less difficult, and hence, the optimization landscape may become relatively easier to navigate for a model of lower complexity. Consequently, a reduced performance increase of 1.95% is achieved by D3GA++ with respect to GA in this setting as compared to 2.96% in the default setting.

#### E. Performance Assessment With Actual Missing Data

1) *Quantitative Performance Assessment*: In this section, the trajectory reconstruction performance of different EO models: GA, D3GA, and D3GA++, are compared based on  $Dataset_KM$  containing actual missing data. Thus, no complete  $V_{truth}$  trajectories are available to evaluate  $J_{TRUE}$ . A similar situation is also encountered by an existing study [11], which did not conduct any controlled experiments using complete trajectories, unlike our current work. The purpose of this section is to investigate whether the optimization performance of different models observed during controlled experiments is consistent when extended to real-world trajectory data with similar missing patterns.

The experiment results are tabulated in Table IV and optimization performances are shown in Figure 7. In terms of mean  $J_{OBS}$ , D3GA++ has again achieved the best performance among all models whereas D3GA has obtained the worst performance in this experiment. This superior performance of D3GA++ is also reflected in the optimization curves plotted in Figure 7. As compared to GA, D3GA++ has a 4.02% relative increase in terms of mean  $J_{OBS}$ . However, when examining max  $J_{OBS}$ , GA performed better than D3GA++ by 2.32%. As hypothesized earlier, when the optimization problem becomes more difficult, the underlying EO model requires a higher population size and a higher number of offline/online samples for training the surrogate. Since GA employs twice larger population size than our variants, it has an advantage during both initialization and optimization for solution diversification. This facilitates its exploration process to search for more optimal solutions, thus outperforming D3GA++ in terms of max  $J_{OBS}$ .

To verify the above analysis, a new variant of GA, namely GA-2, is also added in this experiment for comparison. It has half the population size of GA. To keep the total budget spent the same, GA-2 was run for twice larger number of

TABLE IV  
TRAJECTORY RECONSTRUCTION PERFORMANCE USING  
THE KUNMING INTERSECTION DATASET

KM	$J_{OBS}$	
	MAX	MEAN
GA	<b>0.1811</b>	0.1591
GA-2	0.1669	0.1534
D3GA	0.1546	0.149
D3GA++	0.177	<b>0.1655</b>

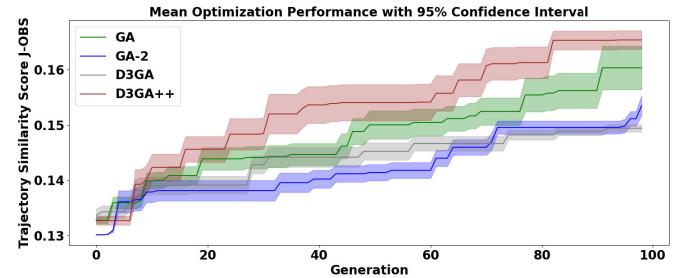


Fig. 7. Optimization performance comparison using the Kunming Intersection Dataset.

generations than other models. Its fitness score after every two generations was recorded for plotting along with others, and shown in Figure 7. As expected, in the figure, its optimization performance across generations is much worse than both GA and D3GA++. As compared to GA-2, D3GA++ shows a 6.05% and 7.89% increase in terms of max and mean  $J_{OBS}$ , respectively. Therefore, the above experiment results have demonstrated that D3GA++ is an effective and consistent EO model to be used along with MTS-based trajectory reconstruction.

2) *Qualitative Performance Assessment*: In Figure 8, some examples of reconstructed trajectories of four randomly selected vehicles are illustrated. For each model, their representative trajectories were taken from their respective runs that achieved the max  $J_{OBS}$ . Partially observed  $V_{truth}$  trajectories which are fragmented into two parts can be seen in the figure. This is caused by the hidden zone where no sensor was available to capture the trajectories of these vehicles for a considerably long distance and duration.

For Vehicle 2149, the reconstructed trajectories given by GA and D3GA++ are very promising as they accurately link the two sub-trajectories that were fragmented due to the hidden zone. The reconstructed trajectory of D3GA is quite far off from the observed ground-truth data points, thus achieving the lowest performance among all models. This could be due to the ineffective trajectory-matching process caused by a poor-quality solution.

