Dictionary-based Fidelity Measure for Virtual Traffic

Qianwen Chao, *Member, IEEE,* Zhigang Deng, *Senior Member, IEEE,* Yangxi Xiao, Dunbang He, Qiguang Miao, *Senior Member, IEEE,* and Xiaogang Jin, *Member, IEEE*

Abstract—Aiming at objectively measuring the realism of virtual traffic flows and evaluating the effectiveness of different traffic simulation techniques, this paper introduces a general, dictionary-based learning method to evaluate the fidelity of any traffic trajectory data. First, a *traffic pattern dictionary* that characterizes common patterns of real-world traffic behavior is built offline from pre-collected ground truth traffic data. The corresponding learning error is set as the benchmark of the dictionary-based traffic representation. With the aid of the constructed dictionary, the realism of input simulated traffic flow data can be evaluated by comparing its dictionary-based reconstruction error with the dictionary error benchmark. This evaluation metric can be robustly applied to any simulated traffic flow data; in other words, it is independent of how the traffic data are generated. We demonstrated the effectiveness and robustness of this metric through many experiments on real-world traffic data and various simulated traffic data, comparisons with the state-of-the-art entropy-based similarity metric for aggregate crowd motions, and perceptual evaluation studies.

Index Terms—traffic simulation, crowd animation, data-driven simulation, dictionary learning, user study.

1 Introduction

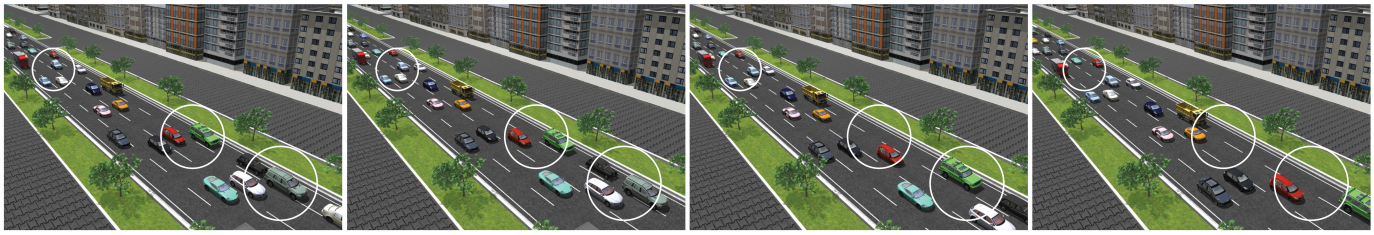
Incorporating realistic traffic flows into virtual environments has become increasingly important with the popularity of virtual reality (VR), smart cities, urban planning, and safety engineering. Traffic simulation could also be an effective tool to generate various kinds of traffic conditions for the use of autonomous vehicles [2]. For instance, simulating highly realistic virtual traffic environments for the test of autonomous vehicle systems is highly desired and cost-effective before real-world road tests. This drives researchers to develop effective methods to simulate traffic on virtual road networks as realistically as possible, including microscopic models [1][3], continuum-based models [4][5], and data-driven traffic visualization [6][7][8]. Despite various progresses as mentioned above, a fundamental question regarding the realism of traffic simulation has largely been under-explored to date, namely, *how can we measure the realism of any synthesized traffic flow?* Current evaluation methods of virtual traffic are often limited to the conducting of subjective user studies, which is unavoidably time-consuming, error-prone, and costly. Indeed, if a quantitative and objective measure can be developed for this purpose, it can be used not only for measuring the realism of various synthesized traffic, but also for objectively comparing the performances of different traffic simulation models in a fair manner.

Previously, several motion parameters, including velocity, acceleration, and vehicle gap, have been used in traffic simulation to validate the effectiveness of the proposed model [9]. Moreover, the polarization factor [10], the number of collisions [11], and the total path length [12] have also been exploited to evaluate the realism of generated crowd dynamics. However, these parameter metrics are domain-specific and are not designed for traffic, making it difficult to capture the essential dynamics of traffic. Recently, Guy et al. [13] proposed an entropy-based similarity metric to measure the prediction error of a given crowd simulator, requiring the ground-truth (real world) crowd data in the *same* environment as the reference. Because of this, it is only defined for a given set of real-world crowd data and thus cannot directly compare different simulators or measure the plausibility of a simulator in the absence of ground truth data. As a result, it would be difficult, without considerable effort, to straightforwardly apply or extend this metric for the fidelity evaluation of virtual traffic flow, since the acquisition of the ground-truth traffic data (as the reference) for *specific virtual* road networks is practically infeasible in most cases.

In the real world, traffic rules and physical laws of driving lead different people to make similar decisions when facing the same driving circumstances. For experienced drivers, such driving decisions or driving patterns are somehow built into their mindsets of driving [14]. In transportation research, the driving pattern concept has been successfully used to characterize driving behaviors, where the driving pattern formulations are typically represented by featuring vehicle states [15]. Inspired by this insight, in this paper we propose a novel pattern dictionary-based method to measure the fidelity of virtual traffic. Its central idea is to learn a *Traffic Pattern Dictionary* (TPD) from a set of real-world traffic data using dictionary learning algorithms. All of the vehicle movement characteristics are treated together as an atom in the TPD.

Specifically, given a collection of real-world traffic trajectory data as input samples, we first employ an adaptive dictionary learning algorithm to build up the TPD for characterizing common traffic behaviors. This iterative learning process is done offline, while the corresponding learning

- *Q. Chao and Q. Miao are with the Department of Computer Science, Xidian University, Xi'an, P. R. China, 710038. E-mail: chaoqianwen15@gmail.coml*
- *Z. Deng is with Computer Science Department, University of Houston, Houston, TX, USA. Y. Xiao, D. He and X. Jin are with State Key Lab of CAD&CG, Zhejiang University, Hangzhou, P. R. China, 310058.*



(a) Data from real world (b) Fidelity score: 0.73 (c) Fidelity score: 5.41 (d) Fidelity score: 8.99

Fig. 1. Fidelity measure comparisons among three virtual traffic flows generated by a microscopic IDM model [1] with three different parameter sets ((b)-(d)). The initial traffic states of the simulator were set as the same as the real-world traffic flow in the same scenario (a). Differences between the simulated traffic and real-world ground truth are highlighted with white circles. For the dictionary-based fidelity evaluation, a smaller value of the metric indicates a higher fidelity of virtual traffic.

error is set as the benchmark of the dictionary-based traffic representation. To evaluate the fidelity of any simulated traffic flow, we then approximate the flow trajectory data through the TPD-based reconstruction. Finally, the plausibility of virtual traffic can be objectively evaluated through the comparison of the resulting reconstruction error and the dictionary-based traffic benchmark. Through direct comparisons with the state-of-the-art entropy-based similarity metric [13] and perceptual evaluation studies, we demonstrate that our approach can effectively measure the fidelity of any virtual traffic trajectories that can be generated by any traffic simulation methods. Fig. 1 shows the evaluation results of several different traffic data.

The main contributions of this work can be summarized as follows:

- A novel dictionary-based scheme to quantitatively and objectively measure the fidelity of any virtual traffic flow through dictionary-based learning and reconstruction. To the best of our knowledge, our dictionary-based evaluation method is the first general framework to objectively measure the plausibility of any traffic flow data.
- An adaptive dictionary learning algorithm to build the TPD dictionary for common traffic behaviors, where atoms are added dynamically according to the current learning error. This effectively removes the time-consuming, trial-and-error parameter tuning process.

2 Related Work

In this section, we first give a focused review on previous related traffic synthesis efforts including traffic control models and data-driven traffic simulation. Then, we also describe prior works on data-driven crowd evaluation.

2.1 Traffic Control Models

In traffic simulation, there are two kinds of widely-used traffic control models, based on the expression level of simulation details: continuum-based macroscopic and agent-based microscopic models.

In macroscopic models, a traffic stream is represented by a continuum in terms of characteristics including flow speed and density [16][17]. Sewall et al. [4] extend the single-lane macroscopic model [5][18] to handle multi-lanes traffic simulation by introducing a lane-changing model and using a discrete visual representation for each vehicle. [19] focus on the lane-changing behavior in flow-based continuum traffic simulations. In general, macroscopic methods are computationally efficient but lack the details of individual vehicle behavior.

By contrast, a microscopic model treats each vehicle as an autonomous agent, whose behavior is controlled based on the instantaneous states of surrounding vehicles and road information. According to the car-following principles [20], researchers have derived a variety of microscopic control models, including the optimal velocity model [21] and the intelligent driver model (IDM) [3]. Shen et al. [1] combine IDM with a flexible lane-changing model for urban traffic simulation. Chao et al. [22] model vehicles' interaction behaviors with pedestrians for mixed traffic simulation. The work of Garcia-Dorado et al. [23] provides users with the flexibility to specify a desired vehicular traffic behavior to a road network. Microscopic methods can simulate complex vehicle behavior details, but only afford a limited scale of traffic due to their computational requirements. To address this issue, Sewall et al. [24] present a hybrid traffic simulation model by integrating continuum and agent-based methods to balance the trade-off between quality and efficiency at runtime.

2.2 Data-driven Traffic Visualization and Simulation

With the development of advanced sensing hardware and computer vision techniques, empirical traffic flow datasets in the forms of video, LiDAR, and GPS sensors are becoming increasingly available. Traffic visualization techniques based on existing data collections have received notable attention in recent years. The works of Sewall et al. [6] and Wilkie et al. [7] reconstruct traffic flow from temporal-spatial data acquired by in-road sensors. Researchers simulate the process of lane-changing in traffic simulation from a pre-collected vehicle trajectory dataset [25], or learn individual-specific driving characteristics from the vehicle trajectory data extracted from driving video samples [9]. Recently, Li et al. [26] propose a city-scale traffic simulation framework from mobile vehicle data (i.e., GPS traces) using statistical learning and metamodel-based optimization. Chao et al. [8] synthesize new vehicle trajectories through the combination of texture synthesis with microscopic traffic behavior rules, given a limited set of vehicle trajectories as the input samples. It is noteworthy that the real-world traffic datasets from the Federal Highway Administration's Next Genera-

tion Simulation (NGSIM) [27] are typically used in the above works.

2.3 Data-driven Crowd Analysis

Researchers have proposed a few methods to evaluate crowd simulators in reference to real-world crowd data. Pettre et al. [28] calibrate a crowd simulator using experimental trajectory samples based on the maximum likelihood estimation technique, and the likelihood estimator is directly used as a metric for quantitative evaluation. Similarly, the works of Wolinski et al. [29] and Ren et al. [30] formulate the evaluation of a simulation algorithm as an optimization problem, by finding a set of parameters that enables the best match between each simulation algorithm and the reference data. An entropy-based metric [13] is introduced to measure the prediction error of a simulator based on a statistical estimation of true crowd states from noisy real-world data. Researchers also compare density-based measures for the output of a simulator with the observed densities in recorded real-world data [31][32].

However, all of the above evaluation approaches require the ground truth data at the *same* scenario/environment as the reference, which seriously limits their applicability, generality, and usefulness. By contrast, our dictionary-based evaluation method does not need such a ground truth reference at the same scenario. Instead, it pre-computes a generalized dictionary from real-world traffic trajectory data. Once the dictionary is constructed, it can be used to evaluate any simulated traffic data.

In addition, there are some efforts on identifying the common intrinsic features of a crowd. Charalambous and Chrysanthou present a data-driven crowd simulation approach based on the Perception-Action Graph [33], where similar temporal perception patterns are identified and grouped together into a graph which in turn is used by simulated agents at run-time. Charalambous et al. [34] derive behavioral metrics from input training trajectories automatically using outlier detection. Recently, Wang et al. [35] proposed an evaluation solution in the absence of ground truth by learning path patterns from groups of individuals. However, the patterns are more concerned with path planning and are totally different from the dictionary in our dictionary-based metric. Furthermore, the proposed path patterns are dependent on the environment, which makes it less suitable for traffic evaluation. In this way, our dictionary-based metric measures the intrinsic properties of traffic and is less affected by the environment.

3 Approach Overview

Conceptually, our approach works as follows: First, it starts to construct the TPD in an offline manner, from vehicle trajectory samples acquired from real-world traffic scenarios that enclose common traffic patterns. The learning error resulted from the TPD construction is set as the benchmark of the dictionary-based traffic representation. Then, given novel simulated traffic flow data, at runtime we can utilize the TPD to reconstruct and approximate the input traffic flow as closely as possible. By comparing the reconstruction error with the above benchmark, we can quantify the realism of the input traffic flow data.

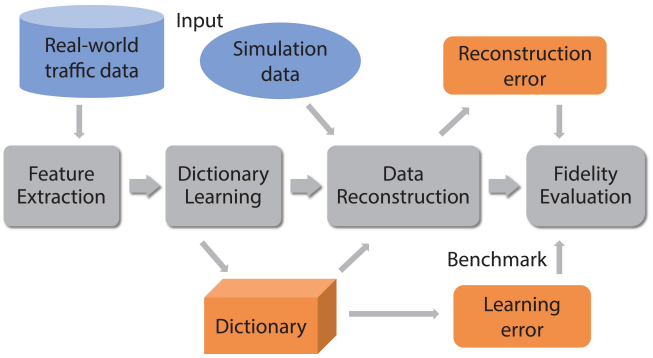


Fig. 2. The pipeline of our approach. The blue boxes show the input of the system, which contains real-world traffic dataset and simulation data to be evaluated.

Technically, our method consists of four stages: (i) the extraction of spatio-temporal traffic flow features, (ii) dictionary learning (i.e., construction of the TPD) from real-world traffic data, (iii) dictionary-based reconstruction of any input traffic flow data, and (iv) the computation of a quantitative measure based on the reconstruction error. Fig. 2 illustrates the pipeline of our approach.

Features extraction. The first stage is to develop a spatio-temporal representation to characterize each vehicle's behavior at any time step. In order to describe the vehicle's ac/deceleration behaviors in the current lane and lane-changing behaviors to the adjacent lanes, we partition each vehicle's movement in two directions: the forward moving direction (i.e., x direction) and the one perpendicular to the forward direction (i.e., y direction).

Inspired by the detailed rules in microscopic traffic control models [1][3][36][37], we extract a vehicle's accelerations a_x, a_y; its driving speeds v_x, v_y; the relative speeds dv_x^L, dv_y^L, dv_x^F, dv_y^F; and its gap distances $d_x^L, d_y^L, d_x^F, d_y^F$ to its leader L (the vehicle immediately in front of it on the same lane) and its follower F (the vehicle immediately following it on the same lane) to describe the vehicle's instantaneous states.

In this way, we have 12 features for each vehicle at each frame as the input to our follow-up dictionary learning process. Furthermore, to encode traffic dynamics, we extract the movement information of vehicles with the duration of t_e seconds (i.e., f frames given a frame rate). Finally, we normalize these input features into the 0-1 range using min-max normalization to remove the variations between different traffic flow samples.

System input. We denote the input real-world traffic flow dataset as $\boldsymbol{S} = \{\boldsymbol{S}_1, \boldsymbol{S}_2, ..., \boldsymbol{S}_i, ..., \boldsymbol{S}_M\}$, where M is the total number of traffic flow samples. The i-th sample $\boldsymbol{S}_i$ contains a few vehicles' trajectories in f frames, denoted as $\{\boldsymbol{s}_i^1, \boldsymbol{s}_i^2, ..., \boldsymbol{s}_i^j, ..., \boldsymbol{s}_i^{N_i}\}$. Here, N_i is the number of vehicles in sample $\boldsymbol{S}_i$. $\boldsymbol{s}_i^j$ is the features of the j-th vehicle. Therefore, the input traffic flow dataset for dictionary learning can be represented in a matrix form: $S = [s_1, ..., s_n] \in \mathbb{R}^{k \times n}$ ($k = 12 \times f$, $n = \sum_{i=1,2,...,M} N_i$).

Similarly, with the definition of an input traffic sample $\boldsymbol{S}_i$, let $\boldsymbol{Y} \in \mathbb{R}^{k \times N_Y}$ be the input traffic simulation data to be evaluated, where N_Y is the number of vehicles.

Dictionary learning. Given the input real-world traffic flow dataset $\boldsymbol{S}$, we aim to build a dictionary (called Traffic Pattern Dictionary, TPD), D, that can well represent

common real-world traffic patterns. Each column in the dictionary is called one *atom*, representing a specific pattern of vehicle behaviors in real-world traffic. Then, each vehicle trajectory in $\boldsymbol{S}$ can be represented as a linear combination of the dictionary atoms. Due to the computational efficiency and tractability, we consider its sparse approximation solution (i.e., with the fewest number of nonzero coefficients) in the dictionary learning process.

Let X_S denote the corresponding coefficient matrix for the dictionary D to represent traffic trajectory dataset S, this dictionary learning problem can be formulated as the following optimization problem with respect to D and X_S:

$$\min_{D,X_S} \frac{1}{kn} \|S - DX_S\|_2^2 + \|X_S\|_1 , \tag{1}$$

where the first term is the learning error, which captures the sparse approximation quality of the dictionary D, and the second term is the sparsity constraint. $\|\cdot\|_1$ denotes the ℓ_1-norm. $\frac{1}{kn}$ is the normalization operator that balances the trade-off between the learning error and sparsity. It also ensures the dictionary-based metric independent of the number of vehicles in the input traffic flow data.

As it is difficult, if not impossible, to determine the number of traffic behavior patterns and accordingly specify a universally optimal dictionary size beforehand, inspired by the work [38] we employ an adaptive dictionary learning method in which the number of dictionary atoms increases dynamically according to the current learning error and the current dictionary size. Because both the dictionary D and the coefficients X_S are unknown, the optimization problem in Eq. (1) can be solved as a convex problem based on the block coordinate descent idea of alternately updating D and X_S within an iterative loop. Specifically, the iterative process can be split into two parts: (i) *dictionary update* (finding an optimal D with a fixed X_S) and (ii) *coefficients update* (finding an optimal X_S with a fixed D).