For Vehicle 1365, the reconstructed trajectory of D3GA++ matches the partially observed ground-truth data points more closely than the rest. However, it is not as accurate as in the case of Vehicle 2149. This shows that the reconstruction process is harder for the trajectory of a vehicle that encounters multiple stops along its path (which are quite common in congested traffic) than the one that is able to drive without any interruptions (typically common in free-flow traffic).

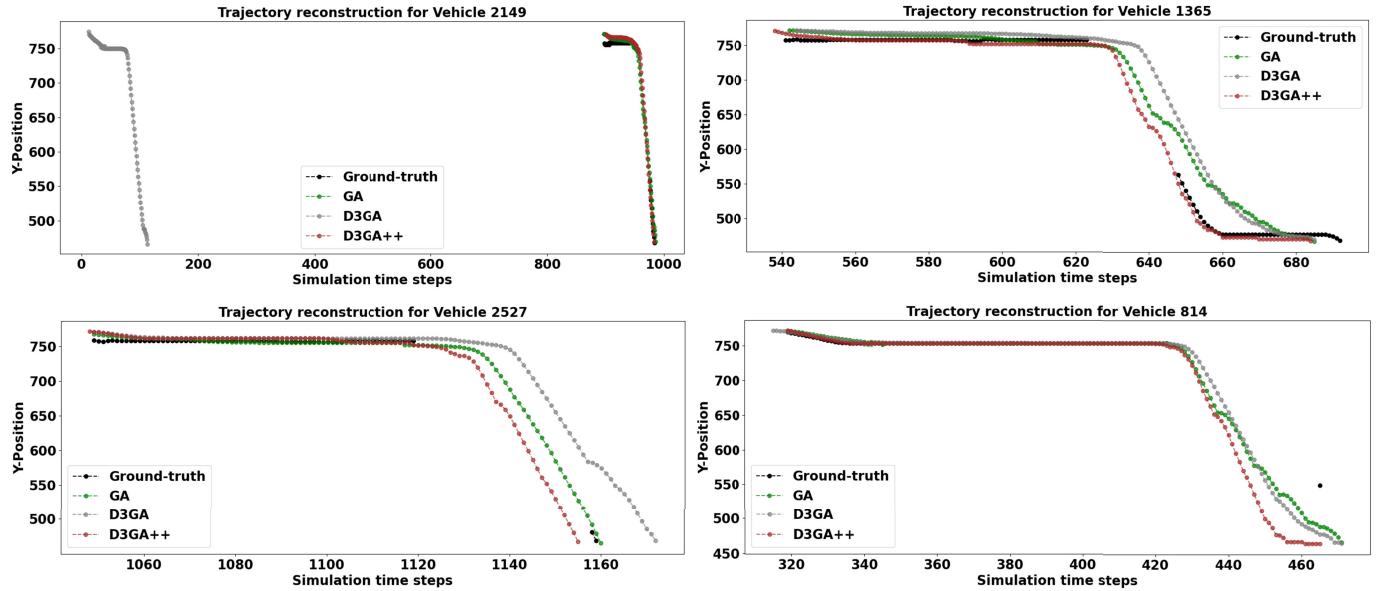


Fig. 8. Examples of reconstructed trajectories for the Kunming Intersection Dataset.

For Vehicle 2527, the reconstructed trajectory of GA closely matches the observed data points. Since there are only two observed data points in the second part of the target trajectory, this renders the reconstruction process difficult for optimization only relying on these few points. Thus, reconstructed trajectories of D3GA and D3GA++ are not as accurate as that of GA.

Finally, for the last Vehicle 814, its reconstruction process is the most challenging among all four vehicles since only one data point is available in the second sub-trajectory. Consequently, the reconstructed trajectories of all models seem unable to link these two sub-trajectories effectively. Nevertheless, these reconstructed trajectories show the promising potential of using our proposed MTS-based optimization framework in combination with the devised D3GA++ in tackling the FTR under challenging missing patterns.

## V. DISCUSSION

### A. Research Implications

Generally, there are three implications of our research targeting FTR as follows:

**1) Improved Traffic Simulation Modeling and Analysis:** The enhanced trajectory reconstruction capabilities can significantly improve traffic flow simulation modeling and analysis, leading to more accurate forecasts and efficient traffic management strategies. This is also reflected in an existing study [7] utilizing an MTS-based reconstruction method demonstrating the benefits of FTR over coarse-grained variational theory approaches. Their experimental results indicated that their proposed method achieved reduced errors in estimating queue positions, travel times, and distances which are critical factors often involved in traffic light signal optimization for efficient traffic management. Furthermore, in another study [6], the MTS-based reconstruction method has been shown to reproduce most shock wave patterns under different traffic densities,

thus potentially improving the shock wave analysis for traffic simulation modeling with incomplete trajectory data.

**2) Policy and Infrastructure Planning:** As highlighted in [8], trajectory data provides insights into land use, transportation demand, and network design, enabling more informed and efficient urban planning. Studies reviewed in [8] have shown how trajectory data helps optimize public services, such as improving bus and taxi operations by identifying mismatches between demand and supply. Therefore, incomplete trajectory data could have negative consequences in these areas, leading to inaccurate traffic analysis, suboptimal infrastructure planning, and flawed public transportation services. With more complete and comprehensive trajectories reconstructed via FTR, urban planners, and policymakers can make better-informed decisions regarding infrastructure development and traffic regulation, resulting in improved smart urban mobility solutions.

**3) Potential in Autonomous Vehicle Technology:** The reconstructed microscopic trajectory data can have direct implications for the development of autonomous vehicle technologies, where accurate and physically consistent trajectory data are critical for safe and efficient navigation. For instance, trajectory reconstruction based on other methods may not always guarantee physical consistency at the microscopic level, often leading to unrealistic values for key variables like speed and acceleration [9], [62]. If such unreliable data were used to develop autonomous vehicle (AV) driving models, it could negatively impact safety. Recognizing this, recent research [9] has increasingly focused on ensuring physical plausibility in trajectory reconstruction. This supports that the MTS-based FTR which maintains the physical consistency of reconstructed trajectories can play a crucial role in ensuring the reliability of these trajectories for AV applications.

### B. Computational Efficiency

In our experiments, the computational efficiency of the proposed method, D3GA++, was not explicitly benchmarked. However, in optimization problems, there is always a trade-off between accuracy and efficiency. If the reconstruction accuracy of the DEFAULT model is sufficient, the MTS-based FTR can be performed with no optimization cost. However, when optimization is needed to achieve the desired level of accuracy, the proposed method proves advantageous by offering more accurate reconstruction results (due to its capability to search more optimal solutions) compared to approaches such as brute force RANDOM, standard GA, and sophisticated DDPG. By coupling the search process with a surrogate model, our approach achieves higher accuracy than the above baselines under the same computing budget, thus demonstrating the efficiency of the *surrogate-assisted n-step evolution* method used in D3GA++.

The advantages of the proposed method are likely to be even more pronounced in large-scale scenarios. However, due to the current scope of this paper, evaluating the method in large-scale settings, such as city-level FTR, is left for future work. The strength of the current method lies in leveraging surrogate-assisted fitness evaluation (in terms of trajectory similarity) to approximate the results of expensive simulation runs. As simulation runtime increases relatively with the size of the study area, the effectiveness of the method in terms of saving simulation runs for optimization is also expected to improve. A consideration for future research is exploring the use of single versus multiple surrogates for approximating simulation results for large-scale scenarios. While a single surrogate may suffice for a few intersections as shown in this paper, multiple surrogates may be more effective for larger areas, where the road network could be partitioned, and the simulation results for each partition could be approximated by its own surrogate.

From a temporal perspective, when longer simulation periods are considered for FTR, changes in traffic regimes must also be accounted for when using surrogate models. In such cases, multiple surrogates can be employed, each responsible for approximating the results of different simulation periods to better capture the dynamic nature of traffic flow across varying regimes.

### C. Comparison With Existing Approaches

It is challenging to perform an objective comparison across all trajectory reconstruction methods due to the diversity of approaches and their application contexts. Therefore, an objective comparison across different approaches remains difficult. However, a qualitative comparison may highlight the unique advantages of our proposed method, which excels in handling large spatiotemporal gaps in vehicle trajectories and ensuring physical consistency while considering vehicle interactions in the reconstruction process. Hence, the key advantages of the proposed method over others can be discussed as follows.

- *Smoothing:* While smoothing methods are effective for interpolating sparsely sampled data, they face challenges when dealing with frequent state changes (e.g., those

induced by traffic lights). Furthermore, smoothing when applied for an incomplete trajectory subject to a relatively large undetected area (such as 200m as studied in this paper) is not reasonable. Our method, designed to handle larger gaps and maintain physical consistency, addresses these shortcomings.