Dictionary-based reconstruction. The traffic pattern dictionary as computed above covers a wide variety of real-world traffic behavior patterns. Based on the reconstruction quality of an input traffic flow data, the difference between the input traffic and real-world traffic can be judged, thereby quantifying the plausibility of the input traffic.

Specifically, based on the constructed dictionary D, the problem of reconstructing new traffic flow data Y can be viewed as a sparse coding problem in which the input data matrix Y needs to be factorized into D and the optimal coefficient matrix X_Y, that is:

$$\min_{X_Y} \frac{1}{kN_Y} \|Y - DX_Y\|_2^2 + \|X_Y\|_1 . \tag{2}$$

The process of solving X_Y is similar with the coefficient update part in dictionary learning, which will be described in details in Section 4.2. To the end, the reconstruction error can be calculated as $R_Y = \frac{1}{kN_Y} \|Y - DX_Y\|_2^2$.

Fidelity evaluation. The fidelity of the virtual traffic Y can be quantitatively measured by comparing the above reconstruction error R_Y to the benchmark of the dictionary-based traffic representation R_S (described below). After the whole dictionary is built, R_S is computed as follows:

$$R_S = \frac{1}{kn} \|S - DX_S\|_2^2 . \tag{3}$$

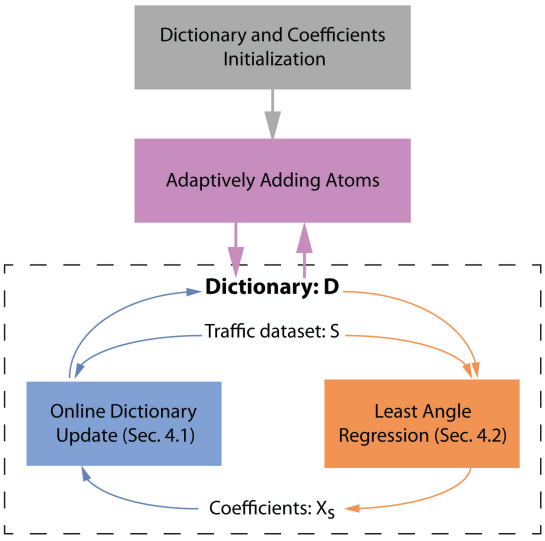


Fig. 3. Illustration of the joint dictionary-coefficients optimization process after adaptively adding atoms to the dictionary. The dictionary D is updated for the input real-world traffic flow dataset S via the online dictionary update algorithm. The least angle regression algorithm is employed to compute the coefficients X_S given the updated dictionary D. This process is repeated until convergence.

Finally, the dictionary-based fidelity metric F_Y for the virtual traffic Y can be computed using the following Eq. (4):

$$F_Y = log_2(\frac{R_Y}{R_S}). \tag{4}$$

The above dictionary-based fidelity metric follows a log-scale, and the base of the log function can be specified by users to obtain different ranges of fidelity scores. In our experiments, the range of fidelity scores is set to $[0, 10]$ with base 2. If simulated traffic Y is more similar to the real-world (training) traffic dataset S, the fidelity score F_Y will have a smaller value; and vice versa.

As the construction of the traffic pattern dictionary is the core part of the dictionary-based fidelity metric, we will describe in detail the dictionary learning algorithm in the following Section 4.

4 Dictionary Learning

Dictionary learning is a representation learning method which aims at finding some meaningful patterns in the input data. Different from Principal Components Analysis (PCA), the extracted dictionary patterns are not required to be orthogonal. Moreover, the sparsity inducing term allows the dictionary to include more items than the dimensionality of the data to be reconstructed, which improves the flexibility of the dictionary representation. Dictionary learning has been widely used in the field of image denoising [39], animation compression [38], reconstruction and classification [40][41][42], facial expression synthesis [43], and video and audio processing [44][45]. These works show that a sparse linear combination of elements from an appropriately chosen dictionary can effectively represent the intrinsic structures of features [43]. The problem in the field of reconstruction and denoising is similar to our traffic flow reconstruction and evaluation problem, whose goal is to

Algorithm 1: Online Dictionary Learning

Input: a matrix of traffic flow sample sets $S = [s_1, ..., s_n] \epsilon \mathbb{R}^{k \times n}$ and an error threshold ε.
Output: The dictionary $D = [d_1, ..., d_m] \epsilon \mathbb{R}^{k \times m}$, coefficient matrix $X_S = [x_1, ..., x_n] \epsilon \mathbb{R}^{m \times n}$ and the representation error R_S.

1 Initialize a dictionary with $m = 2$ atoms: $D = [d_1, d_2]$;
2 Initialize coefficient matrix X_S according to D;
3 **repeat**
4 **for** $t = 1 \rightarrow k(R_S, m)$ **do**
5 Find vehicle z with the largest approximation error;
6 Add feature data s_z to the dictionary D;
7 Update coefficient X_S from vehicle z ;
8 **end**
9 **repeat**
10 Update dictionary D;
11 **for** $i = 1 \rightarrow n$ **do**
12 Update coefficients X_S from vehicle i;
13 **end**
14 **until** *Convergence*;
15 **until** $R_S < \varepsilon$;
16 **return** D *and* X_S

reconstruct arbitrary traffic flow as a linear combination of dictionary atoms for the purpose of evaluating the fidelity of virtual traffic. Inspired by the driving pattern concept in transportation research [15], we can reasonably assume that a virtual traffic flow and its corresponding landmarks share similar (but unknown) intrinsic patterns. These patterns can be characterized by the learned dictionary. Thus, we obtain the traffic pattern landmarks through dictionary learning.

In traditional dictionary learning algorithms, the number of dictionary atoms is pre-defined as an input parameter. However, in our traffic evaluation problem, it is difficult to pre-specify the number of traffic behavior patterns. Therefore, we use an adaptive dictionary learning algorithm to dynamically build and update the dictionary. As illustrated in Fig. 3, our adaptive dictionary learning algorithm consists of the following steps. We first initialize a minimum dictionary D with only 2 atoms and compute an initial coefficient matrix X_S with D. Then, we sequentially add atoms to the dictionary along with jointly optimizing the dictionary D (online dictionary update) and coefficients X_S (coefficients update) until the dictionary-based representation error $R_S < \varepsilon$ (ε is the error threshold specified by users). Its detailed pseudo-code description is given in Algorithm 1.

To start the algorithm, an initial dictionary with 2 atoms (line 1 in Algorithm 1) is built, in which the first atom d_1 of the dictionary is selected from the trajectory features of a random vehicle (i.e., $d_1 = \boldsymbol{s}_{random(n)}$). The second atom d_2 is initialized using the trajectory features of another vehicle that has the smallest dot product with d_1 (i.e., $d_2 = \arg\min_{s_i} \{s_i * d_1\}$). Then, according to the above two atoms d_1 and d_2, the coefficient matrix X_S is initialized (line 2 in Algorithm 1) to minimize the difference between the real-world data and dictionary representation in a least-squares sense: $x_i = \arg\min_{x_i} \|Dx_i - s_i\|_2^2$.

Line 4-8 in Algorithm 1 give the process of adaptively adding atoms to the dictionary. In this work, we automatically determine the optimal size of the traffic pattern dictionary, which is inferred from the given real-world traffic data. For the sake of efficiency, we adaptively add several atoms into the dictionary in each iteration step. The number of added atoms $k(R_S, m)$ is computed based on the current learning error R_S (refer to Eq. 3) and the current dictionary size m:

$$k(R_S, m) = \lambda \sqrt{(\frac{R_S}{\varepsilon} - 1)m} + 1, \tag{5}$$

where λ is a speed control parameter for adding new atoms, which is specified as 1.0 in our experiments. The second constant term 1 ensures that at least one atom is added into the current dictionary when the learning error R_S is approaching the error threshold ε.

Once the number of added atoms is determined, we sequentially add the trajectory feature data s_z of a vehicle z that has the largest representation error to the current dictionary D (see Eq. 6) and update coefficients X_S according to the added atom, using the coefficient update algorithm (Algorithm 3 in Sec. 4.2):

$$D \leftarrow [D, s_z] \quad s.t. \quad z = \arg\max_i \|Dx_i - s_i\|_2^2. \tag{6}$$

With the constructed dictionary after adding several atoms, we seek a solution to the joint dictionary-coefficients optimization problem described in Equation 1 by using the algorithm of online dictionary learning proposed by [46]. That is, we use the method of block-coordinate descent with warm restarts to update the dictionary D and LARS-Lasso algorithm to optimize the coefficients X_S alternatively until convergence (line 9 to line 14 in Algorithm 1). In the following sections, we describe the details of these two major steps: i.e., dictionary update in line 10 (Sec. 4.1) and coefficients update in line 12 (Sec. 4.2).

4.1 Dictionary Update

We employ an online dictionary update algorithm by using block-coordinate descent with warm restarts [47]. Compared to other dictionary update methods (e.g., K-SVD [48] and MOD method [49][50]), it leads to a faster performance, better dictionaries, and more suitable for our traffic evaluation problem. Specifically, the online dictionary update algorithm is parameter-free and does not require any tuning of the learning rate, which eliminates the difficulty of predefining the dictionary size. Moreover, the procedure does not require to store the features $\{s_i\}$ of all the vehicles in traffic flow samples S and the corresponding coefficients $\{x_i\}$, but only precomputes two related matrices A and B as the input for the dictionary update in the following way:

$$A = \sum_{i=1}^{n} x_i x_i^T = [a_1, \cdots, a_m] \in \mathbb{R}^{m \times m}, \tag{7a}$$

$$B = \sum_{i=1}^{n} s_i x_i^T = [b_1, \cdots, b_m] \in \mathbb{R}^{k \times m}. \tag{7b}$$

The pseudo-code description of the dictionary update algorithm is given in Algorithm 2. In each iteration, each atom (column) d_j of the dictionary D is sequentially updated (line 3 to line 5 in Algorithm 2), while freezing the

Algorithm 2: Dictionary Update

Input: Input dictionary $D = [d_1, ..., d_m] \epsilon \mathbb{R}^{k \times m}$,
matrix $A = [a_1, \cdots, a_m] \in \mathbb{R}^{m \times m}$,
and $B = [b_1, \cdots, b_m] \in \mathbb{R}^{k \times m}$.
Output: The updated dictionary D.

repeat
  for $j = 1 \rightarrow m$ **do**
    Update the j-th column of D:
      $u_j \leftarrow \frac{1}{A_{j,j}}(b_j - Da_j) + d_j$;
      $d_j \leftarrow \frac{1}{\max(\|u_j\|_2, 1)} u_j$;
  end
until *Convergence*;
return D

Algorithm 3: Coefficient Update

Input: dictionary $D = [d_1, ..., d_m] \epsilon \mathbb{R}^{k \times m}$,
traffic flow sample sets $S = [s_1, ..., s_n] \epsilon \mathbb{R}^{k \times n}$.
Output: coefficient matrix $X_S = [x_1, ..., x_n] \epsilon \mathbb{R}^{m \times n}$.

1 Initialize $X_S^0 = 0$, $I = supp(X_S) = \phi$, residual $r^0 = S$;
2 Find the dictionary atom d_{j_1} most correlated with r^0;
3 $I \leftarrow I \cup \{j_1\}$;
4 **for** $t = 1 \rightarrow m$ **do**
5 Compute the least-squares moving direction δ^t of X_S^t;
6 **repeat**
7 Update X_S^t from X_S^{t-1} in the direction δ^t;
8 Compute the current residual r^t;
9 **until** *another atom* $d_{j_t} (j_t \nsubseteq I)$ *makes* $|\langle d_{j_t}, r^t \rangle| = ||r^t||$;
10 $I \leftarrow I \cup \{j_t\}$;
11 **end**
12 **return** X_S

other ones under the constraint $d_j^T d_j < 1$. Specifically, d_j is updated as follows:

$$u_j \leftarrow \frac{1}{A_{j,j}}(b_j - Da_j) + d_j, \tag{8a}$$

$$d_j \leftarrow \frac{1}{\max(\|u_j\|_2, 1)} u_j, \tag{8b}$$

where b_j and a_j are the j-th column of B and A, respectively. $A_{j,j}$ is the j-th element in the diagonal of A. Eq. 8b normalizes d_j to satisfy the affinity constraint after its update using Eq. 8a.

With using the value of D at the previous iteration as a warm restart for computing the new dictionary, Algorithm 2 has been empirically found to converge within a small number of iterations. [51] demonstrated that the convergence to a global optimum is guaranteed.

4.2 Coefficients Update

While keeping the updated dictionary D fixed, the coefficients update can be considered as solving the optimization problem in Eq. 1 with ℓ_1-norm penalty over the sparse matrix X_S. This sparse coding problem can be formulated as a ℓ_1-regularized linear least-squares problem, which is called least absolute shrinkage and selection operator (LASSO) [52]. The Least Angle Regression (LARS) algorithm [53][54] is a model selection technique that has been widely used to solve the LASSO problem. It begins with all coefficients initialized with zero and incrementally adds one atom at a time based on the correlations with the current residual. Here, we utilize the LARS to update the coefficients of a given traffic behavior dictionary. The scheme is presented in Algorithm 3.

The process starts with the initial coefficient solution $X_S^0 = 0$ and initial residual $r^0 = S - DX_S^0 = S$. Let I be the active set of dictionary atoms most correlated with the current residual. Initially, $I = \phi$.

The first step of the algorithm (line 2 in Algorithm 3) is applying the dictionary D to the current residual r^0 to find the atom d_{j_1} in D with largest residual correlations (that is, $j_1 = \arg\max_{j=1,...,m} \left| d_j^T r^0 \right|$). Then, j_1 will be the first element in the active set I (line 3 in Algorithm 3).

The main part of LARS is an iterative process (line 4 to line 11 in Algorithm 3). During each iteration, the coefficient X_S is continuously moved towards the least-squares direction until another dictionary atom is equally correlated with the current residual. Then, this atom is added into the active set and a new moving direction is computed.

Specifically, suppose $I = \{j_1, j_2, ..., j_t\}$ is the active set of dictionary atoms that are most correlated with the current residual at the beginning of the t-th step. Then, there will be $t-1$ nonzero vectors in the coefficient matrix X_S^t, and the one x_{j_t} just entered will be zero. If $r^{t-1} = S - D_I X_S^{t-1}$ is the current residual, the least-squares moving direction δ^t of coefficient X_S^t for the t-th step is computed (line 5 in Algorithm 3) using Equation 9:

$$\delta^t(I) = (D_I^T D_I)^{-1} D_I^T r^{t-1}, \tag{9}$$

where D_I represents a subspace spanned by the columns of D belonging to I.

The coefficient profile X_S^t then evolves continuously in the direction δ^t (line 7 in Algorithm 3) that can be computed using Eq. 10:

$$X_S^t(\mu) = X_S^{t-1} + \mu \delta^t, \tag{10}$$

where μ is a heuristic step-size parameter. According to X_S^t, the residual r^t is updated continuously (line 8 in Algorithm 3).

At the same time, keep tracking of any dictionary atom that has not yet been entered into the active set I. As soon as another competitor d_{j_t} in D has as much correlation with the current residual, that is, $|\langle d_{j_t}, r^t \rangle| = ||r^t||$, the moving process in the direction δ^t is paused (line 9 in Algorithm 3). The atom d_{j_t} is added into the active set I (line 10). Then, we set $t = t + 1$ and compute the new least-square moving direction, that is, going back to line 5 in Algorithm 3. This iterative process continues until all the dictionary atoms are added in the active set I. In other words, the Algorithm 3 arrives at the full least-squares solution for coefficient X_S.

We update the dictionary and corresponding coefficients in an iterative way until the representation error R_S of the dictionary is smaller than a pre-defined error threshold ε. In this way, the optimal dictionary can be constructed to represent the common behavior patterns in a given real-world traffic dataset. The representation error R_S is com-

puted using Eq. 3 with the optimal coefficients X_S, and it will be treated as the benchmark of the dictionary-based representation in the fidelity evaluation procedure.

Online reconstruction and fidelity evaluation: Given an input virtual traffic to be evaluated, its spatio-temporal traffic features are sampled at the same frequency as the real-world training data, which leads to the input data Y for dictionary-based reconstruction. The reconstruction process is similar to the coefficients update process described in Sec. 4.2. With the computed optimal coefficients for Y, the reconstruction error R_Y can be calculated and the fidelity of the virtual traffic Y can be quantitatively measured by comparing the reconstruction error R_Y to the error benchmark R_S of the dictionary representation using Eq. 4.

It is noteworthy that with the precomputed, fixed dictionary, the computational time for reconstruction is linear to the number of vehicles in the input traffic flow. In our experiments, the computational time was 0.13s for input traffic data with 154 vehicles, and 7.25s for 6567 vehicles using the dictionary containing 252 atoms.

5 Experimental Results And Evaluations

The real-world traffic datasets we used for dictionary learning are provided by the Next Generation Simulation (NGSIM) program [27], which contains a variety of detailed vehicle trajectories in terms of flow speed and density. The data was collected in both highway networks and traffic-light controlled urban roads at different times using multiple video cameras installed along the road.