- *Probabilistic Methods:* These approaches typically reconstruct individual GPS-based trajectories, making it difficult to account for interactions among multiple vehicles in close spatial-temporal proximity, which is essential in our context.
- *Particle Filtering:* While particle filters are useful for coarse-grained vehicle path reconstruction, they are not suitable for FTR, the core focus of our work. Since these methods only target path-level reconstruction, it is hard to adapt them for accurate position-level/lane-level trajectory reconstruction as offered by our proposed method.
- *Pure Learning-Based Models:* These models work well for GPS-based trajectory data, but they struggle with large spatiotemporal gaps and lack of ground truth in cases where sensor data is unavailable. This limitation makes an objective comparison against these methods challenging for our specific scenario.
- *Microscopic Traffic Simulation (MTS):* These are ideal for FTR since they consider traffic rules and vehicle interaction in the reconstruction process, and thus, were chosen as our baselines. The experiment results have shown that our proposed method outperforms all variants of MTS-based approaches – DEFAULT, RANDOM, GA, and GA-2.
- *Variational Theory:* MTS-based methods have been shown to outperform variational theory in FTR [7], which further justifies our choice of MTS-based models as baselines.
- *Combination of MTS and Learning-based Models:* The latest research combines MTS-based approaches with learning-based models, as demonstrated in [11]. We adapted this hybrid approach in our baseline comparison. Our experiment results have shown that their RL-based approach, DDPG, is not as effective as our proposed method for FTR.

To summarize, in contrast to other reconstruction methods, our approach integrates MTS for vehicle trajectory reconstruction while inherently considering vehicle interactions through the simulation process. Since most existing methods are not designed for this, a comprehensive objective comparison also seems absent in the current literature and remains an open challenge in the field.

## VI. CONCLUSION

In this paper, a novel trajectory reconstruction framework involving microscopic traffic simulation (MTS) is proposed in order to address a crucial need in fine-grained trajectory reconstruction (FTR), especially in the context of incomplete and irregular traffic data often encountered in real-world scenarios. The framework consists of a new trajectory similarity evaluation method involving MTS, and a newly devised

EO technique, D3GA++. The superiority of D3GA++ over other baseline models, such as brute-force random search, reinforcement learning, and standard evolutionary algorithm, has been demonstrated through an extensive set of quantitative performance assessments as well as through qualitative analysis of reconstructed trajectories. The experiment results have not only demonstrated the effectiveness of our approach but also its practical relevance towards the application in the real-world dataset containing actual missing data.

Future works should focus on developing more sophisticated FTR approaches by leveraging the methods and findings provided in this paper. For instance, based on our experimental findings, a more robust trajectory similarity measure should be developed to establish a proper relationship between the observed similarity (with partial trajectories) and the true similarity (with complete trajectories). Furthermore, how various critical factors in the proposed method such as – 1) computing budget for real function evaluations, 2) population sizes used in D3GA++, and 3) the number of offline/online samples required for a chosen surrogate model – are related to the complexity of the underlying reconstruction problem, should be rigorously studied. Incorporating data from additional sources like IoT devices and mobile sensor data can also further refine the FTR process, potentially improving both its accuracy and reliability. For more complex scenarios, interaction between vehicles and other entities such as bicycles and pedestrians are also important to be taken into consideration and left for future works.

In conclusion, our work underscores the importance of advanced trajectory reconstruction methods in addressing real-world traffic data challenges. The implications of these findings extend beyond theoretical advancements, offering practical solutions that can have a positive impact on traffic management, urban planning, and autonomous vehicle technology. It serves as a new initiative in the field of FTR with its promising potential for further research and development.