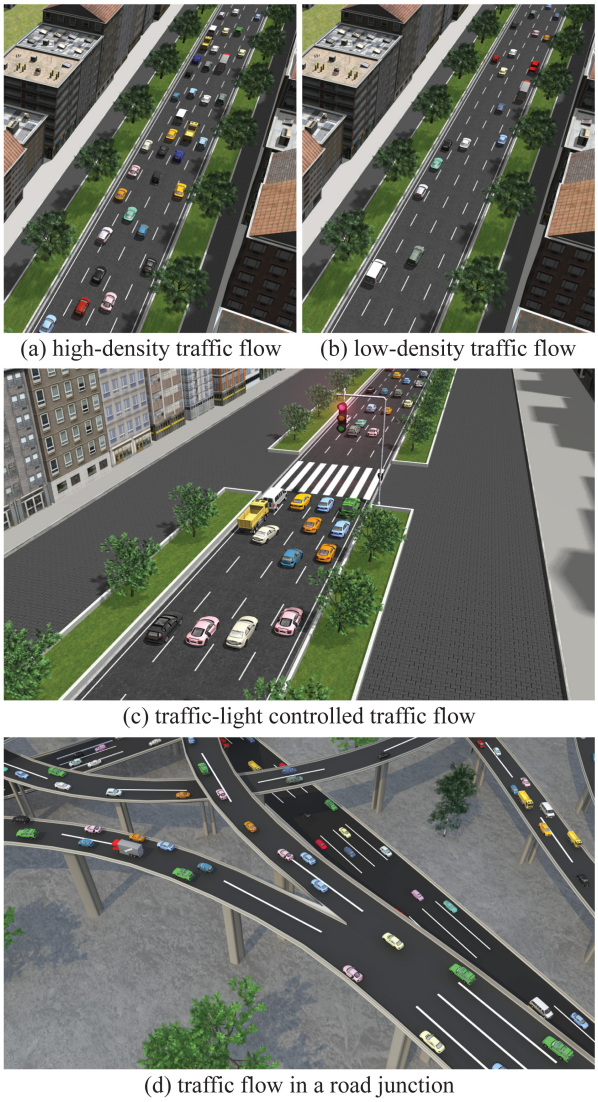
In our implementation, we extracted 80 traffic flow segments from the NGSIM dataset as the input, which contains the trajectory information of a total of 71,529 vehicles with the frequency of 10 frames per second. The duration t_e for the feature extraction of each vehicle was set to 10 seconds (100 frames in total). The error threshold ε for the stopping criterion of Algorithm 1 was specified as 1.0. Based on the above training dataset, the constructed TPD dictionary contains 252 atoms.

First, we tested the performance of our dictionary-based metric on real-world traffic data. We randomly extracted 10 traffic flow segments from the NGSIM dataset, which were not used for the dictionary learning process. These retained test datasets were marked as Real-1 to Real-10. Table 1 shows the fidelity evaluation results using our dictionary-based metric. It can be seen that the scores are distributed in a narrow range between 0.32 and 0.45, in particular compared to the evaluation scores in Table 2, which means the traffic is very realistic.

TABLE 1
Fidelity evaluation scores for real-world traffic data.

Dataset	Real-1	Real-2	Real-3	Real-4	Real-5
Fidelity score	0.35	0.34	0.37	0.45	0.42
Dataset	Real-6	Real-7	Real-8	Real-9	Real-10
Fidelity score	0.34	0.40	0.38	0.41	0.32

In order to evaluate the effectiveness of our dictionary-based metric, we applied it to three representative traffic synthesis methods: (1) one of the latest developments of the microscopic IDM model [1], (2) a continuum traffic control model [4], and (3) a data-driven texture-based traffic synthesis method [8]. For each of the three methods, three different sets of parameters were chosen that vary acceleration, speed, minimum vehicle gap, and other internal simulation parameters. The resulting simulations that use these parameters are referred to as IDM-1, IDM-2, and IDM-3 for the microscopic IDM model, CON-1, CON-2, and CON-3 for the continuum traffic approach, and TEX-1, TEX-2, and TEX-3 for the texture-based synthesis method, respectively.



(a) high-density traffic flow (b) low-density traffic flow

(c) traffic-light controlled traffic flow

(d) traffic flow in a road junction

Fig. 4. The rendering of simulated traffics in different scenarios used for fidelity evaluation.

With the above settings, we evaluated our dictionary-based fidelity metric in four different traffic flow scenarios (see Fig. 4): (1) high-density traffic flow (35 vehicles per mile per lane), (2) low-density traffic flow (18 vehicles per mile per lane), (3) traffic-light controlled traffic flow, and (4) traffic flow in the junctions of road networks.

5.1 Evaluation Results of Virtual Traffic

We applied our dictionary-based metric to evaluate the above simulated traffic flows, and obtained their results in Table 2. From this table, we can see that different simulators

TABLE 2
Fidelity evaluation results for the virtual traffic flow in different scenarios generated by different traffic synthesis methods. The range of the evaluation results is [0, 10], and lower is better. / denotes the unavailability of the corresponding stimulus.

Scenarios	IDM-1	IDM-2	IDM-3	CON-1	CON-2	CON-3	TEX-1	TEX-2	TEX-3
High-density	5.39	4.62	3.49	7.97	7.04	6.03	2.31	3.68	3.76
Low-density	3.96	2.93	4.91	6.13	5.96	3.57	0.57	0.85	1.11
Traffic-light	3.16	6.76	2.42	/	/	/	4.12	5.54	2.03
Road junction	5.07	3.03	3.37	3.63	4.43	0.93	2.15	1.22	2.69

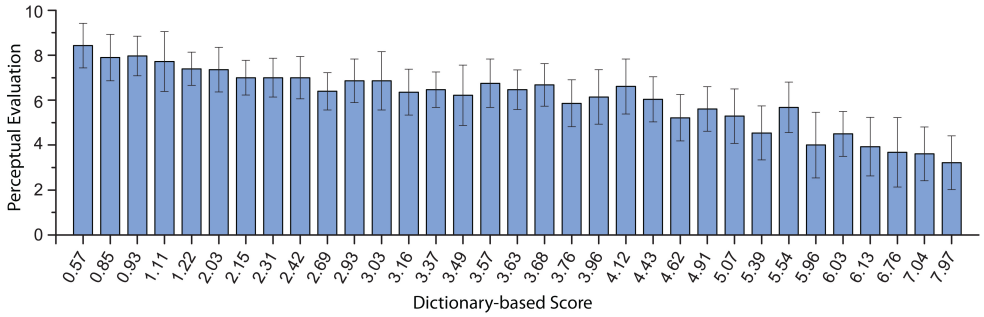


Fig. 5. Comparison between the dictionary-based scores (lower is better) and the perceived similarity scores (higher is better) of 33 virtual traffic flows generated by three different traffic synthesis methods with three different parameter settings specifically for four traffic scenarios. The histograms show the mean values and standard deviations of the perceptual scores.

varied in their capabilities to model the motion characteristics of different traffic scenarios. Furthermore, for a given synthesis method, different parameter sets also scored quite differently, in particular, the optimal parameters for one simulation scenario may not produce a similar performance for another simulation scenario. For example, the TEX-1 on the low-density traffic simulation had a low evaluation score of 0.57, but its evaluation score was 4.12 when the same method with the same parameter setting was applied to simulate a traffic-light scenario.

It is noteworthy that the continuum traffic control model is limited to simulate traffic on highway road networks and cannot simulate street-level traffic with detailed intersections. So, in this table, CON-1, CON-2, and CON-3 did not have the dictionary-based metric scores for traffic-light controlled traffic flow due to the unavailability of the corresponding stimuli. Moreover, the continuum traffic approach tends to get relatively high dictionary-based scores (> 6.0) for *high-density* traffic flows due to its failure in simulating the stop-and-go waves in congestion. By contrast, the microscopic IDM method models the specific behavior of each vehicle, which results in lower scores of the simulated traffic flows than those by the continuum traffic approach. Among the three traffic synthesis methods in this comparison, on average the data-driven traffic synthesis method [8] achieved the best performance in all tested scenarios, thanks to its direct utilization of real-world traffic trajectories.

5.2 Evaluation through Perceptual Study

We conducted a user study to understand and analyze how well the numerical evaluation scores produced by our dictionary-based metric match the ground-truth perceptual similarities obtained through user study. Corresponding to Table 2, we rendered all the 33 simulated traffic animations as the stimuli in our study (4 scenes $\times$ 3 methods $\times$ 3 parameter settings, minus 3 unavailable cases). Then we recruited 30 participants to participate in this user study. All the participants are graduate students in a university, whose ages are in the range of 20 to 30, with normal visions. At each time they were asked to watch one virtual traffic animation stimulus and then rate it in terms of its perceived fidelity. The score range is from 0 (not at all realistic) to 10 (very realistic). To counterbalance the order of the visual stimuli, the stimuli were displayed in a random order for each participant. The participants were allowed to view an animation stimulus many times before scoring it. The outcomes of this user study are illustrated in Fig. 5.

As can be seen in Fig. 5, when the average dictionary-based fidelity score of a stimulus is larger than 6.0, participants often gave the stimulus a low fidelity score, such as from 0.0 to 4.0. On the other hand, when the average dictionary-based fidelity score of a stimulus was smaller than 3.0, participants tended to give it a higher fidelity score such as from 7.0 to 10.0. To further quantify the relationship between the dictionary-based fidelity metrics and the perceptual scores from participants, we also calculated the Pearson's correlation coefficient between the two datasets. The obtained correlation coefficient is -0.96, which indicates that our dictionary-based fidelity metrics are strongly correlated with the perceptual evaluation outcomes. It is noteworthy that for some cases, the evaluation scores by our dictionary-based fidelity metric have more fine granularity than those rated by participants. For instance, as shown in Fig. 5, two virtual traffics received the same perceptual score of 6.86, but they can be distinguished by our dictionary-based fidelity metric, producing the scores 2.93 and 3.03, respectively.

In few cases, however, our dictionary-based fidelity metrics may not be always consistent with the perceptual evaluation scores by participants. For example, as shown in Fig. 5, the two virtual traffics received the dictionary-based scores of 3.16 and 4.12, respectively. This means, measured by our approach, the first one (3.16) is more realistic than

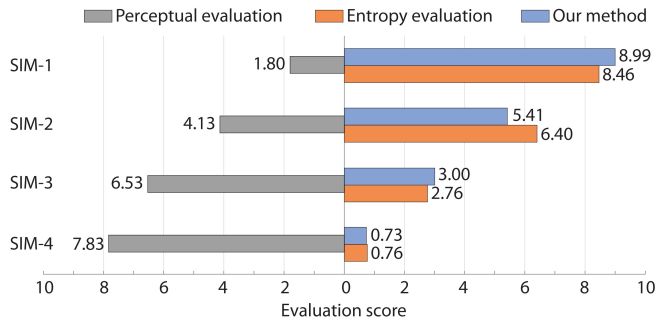


Fig. 6. The comparative user study results among the entropy-based scores (lower is better), our dictionary-based scores (lower is better), and the perceived similarity scores by participants (higher is better).

the second one (4.12). However, their perceptual scores by participants are 6.36 and 6.61, respectively. This shows that on average the participants in our study believed that the second one (6.61) is slightly more realistic than the first one (6.36). Such sporadic inconsistencies could be attributed to the subjective nature of user studies such as unavoidable human bias.

5.3 Comparison with Entropy-based Similarity Metric

We also conducted a direct comparison among the entropy-based similarity metric for aggregated crowd dynamics [13], our dictionary-based fidelity metric, and the perceptual evaluation outcome obtained through user study. The main reason why we chose to compare our metric with the entropy-based metric, instead of other related previous works in crowd animation, is: Our dictionary-based metric and the entropy-based metric are both designed to *quantitatively* measure the plausibility of simulated crowds, although they model different information. Both of them have a similar range of metric scores and make comparisons with perceptual evaluation.

Because the entropy-based metric works at the level of individual motion decision, in this comparison we chose the microscopic IDM model [1] to generate virtual traffic data. With a given set of real-world traffic data, we generated four virtual traffic flows using the above microscopic IDM model with four different sets of parameters. The virtual traffic flow data are labeled as SIM-1, SIM-2, SIM-3, and SIM-4, respectively.

Similar to [13], we also conducted a user study to perceptually evaluate the above animation stimuli. 30 participants were shown two animation stimuli side-by-side at each time. The left is the rendered virtual traffic animation (refer to Fig. 1(b)-(d)) and the right is the rendered animation based on real-world traffic data in the same scenario (Fig. 1(a)). The participants were asked to rate the similarity between the simulated traffic animation (the left) and the reference animation of the real-world traffic (the right) on a Likert Scale 0 (not at all realistic) to 10 (very realistic). The other settings of this user study were similar with those in the previous user study (Sec. 5.2) . The user study results are illustrated in Fig. 6.

As can be clearly seen in Fig. 6, both our dictionary-based fidelity scores and the entropy-based scores are approximately consistent with each other. This indicates that our dictionary-based fidelity metric has a similar capability with the entropy-based metric to measure the plausibility of microscopic traffic simulators.

Furthermore, we wish to point out the differences between our dictionary-based fidelity metric and the entropy-based similarity metric. On one hand, the entropy-based metric is more general than our dictionary-based metric, since the former can be used for general crowds (including virtual traffic), while the latter is specifically narrowed on the fidelity measure of virtual traffic. On the other hand, in terms of measuring the fidelity of virtual traffic, our dictionary-based fidelity metric is more general and versatile than the entropy-based similarity metric, because the entropy-based metric is only defined in reference to a given set of real-world crowd data and cannot measure the plausibility of a simulator/simulation in the absence of the corresponding real-world ground truth data. However, in many scenarios and applications, such ground truth reference data is simply unavailable or infeasible to acquire. This limitation was also clearly mentioned in the original paper of the entropy-based similarity metric [13]. By contrast, with the aid of the precomputed TPD dictionary, our method does not require the corresponding real-world ground truth data as one of the inputs; simulated traffic trajectory data is the only required input for our metric.

5.4 Algorithm Performance Analysis

Convergence analysis: To build the optimal TPD dictionary from the training data with 71,529 vehicle trajectory information, Fig. 7 shows how atoms are added sequentially into the dictionary and how the dictionary-based representation error is decreased over iterations in this process. The curve segment in one color represents one iteration (i.e., one joint dictionary-coefficients optimization process, corresponding to line 4-14 in Algorithm 1), in which one point denotes the representation error after one dictionary-coefficient update step (line 10-13 in Algorithm 1). We can see that the alternative update process converged within 10 iterations owing to the employed warm restart strategy. Furthermore, the number of atoms added into the dictionary also decreased adaptively according to the current representation error and the current dictionary size. The representation error dropped below the error threshold (1.0) after atoms were added 6 times. This can effectively avoid time-consuming, trial-and-error parameter tuning, compared to the case of specifying a fixed number of atoms at the beginning.

Effect of sampling frequency: Regarding the use of traffic patterns to describe vehicle behavior over a fixed period of time, different sampling frequencies of real-world data will result in different granularities of vehicle behavior as expressed by the learned dictionary. In the case of a fixed time period for vehicle feature extraction (10 seconds in our experiments), we tested the effect of sampling frequency of traffic trajectory data on the evaluation outcome by generating three dictionaries with different frame rates (10 fps, 5 fps and 2 fps). The corresponding sample frame numbers in the features of each vehicle are 100, 50, and 20. Note that the sampling frequencies of the evaluated traffic flow and the real-world training data must be consistent. The simulation data SIM-1, SIM-2, SIM-3, and SIM-4 used in Sec. 5.3 were

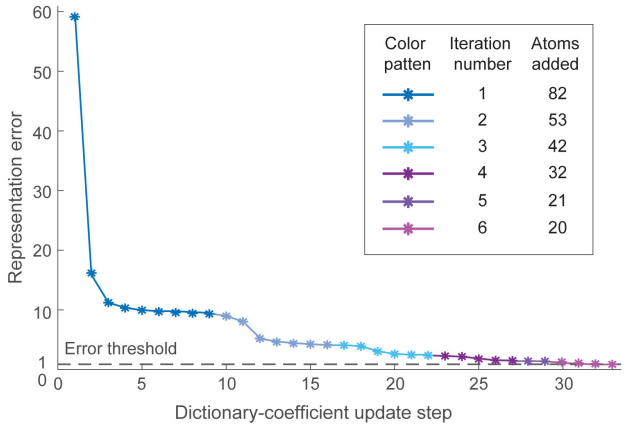


Fig. 7. The convergence of our adaptive dictionary learning algorithm. The curve segment in one color represents one iteration (i.e., one joint dictionary-coefficients optimization process after adaptively adding atoms to the dictionary). One point means the representation error after one dictionary-coefficient update step.

used here to verify the evaluation outcomes. Table 3 shows the dictionary-based evaluation results.

The evaluation scores at the sampling frequency of 10 fps are consistent with those shown in Fig. 5.3, and can be considered as reliable evaluation results, compared with the entropy-based method and user study outcomes. It can be seen from Table 3 that as the sampling frequency is gradually reduced, the discriminative performance of our method gradually becomes inconspicuous. For example, the evaluation scores of SIM-3 and SIM-4 are difficult to distinguish at 5fps (i.e., 3.36 and 3.35, respectively). Therefore, we can infer that in general the larger the sampling frequency, the more accurate the traffic pattern dictionary describing the features of the vehicle behavior, and the more reliable the corresponding evaluation outcome.

TABLE 3
Evaluation results using three dictionaries with sampling frequency.

frame rate (fps)	SIM-1	SIM-2	SIM-3	SIM-4
10	8.99	5.41	3.00	0.73
5	7.94	6.49	3.36	3.35
2	2.78	4.06	2.93	2.47

TABLE 4
Reconstruction errors using five dictionaries with different numbers of atoms.