## REFERENCES

- [1] Y. Zheng, “Trajectory data mining: An overview,” *ACM Trans. Intell. Syst. Technol. (TIST)*, vol. 6, no. 3, pp. 1–41, 2015.
- [2] S. Wang, Z. Bao, J. S. Culpepper, and G. Cong, “A survey on trajectory data management, analytics, and learning,” *ACM Comput. Surv. (CSUR)*, vol. 54, no. 2, pp. 1–36, 2021.
- [3] X. Tang, H. Yao, Y. Sun, C. Aggarwal, P. Mitra, and S. Wang, “Joint modeling of local and global temporal dynamics for multivariate time series forecasting with missing values,” in *Proc. AAAI Conf. Artif. Intell.*, 2020, vol. 34, no. 4, pp. 5956–5963.
- [4] Z. Wang, S. Zhang, and J. J. Q. Yu, “Reconstruction of missing trajectory data: A deep learning approach,” in *Proc. IEEE 23rd Int. Conf. Intell. Transp. Syst. (ITSC)*, Sep. 2020, pp. 1–6.
- [5] L. Yu et al., “Large scale traffic signal network optimization—A paradigm shift driven by big data,” in *Proc. IEEE 35th Int. Conf. Data Eng. (ICDE)*, Apr. 2019, pp. 1832–1840.
- [6] Y. Wang, L. Wei, and P. Chen, “Trajectory reconstruction for freeway traffic mixed with human-driven vehicles and connected and automated vehicles,” *Transp. Res. C, Emerg. Technol.*, vol. 111, pp. 135–155, Feb. 2020.
- [7] X. Chen, J. Yin, K. Tang, Y. Tian, and J. Sun, “Vehicle trajectory reconstruction at signalized intersections under connected and automated vehicle environment,” *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 10, pp. 17986–18000, Oct. 2022.
- [8] X. Kong et al., “Big trajectory data: A survey of applications and services,” *IEEE Access*, vol. 6, pp. 58295–58306, 2018.
- [9] M. A. Makridis and A. Kouvelas, “Adaptive physics-informed trajectory reconstruction exploiting driver behavior and car dynamics,” *Sci. Rep.*, vol. 13, no. 1, p. 1121, Jan. 2023.
- [10] X. Xie, H. van Lint, and A. Verbraeck, “A generic data assimilation framework for vehicle trajectory reconstruction on signalized urban arterials using particle filters,” *Transp. Res.-C, Emerg. Technol.*, vol. 92, pp. 364–391, Jul. 2018.
- [11] X. Tang et al., “Joint modeling of dense and incomplete trajectories for citywide traffic volume inference,” in *Proc. World Wide Web Conf.*, May 2019, pp. 1806–1817.
- [12] X. Chen, S. Zhang, L. Li, and L. Li, “Adaptive rolling smoothing with heterogeneous data for traffic state estimation and prediction,” *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 4, pp. 1247–1258, Apr. 2019.
- [13] S. P. Venturuthiyil and M. Chunchu, “Vehicle path reconstruction using recursively ensembled low-pass filter (RELP) and adaptive tri-cubic kernel smoother,” *Transp. Res. C, Emerg. Technol.*, vol. 120, Nov. 2020, Art. no. 102847.
- [14] Z. Cao, D. Bryant, T. C. A. Molteno, C. Fox, and M. Parry, “V-spline: An adaptive smoothing spline for trajectory reconstruction,” *Sensors*, vol. 21, no. 9, p. 3215, May 2021.
- [15] P. Hao, K. Boriboonsomsin, G. Wu, and M. Barth, “Probabilistic model for estimating vehicle trajectories using sparse mobile sensor data,” in *Proc. 17th Int. IEEE Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2014, pp. 1363–1368.
- [16] X. Shan, P. Hao, X. Chen, K. Boriboonsomsin, G. Wu, and M. J. Barth, “Probabilistic model for vehicle trajectories reconstruction using sparse mobile sensor data on freeways,” in *Proc. IEEE 19th Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2016, pp. 689–694.
- [17] Y. Feng, J. Sun, and P. Chen, “Vehicle trajectory reconstruction using automatic vehicle identification and traffic count data,” *J. Adv. Transp.*, vol. 49, no. 2, pp. 174–194, Mar. 2015.
- [18] L. Wei, Y. Wang, and P. Chen, “A particle filter-based approach for vehicle trajectory reconstruction using sparse probe data,” *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 5, pp. 2878–2890, May 2021.
- [19] B. Mehran, M. Kuwahara, and F. Naznin, “Implementing kinematic wave theory to reconstruct vehicle trajectories from fixed and probe sensor data,” *Transp. Res. C, Emerg. Technol.*, vol. 20, no. 1, pp. 144–163, Feb. 2012.
- [20] P. Chen, L. Wei, F. Meng, and N. Zheng, “Vehicle trajectory reconstruction for signalized intersections: A hybrid approach integrating Kalman filtering and variational theory,” *Transportmetrica B, Transp. Dyn.*, vol. 9, no. 1, pp. 22–41, Jan. 2021.
- [21] J. Wang, N. Wu, X. Lu, W. X. Zhao, and K. Feng, “Deep trajectory recovery with fine-grained calibration using Kalman filter,” *IEEE Trans. Knowl. Data Eng.*, vol. 33, no. 3, pp. 921–934, Mar. 2021.
- [22] P. Tong, M. Li, M. Li, J. Huang, and X. Hua, “Large-scale vehicle trajectory reconstruction with camera sensing network,” in *Proc. Annu. Int. Conf. Mobile Comput. Netw.*, 2021, pp. 188–200.
- [23] M. Fellendorf and P. Vortisch, “Microscopic traffic flow simulator VISSIM,” in *Fundamentals of Traffic Simulation* (International Series in Operations Research & Management Science), vol. 145. Springer, 2010, pp. 63–93. [Online]. Available: <https://link.springer.com/book/10.1007/978-1-4419-6142-6>
- [24] M. J. Lighthill and G. Whitham, “On kinematic waves I. Flood movement in long rivers,” *Proc. Roy. Soc. London A, Math., Phys. Eng. Sci.*, vol. 229, no. 1178, pp. 281–316, May 1955.
- [25] C. F. Daganzo, “A variational formulation of kinematic waves: Solution methods,” *Transp. Res. B, Methodol.*, vol. 39, no. 10, pp. 934–950, Dec. 2005.
- [26] C. F. Daganzo, “A variational formulation of kinematic waves: Basic theory and complex boundary conditions,” *Transp. Res. B, Methodol.*, vol. 39, no. 2, pp. 187–196, Feb. 2005.
- [27] Y. Jin, H. Wang, T. Chugh, D. Guo, and K. Miettinen, “Data-driven evolutionary optimization: An overview and case studies,” *IEEE Trans. Evol. Comput.*, vol. 23, no. 3, pp. 442–458, Jun. 2018.
- [28] X. Li et al., “An online data-driven evolutionary algorithm-based optimal design of urban stormwater-drainage systems,” *J. Irrigation Drainage Eng.*, vol. 148, no. 11, Nov. 2022, Art. no. 04022041.
- [29] Z.-G. Chen, Z.-H. Zhan, S. Kwong, and J. Zhang, “Evolutionary computation for intelligent transportation in smart cities: A survey,” *IEEE Comput. Intell. Mag.*, vol. 17, no. 2, pp. 83–102, May 2022.