Atom number	Data-1	Data-2	Data-3	Data-4	Data-5
84	10.29	10.39	11.21	11.24	38.91
179	3.95	3.80	3.56	4.03	13.31
252	1.33	1.79	1.13	1.81	1.80
302	1.33	1.78	1.11	1.79	1.77
352	1.32	1.73	1.10	1.78	1.74

Effect of the dictionary size: To test the effect of dictionary size on the reconstruction of virtual traffic, we tested five dictionaries with different number of atoms on five segments of traffic simulation data: DATA-1, DATA-2, DATA-3, DATA-4, and DATA-5. Table 4 gives the resulting reconstruction errors R_Y. For each simulation data, R_Y decreases as the atom number increases. Our adaptive dictionary learning algorithm indicates the optimal atom number is 252. It is notable that the reconstruction error of DATA-1 and DATA-2 are similar (10.29 and 10.39), when the dictionary contains a relatively small number of atoms (84 atoms). Their fidelity can be clearly distinguished and evaluated only when the number of atoms increases up to 252 which is the optimal atom number selected by our algorithm. The reconstruction error does not have significant changes when more atoms are further added into the dictionary. It should be noted that, an incomplete dictionary could make misleading evaluation in some cases. For instance, Data-2 exhibits a significantly lower value than Data-5 when the atom number is 84, which indicates Data-2 is much more realistic. However, a complete dictionary shows that Data-2 and Data-5 have approximately the same fidelity. This demonstrates the effectiveness of our adaptive dictionary-based fidelity measure.

6 LIMITATIONS

As a common problem with data-driven methods, the composition of the training real-world traffic data has a direct impact on the generated dictionary, thereby further affecting the dictionary-based evaluation outcome. To test the influence, we generated two input real-world traffic datasets with a combination of low-density and high-density traffic flows. These two datasets have the same size but different compositions: one (called the Low-Set) is with 70% low-density and 30% high-density traffic flows, while the other (called the High-Set) has 30% low-density and 70% high-density traffic flows. Then, we generated two TPD dictionaries using the two datasets, respectively. The performances of the dictionaries were tested with three virtual traffic flows in low density (marked as Low-1, Low-2, and Low-3) and three in high density (marked as High-1, High-2, and High-3), respectively. Table 5 shows the dictionary-based evaluation results.

TABLE 5
Fidelity evaluation results of three low-density virtual traffic flows (Low-1, Low-2, and Low-3) and three high-density traffic flows (High-1, High-2, and High-3), using two different dictionaries generated from different real-world traffic datasets (the Low-Set and the High-Set).

Dataset	Low-1	Low-2	Low-3	High-1	High-2	High-3
Low-Set	1.97	8.99	5.75	2.70	8.60	4.50
High-Set	2.98	8.99	6.09	0.84	8.52	2.96

From Table 5, we can see that the virtual traffic flows with low density typically get lower dictionary-based scores (that is, higher plausibility) if the training real-world traffic dataset contains a large proportion of low-density traffic flow data. Similar results can also be obtained for high-density virtual traffic flows. It is notable that, for virtual traffic simulations that are very unrealistic (e.g., Low-2 and High-2), their measured fidelity scores were the same or very close based on the two different training datasets. This is because the vehicles' behavior patterns in unrealistic traffic simulations cannot easily find their matches in both of the two dictionaries.

Finally, in the real world, drivers may make distinct driving decisions on roads with different shapes, such as tend to make a deceleration decision at the corners of a road. However, our current dictionary-based metric does not take the geometric shapes of roads into consideration.

7 Conclusion and Future Work

We present an effective dictionary-based metric to evaluate the fidelity of any traffic simulations, given a training dataset of real-world traffic flows. By adaptively learning a *Traffic Pattern Dictionary* in an unsupervised way to describe the common patterns in vehicle behavior dynamics, our approach can be directly applied to any input traffic flow and objectively measure its fidelity through a dictionary-based reconstruction process. Moreover, our method can also be directly used to quantitatively compare the performances of different traffic simulation techniques.

As the future work, we plan to explore various applications of this evaluation framework, such as the automatic optimization of traffic simulation models according to real-world traffic flow input. In addition, more features on traffic flow should be considered and extracted to describe traffic patterns in the current framework, such as vehicle constraints, road restriction rules, and driver characteristics. However, it is challenging to extract these features from real-world traffic flow data, which is often unavailable in many acquired traffic flow datasets, unfortunately. We also plan to extend this dictionary-based metric to measure several macroscopic aspects of traffic flow motions including flow density and velocity, and generalize this traffic evaluation framework to other kinds of aggregated crowds such as insect swarms, flocks, and human crowds. Last but not least, we are interested in applying the dictionary-based metric to the assessment of autonomous vehicle behaviors.

Acknowledgments

The authors would like to thank Dr. Hen-wei Huang in ETH Zürich for the discussion and support. Qianwen Chao was supported by the National Natural Science Foundation of China (Grant No. 61702393), and the Fundamental Research Funds for the Central Universities (Grant Nos. JBX170310, XJS17052). Zhigang Deng was in part supported by US NSF grant IIS-1524782. Xiaogang Jin was supported by the National Key R&D Program of China (Grant No. 2017YFB1002600) and Artificial Intelligence Research Foundation of Baidu Inc. Qiguang Miao was supported by the National Natural Science Foundations of China (Grant Nos. 61772396, 61472302, 61772392, 61672409), and the Fundamental Research Funds for the Central Universities (Grant Nos. JB170306, JB170304).

References

[1] J. Shen and X. Jin, "Detailed traffic animation for urban road networks," *Graphical Models*, vol. 74, no. 5, pp. 265–282, 2012.

[2] W. Schwarting, J. Alonso-Mora, and D. Rus, "Planning and decision-making for autonomous vehicles," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 1, no. 1, pp. 1–1, 2018.

[3] M. Treiber and D. Helbing, "Microsimulations of freeway traffic including control measures," *at-Automatisierungstechnik Methoden und Anwendungen der Steuerungs-, Regelungs-und Informationstechnik*, vol. 49, no. 11/2001, p. 478, 2001.

[4] J. Sewall, D. Wilkie, P. Merrell, and M. C. Lin, "Continuum traffic simulation," *Computer Graphics Forum*, vol. 29, no. 2, pp. 439–448, 2010.

[5] H. M. Zhang, "A non-equilibrium traffic model devoid of gas-like behavior," *Transportation Research Part B: Methodological*, vol. 36, no. 3, pp. 275–290, 2002.

[6] J. Sewall, J. Van Den Berg, M. Lin, and D. Manocha, "Virtualized traffic: Reconstructing traffic flows from discrete spatiotemporal data," *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 1, pp. 26–37, 2011.

[7] D. Wilkie, J. Sewall, and M. Lin, "Flow reconstruction for data-driven traffic animation," *ACM Transactions on Graphics*, vol. 32, no. 4, p. 89, 2013.

[8] Q. Chao, Z. Deng, J. Ren, Q. Ye, and X. Jin, "Realistic data-driven traffic flow animation using texture synthesis," *IEEE Transactions on Visualization and Computer Graphics*, vol. 24, no. 2, pp. 1167–1178, 2018.

[9] Q. Chao, J. Shen, and X. Jin, "Video-based personalized traffic learning," *Graphical Models*, vol. 75, no. 6, pp. 305–317, 2013.

[10] W. Li, D. Wolinski, J. Pettré, and M. C Lin, "Biologically-inspired visual simulation of insect swarms," *Computer Graphics Forum*, vol. 34, no. 2, pp. 425–434, 2015.

[11] S. Singh, M. Kapadia, P. Faloutsos, and G. Reinman, "Steerbench: a benchmark suite for evaluating steering behaviors," *Computer Animation and Virtual Worlds*, vol. 20, no. 5-6, pp. 533–548, 2009.

[12] M. Kapadia, M. Wang, S. Singh, G. Reinman, and P. Faloutsos, "Scenario space: characterizing coverage, quality, and failure of steering algorithms," in *Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. ACM, 2011, pp. 53–62.

[13] S. J. Guy, J. Van Den Berg, W. Liu, R. Lau, M. C. Lin, and D. Manocha, "A statistical similarity measure for aggregate crowd dynamics," *ACM Transactions on Graphics*, vol. 31, no. 6, p. 190, 2012.

[14] H. Yu, F. Tseng, and R. McGee, "Driving pattern identification for ev range estimation," in *IEEE International Electric Vehicle Conference (IEVC)*. IEEE, 2012, pp. 1–7.

[15] W. Wang, J. Xi, and H. Chen, "Modeling and recognizing driver behavior based on driving data: A survey," *Mathematical Problems in Engineering*, vol. 2014, 2014.

[16] J.-P. Lebacque, S. Mammar, and H. H. Salem, "Generic second order traffic flow modelling," in *Transportation and Traffic Theory 2007*, 2007.

[17] D. Ngoduy, "Multiclass first-order traffic model using stochastic fundamental diagrams," *Transportmetrica*, vol. 7, no. 2, pp. 111–125, 2011.

[18] A. Aw and M. Rascle, "Resurrection of "second order" models of traffic flow," *SIAM Journal on Applied Mathematics*, vol. 60, no. 3, pp. 916–938, 2000.

[19] H. Wang, T. Mao, and Z. Wang, "Modeling interactions in continuum traffic," in *IEEE Virtual Reality*. IEEE, 2014, pp. 123–124.

[20] D. L. Gerlough, "Simulation of freeway traffic on a general-purpose discrete variable computer," Ph.D. dissertation, University of California, Los Angeles, 1955.

[21] M. Bando, K. Hasebe, A. Nakayama, A. Shibata, and Y. Sugiyama, "Dynamical model of traffic congestion and numerical simulation," *Physical Review E*, vol. 51, no. 2, p. 1035, 1995.

[22] Q. Chao, Z. Deng, and X. Jin, "Vehicle–pedestrian interaction for mixed traffic simulation," *Computer Animation and Virtual Worlds*, vol. 26, no. 3-4, pp. 405–412, 2015.

[23] I. Garcia-Dorado, D. G Aliaga, and S. V Ukkusuri, "Designing large-scale interactive traffic animations for urban modeling," *Computer Graphics Forum*, vol. 33, no. 2, pp. 411–420, 2014.

[24] J. Sewall, D. Wilkie, and M. C. Lin, "Interactive hybrid simulation of large-scale traffic," *ACM Transactions on Graphics*, vol. 30, no. 6, p. 135, 2011.

[25] H. Bi, T. Mao, Z. Wang, and Z. Deng, "A data-driven model for lane-changing in traffic simulation," in *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. Eurographics Association, 2016, pp. 149–158.

[26] W. Li, D. Wolinski, and M. C. Lin, "City-scale traffic animation using statistical learning and metamodel-based optimization," *ACM Transactions on Graphics*, vol. 36, no. 6, p. 200, 2017.

[27] "Next generation simulation," https://ops.fhwa.dot.gov/trafficanalysistools/ngsim.htm, 2017.

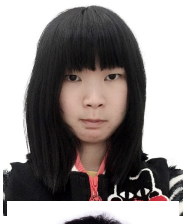
[28] J. Pettré, J. Ondřej, A.-H. Olivier, A. Cretual, and S. Donikian, "Experiment-based modeling, simulation and validation of interactions between virtual walkers," in *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. ACM, 2009, pp. 189–198.

[29] D. Wolinski, S. J Guy, A.-H. Olivier, M. Lin, D. Manocha, and J. Pettré, "Parameter estimation and comparative evaluation of crowd simulations," in *Computer Graphics Forum*, vol. 33, no. 2. Wiley Online Library, 2014, pp. 303–312.

[30] J. Ren, X. Wang, X. Jin, and D. Manocha, "Simulating flying insects using dynamics and data-driven noise modeling to generate diverse collective behaviors," *PloS one*, vol. 11, no. 5, p. e0155698, 2016.
[31] A. Lerner, Y. Chrysanthou, A. Shamir, and D. Cohen-Or, "Data driven evaluation of crowds," *Motion in Games*, pp. 75–83, 2009.
[32] A. Seyfried, M. Boltes, J. Kähler, W. Klingsch, A. Portz, T. Rupprecht, A. Schadschneider, B. Steffen, and A. Winkens, "Enhanced empirical data for the fundamental diagram and the flow through bottlenecks," in *Pedestrian and Evacuation Dynamics 2008*. Springer, 2010, pp. 145–156.
[33] P. Charalambous and Y. Chrysanthou, "The pag crowd: A graph based approach for efficient data-driven crowd simulation," in *Computer Graphics Forum*, vol. 33, no. 8. Wiley Online Library, 2014, pp. 95–108.
[34] P. Charalambous, I. Karamouzas, S. J. Guy, and Y. Chrysanthou, "A data-driven framework for visual crowd analysis," in *Computer Graphics Forum*, vol. 33, no. 7. Wiley Online Library, 2014, pp. 41–50.
[35] H. Wang, J. Ondřej, and C. OSullivan, "Trending paths: A new semantic-level metric for comparing simulated and real crowd data," *IEEE transactions on visualization and computer graphics*, vol. 23, no. 5, pp. 1454–1464, 2017.
[36] A. Kesting, M. Treiber, and D. Helbing, "General lane-changing model mobil for car-following models," *Transportation Research Record: Journal of the Transportation Research Board*, no. 1999, pp. 86–94, 2007.
[37] P. Hidas, "Modelling vehicle interactions in microscopic simulation of merging and weaving," *Transportation Research Part C: Emerging Technologies*, vol. 13, no. 1, pp. 37–62, 2005.
[38] B. H. Le and Z. Deng, "Two-layer sparse compression of dense-weight blend skinning," *ACM Transactions on Graphics*, vol. 32, no. 4, p. 124, 2013.
[39] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Transactions on Image processing*, vol. 15, no. 12, pp. 3736–3745, 2006.
[40] W. Dong, X. Li, L. Zhang, and G. Shi, "Sparsity-based image denoising via dictionary learning and structural clustering," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, pp. 457–464.
[41] S. Ravishankar and Y. Bresler, "Mr image reconstruction from highly undersampled k-space data by dictionary learning," *IEEE transactions on medical imaging*, vol. 30, no. 5, pp. 1028–1041, 2011.
[42] J. Yang, K. Yu, Y. Gong, and T. Huang, "Linear spatial pyramid matching using sparse coding for image classification," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 1794–1801.
[43] H. Liang, R. Liang, M. Song, and X. He, "Coupled dictionary learning for the detail-enhanced synthesis of 3-d facial expressions," *IEEE transactions on cybernetics*, vol. 46, no. 4, pp. 890–901, 2016.
[44] N. Wang, J. Wang, and D.-Y. Yeung, "Online robust non-negative dictionary learning for visual tracking," in *Computer Vision (ICCV), 2013 IEEE International Conference on*. IEEE, 2013, pp. 657–664.
[45] C. D. Sigg, T. Dikk, and J. M. Buhmann, "Speech enhancement using generative dictionary learning," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 6, pp. 1698–1712, 2012.
[46] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online learning for matrix factorization and sparse coding," *Journal of Machine Learning Research*, vol. 11, no. 1, pp. 19–60, 2010.
[47] J. Nocedal and S. J. Wright, *Sequential quadratic programming*. Springer, 2006.
[48] M. Aharon, M. Elad, and A. Bruckstein, "rmk-svd: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Transactions on Signal Processing*, vol. 54, no. 11, pp. 4311–4322, 2006.
[49] K. Engan, K. Skretting, and J. H. Husøy, "Family of iterative ls-based dictionary learning algorithms, ils-dla, for sparse signal representation," *Digital Signal Processing*, vol. 17, no. 1, pp. 32–49, 2007.
[50] K. Engan, S. O. Aase, and J. H. Husoy, "Method of optimal directions for frame design," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 5. IEEE Computer Society, 1999, pp. 2443–2446.
[51] D. P. Bertsekas, *Nonlinear programming*. Athena Scientific Belmont, 1999.
[52] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 267–288, 1996.
[53] M. R. Osborne, B. Presnell, and B. A. Turlach, "A new approach to variable selection in least squares problems," *IMA Journal of Numerical Analysis*, vol. 20, no. 3, pp. 389–403, 2000.
[54] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, "Least angle regression," *The Annals of Statistics*, vol. 32, no. 2, pp. 407–499, 2004.

Qianwen Chao is a Lecturer of Computer Science at Xidian University (China). She earned her Ph.D degree in Computer Science from the State Key Laboratory of CAD&CG, Zhejiang University in 2016. Prior that, she received her B.Sc. degree in computer science in 2011 from Xidian University. Her main research interests include crowd animation, cloth animation and swarm micro-robotics.

Zhigang Deng is a Full Professor of Computer Science at University of Houston. His research interests include computer graphics, computer animation, virtual human modeling and animation, and human computer interaction. He earned his Ph.D. in Computer Science at the Department of Computer Science at the University of Southern California in 2006. Prior that, he also completed B.S. degree in Mathematics from Xiamen University (China), and M.S. in Computer Science from Peking University (China). Besides the CASA 2014 general co-chair and SCA 2015 general co-chair, he currently serves as an Associate Editor of Computer Graphics Forum, and Computer Animation and Virtual Worlds Journal.



Yangxi Xiao is a Ph.D candidate of the State Key Lab of CAD&CG, Zhejiang University, China. She received her B.Sc. degree in digital media technology in 2014 from Zhejiang University. Her main research interests include digital video processing and editing.

Dunbang He is an undergraduate student of the State Key Lab of CAD&CG, Zhejiang University, China. He will receive his B.Eng. degree in Computer Science in 2018 from Zhejiang University. His main research interests include crowd animation and statistical learning.

Qiguang Miao received the M.Eng and Doctor degrees in Computer Science from Xidian University, China. He is currently working as a professor at school of computer science, Xidian University. His research interests include intelligent information processing, intelligent image processing, and multiscale geometric representations for image.

Xiaogang Jin received the BSc degree in computer science, and the MSc and PhD degrees in applied mathematics from Zhejiang University, P. R. China, in 1989, 1992, and 1995, respectively. He is a professor in the State Key Laboratory of CAD&CG, Zhejiang University. His current research interests include traffic simulation, collective behavior simulation, cloth animation, virtual try-on, digital face, implicit surface modeling and applications, creative modeling, computer-generated marbling, sketch-based modeling, and virtual reality. He received an ACM Recognition of Service Award in 2015 and the Best Paper Award from CASA 2017. He is a member of the IEEE and the ACM.