- [30] S. Katoch, S. S. Chauhan, and V. Kumar, "A review on genetic algorithm: Past, present, and future," *Multimedia Tools Appl.*, vol. 80, no. 5, pp. 8091–8126, Feb. 2021.
- [31] H. Wang, Y. Jin, and J. Doherty, "Committee-based active learning for surrogate-assisted particle swarm optimization of expensive problems," *IEEE Trans. Cybern.*, vol. 47, no. 9, pp. 2664–2677, Sep. 2017.
- [32] M. Cui, L. Li, M. Zhou, and A. Abusorrah, "Surrogate-assisted autoencoder-embedded evolutionary optimization algorithm to solve high-dimensional expensive problems," *IEEE Trans. Evol. Comput.*, vol. 26, no. 4, pp. 676–689, Aug. 2022.
- [33] H. Zhen, W. Gong, L. Wang, F. Ming, and Z. Liao, "Two-stage data-driven evolutionary optimization for high-dimensional expensive problems," *IEEE Trans. Cybern.*, vol. 53, no. 4, pp. 2368–2379, Apr. 2023.
- [34] L. Wang, Y. Yao, T. Zhang, C. D. Adenutsi, G. Zhao, and F. Lai, "A novel self-adaptive multi-fidelity surrogate-assisted multi-objective evolutionary algorithm for simulation-based production optimization," *J. Petroleum Sci. Eng.*, vol. 211, Apr. 2022, Art. no. 110111.
- [35] S. Zhou, S. Zheng, M. Treiber, J. Tian, and R. Jiang, "On the calibration of stochastic car following models," 2023, *arXiv:2302.04648*.
- [36] L. H. Lee, E. P. Chew, and P. Manikam, "A general framework on the simulation-based optimization under fixed computing budget," *Eur. J. Oper. Res.*, vol. 174, no. 3, pp. 1828–1841, Nov. 2006.
- [37] P. Emami, P. M. Pardalos, L. Elefteriadou, and S. Ranka, "Machine learning methods for data association in multi-object tracking," *ACM Comput. Surv.*, vol. 53, no. 4, pp. 1–34, Jul. 2021.
- [38] M. Treiber, A. Hennecke, and D. Helbing, "Congested traffic states in empirical observations and microscopic simulations," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 62, no. 2, p. 1805, 2000.
- [39] S. Krauss, "Microscopic modeling of traffic flow: Investigation of collision free vehicle dynamics," Forschungsbericht-Dtsch. Forschungsanstalt fuer Luft-und Raumfahrt e.V., Germany, Tech. Rep. DLR-FB-98-08, 1998, nos. 8–98. [Online]. Available: <https://www.osti.gov/etdweb/biblio/627062>
- [40] J. Erdmann, "Lane-changing model in SUMO," in *Proc. SUMO Model. Mobility Open Data*, vol. 24, 2014, pp. 77–88.
- [41] A. Kesting, M. Treiber, and D. Helbing, "General lane-changing model MOBIL for car-following models," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 1999, no. 1, pp. 86–94, Jan. 2007.
- [42] J. Erdmann and D. Krajzewicz, "SUMO's road intersection model," in *Proc. Simulation Urban Mobility User Conf.* (Lecture Notes in Computer Science), vol. 8594. Springer, 2013, pp. 3–17. [Online]. Available: [https://link.springer.com/chapter/10.1007/978-3-662-45079-6\\_1](https://link.springer.com/chapter/10.1007/978-3-662-45079-6_1)
- [43] M. Behrisch, L. Bieker, J. Erdmann, and D. Krajzewicz, "SUMO—Simulation of urban mobility: An overview," in *Proc. 3rd Int. Conf. Adv. Syst. Simulation*, 2011, pp. 55–60. [Online]. Available: [https://personales.upv.es/thinkmind/SIMUL/SIMUL\\_2011/simul\\_2011\\_3\\_40\\_50150.html](https://personales.upv.es/thinkmind/SIMUL/SIMUL_2011/simul_2011_3_40_50150.html)
- [44] D. K. Hale et al., "Trajectory investigation for enhanced calibration of microsimulation models," Federal Highway Admin., Washington, DC, USA, Tech. Rep. FHWA-HRT-21-071, 2021. [Online]. Available: <https://rosap.ntl.bts.gov/view/dot/57959>
- [45] H. W. Kuhn, "The Hungarian method for the assignment problem," *Nav. Res. Logistics Quart.*, vol. 52, no. 1, pp. 7–21, Feb. 2005.
- [46] H. Su, S. Liu, B. Zheng, X. Zhou, and K. Zheng, "A survey of trajectory distance measures and performance evaluation," *VLDB J.*, vol. 29, no. 1, pp. 3–32, Jan. 2020.
- [47] S. Shang, L. Chen, Z. Wei, C. S. Jensen, K. Zheng, and P. Kalnis, "Trajectory similarity join in spatial networks," *Proc. VLDB Endowment*, vol. 10, no. 11, pp. 1178–1189, Aug. 2017.
- [48] H. Zhen, W. Gong, and L. Wang, "Evolutionary sampling agent for expensive problems," *IEEE Trans. Evol. Comput.*, vol. 27, no. 3, pp. 716–727, Jun. 2023.
- [49] M. Georgiouidakis, N. D. Lagaros, and M. Papadrakakis, "Probabilistic shape design optimization of structural components under fatigue," *Comput. Struct.*, vol. 182, pp. 252–266, Apr. 2017.
- [50] J. O. Agushaka, A. E. Ezugwu, L. Abualigah, S. K. Alharbi, and H. A. E.-W. Khalifa, "Efficient initialization methods for population-based Metaheuristic algorithms: A comparative study," *Arch. Comput. Methods Eng.*, vol. 30, no. 3, pp. 1727–1787, Apr. 2023.
- [51] M. D. McKay, R. J. Beckman, and W. J. Conover, "A comparison of three methods for selecting values of input variables in the analysis of output from a computer code," *Technometrics*, vol. 42, no. 1, pp. 55–61, 2000.
- [52] C. Kamath, "Intelligent sampling for surrogate modeling, hyperparameter optimization, and data analysis," *Mach. Learn. Appl.*, vol. 9, Sep. 2022, Art. no. 100373.
- [53] V. Punzo, B. Ciuffo, and M. Montanino, "Can results of car-following model calibration based on trajectory data be trusted?" *Transp. Res. Record, J. Transp. Res. Board*, vol. 2315, no. 1, pp. 11–24, Jan. 2012.
- [54] V. Punzo, Z. Zheng, and M. Montanino, "About calibration of car-following dynamics of automated and human-driven vehicles: Methodology, guidelines and codes," *Transp. Res. C, Emerg. Technol.*, vol. 128, Jul. 2021, Art. no. 103165.
- [55] A. Sharma, Z. Zheng, and A. Bhaskar, "Is more always better? The impact of vehicular trajectory completeness on car-following model calibration and validation," *Transp. Res. B, Methodol.*, vol. 120, pp. 49–75, Feb. 2019.
- [56] H. Tong, C. Huang, L. L. Minku, and X. Yao, "Surrogate models in evolutionary single-objective optimization: A new taxonomy and experimental study," *Inf. Sci.*, vol. 562, pp. 414–437, Jul. 2021.
- [57] Z.-H. Zhan, J.-Y. Li, S. Kwong, and J. Zhang, "Learning-aided evolution for optimization," *IEEE Trans. Evol. Comput.*, vol. 27, no. 6, pp. 1794–1808, Dec. 2022.
- [58] Y. Jiang, Z.-H. Zhan, K. C. Tan, and J. Zhang, "Knowledge learning for evolutionary computation," *IEEE Trans. Evol. Comput.*, early access, May 19, 2023, doi: [10.1109/TEVC.2023.3278132](https://doi.org/10.1109/TEVC.2023.3278132).
- [59] J. Colyar and J. Halkias, "Lankershim boulevard dataset," Federal Highway Admin. (FHWA), Washington, DC, USA, Tech. FHWA-HRT-07-029, 2007.
- [60] M. Treiber and A. Kesting, *Traffic Flow Dynamics: Data, Models and Simulation*. Berlin, Germany: Springer, 2013, pp. 983–1000.
- [61] T. P. Lillicrap et al., "Continuous control with deep reinforcement learning," 2015, *arXiv:1509.02971*.
- [62] M. Montanino and V. Punzo, "Trajectory data reconstruction and simulation-based validation against macroscopic traffic patterns," *Transp. Res. B, Methodol.*, vol. 80, pp. 82–106, Oct. 2015.



**Htet Naing** received the B.E. degree in computer science from Nanyang Technological University (NTU), Singapore, in 2019, and the Ph.D. degree from NTU, in 2024, specializing in data-driven traffic modeling and simulation with integrated physics and learning. He is currently a Research Scientist with the Centre for Climate Research Singapore, working on physics-informed machine learning for precipitation nowcasting. His research interests include physics-informed machine learning, intelligent transportation systems, and weather prediction.



**Wentong Cai** (Senior Member, IEEE) is currently a Professor and the Associate Dean (Faculty) of the College of Computing and Data Science (CCDS), Nanyang Technological University (NTU), Singapore. His expertise is in the areas of modeling and simulation, particularly in the modeling and simulation of large-scale complex systems and system support for parallel and distributed simulation and distributed virtual environments. He was the winner of the ACM SIGSIM Distinguished Contributions Award in 2023 and is also serving as the Editor-in-Chief for the *ACM Transactions on Modeling and Computer Simulation* (TOMACS) and an Editor for the *Journal of Simulation* (JOS).



**Jinqiang Yu** received the B.Eng. and Ph.D. degrees in electrical engineering from the National University of Singapore, in 2008 and 2015, respectively. He is currently a Senior Algorithm Engineer with Alibaba Cloud Computing. His research interests include intelligent transportation systems, traffic light control, autonomous and connected vehicle, and LLM agent.



**Liang Yu** received the Ph.D. degree in photogrammetry and remote sensing from Wuhan University in 2008. He is currently the Director of data science with Alibaba Cloud. Prior to Alibaba Cloud, he did research at the National University of Singapore, University of Illinois at Urbana-Champaign, Singapore-MIT Alliance for Research and Technology Centre, and the Institute for Infocomm Research, Singapore.



**Jinghui Zhong** received the Ph.D. degree from the School of Information Science and Technology, Sun Yat-sen University, Guangzhou, China, in 2012. From 2013 to 2016, he was a Post-Doctoral Research Fellow with the School of Computer Engineering, Nanyang Technological University, Singapore. He is currently a Professor with the School of Computer Science and Engineering, South China University of Technology, Guangzhou. His research interests include evolutionary computation, machine learning, and agent-based modeling. He has authored or co-authored over 100 journal and conference papers in these areas. He serves as an Editorial Board Member for *Memetic Computing* and *ICT Express*.