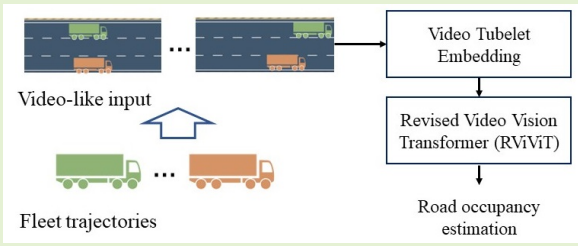
A Revised Video Vision Transformer for Traffic Estimation with Fleet Trajectories

Duo Li, *Senior Member, IEEE*, and Joan Lasenby

***Abstract*— Real-time traffic monitoring represents a key component for transportation management. The increasing penetration rate of connected vehicles with positioning devices encourages the utilization of trajectory data for real-time traffic monitoring. The use of commercial fleet trajectory data could be seen as the first step towards mobile sensing networks. The main objective of this research is to estimate space occupancy of a single road segment with partially observed trajectories (commercial fleet trajectories in our case). We first formulate the trajectory-based traffic estimation as a video computing problem. Then, we reconstruct trajectory series into video-like data by performing spatial discretization. Following this, video input is embedded using a tubelet embedding strategy. Finally, a Revised Video Vision Transformer (RViViT) is proposed to estimate traffic state from video embeddings. The proposed RViViT is tested on a public dataset of naturalistic vehicle trajectories collected from German highways around Cologne during 2017 and 2018. The results witness the effectiveness of the proposed method in traffic estimation with partially observed trajectories.**



***Index Terms*— Traffic estimation; Vehicle trajectory; Deep learning**

I. Introduction

REAL-TIME traffic monitoring represents a key component for transportation management, which provide essential data for many applications, such as route planning, congestion detection, dynamic traffic assignment and demand prediction. Existing in-operation monitoring systems require sensing networks consisting of hundreds or even thousands of fixed-point sensors (e.g., cameras and loop detectors). These large networks represent a major cost for authorities in terms of installation and maintenance.

More than half of the vehicles shipped world-wide are connected vehicles with embedded modems in 2019 [1]. The increasing penetration rate of connected vehicles with positioning devices encourages the utilization of trajectory data for real-time traffic monitoring. The use of commercial fleet trajectory data could be seen as the first step towards mobile sensing networks. Nowadays, commercial fleet tracking systems can generate sufficient data for traffic monitoring, and there is an urgent need for effective tools to process and analyse the huge amounts of data. Teletrac Navman [2] reported that in 2019, 86% of fleets used telematics which is the technology used to monitor a wide range of information relating to an individual vehicle or an entire fleet, and 74% of fleets were tracked with telematics; while only 23% of fleets applied big data analytics to guide decision-making.

The main objective of this research is to estimate traffic state with partially observed data (i.e., commercial fleet trajectories). We propose a pure transformer [3] deep learning architecture - Revised Video Vision Transformer (RViViT). The proposed RViViT is tested on a public dataset of natural vehicle trajectories collected from German highways around Cologne during 2017 and 2018. The main contributions of this paper are as follows.

- We formulate traffic estimation with partial trajectories as a video-like computing problem. Because vehicle trajectories are usually stored as time series, which cannot be directly used to model spatial interactions among vehicles, we apply spatial discretization and temporal sampling to re-map trajectory series into video-like data.
- To the best of our knowledge, this is the first attempt to apply a transformer-like architecture in solving trajectory-based traffic estimation problems. Currently, the transformer is the most prominent architecture in sequence-to-sequence modelling [4] [5], and has shown its capacity of learning from images [6] [7]. Therefore, it is reasonable to explore the potential of such architectures in computing video-like data generated from vehicle trajectories.

"This study was sponsored by the Engineering and Physical Sciences Research Council (EPSRC) (Project No.EP/R035199/1)."

Duo Li was with the Department of Engineering, University of Cambridge, Cambridge CB2 1PZ, United Kingdom. He is now with the Department of Engineering, Nottingham Trent University, Nottingham NG1 4FQ, United Kingdom (e-mail: dl655@cam.ac.uk).

Joan Lasenby is with the Department of Engineering, University of Cambridge, Cambridge CB2 1PZ, United Kingdom (e-mail: jl@eng.cam.ac.uk).

- The original Video Vision Transformer (ViViT) [8] was designed for video classification tasks. In order to estimate a continuous traffic state, we extend the ViViT to have regression functionality. A comprehensive analysis of the Revised ViViT (RViViT) is performed in terms of input resolution, model complexity and training data size.

II. Related Works

A. Traffic Estimation

Traffic estimation methods can be broadly grouped into two categories, namely, model-driven and data-driven methods. Model-driven methods use physical models describing traffic flow dynamics. Physical models are usually built on the basis of empirical relations, in which model parameters are either exogenously calibrated using empirical data or endogenously estimated within the methods. In previous studies, different models were employed for traffic estimation. The fundamental diagram (FD), describing relations among speed, density and flow, is commonly used in combination with other traffic models for estimation [9] [10] [11] [12]. Macroscopic traffic models, using the concept of the FD as a basis for their flux functions, have been extensively adopted by model-driven methods. For example, the first-order Lighthill-Whitham-Richards (LWR) model and its extentions [13] [14], the second-order Payne-Whitham (PW) model and its extensions [15] [16], and the second-order Aw-Rascle-Zhang (ARZ) model and its extensions [17] [18], have been used in previous studies [19] [20] [21] [22] [23] [24] [25] [26] [27] [28].

Data-driven methods require no physical traffic flow model and only rely on empirical data. By using statistical or Machine Learning (ML) algorithms, the methods capture dependencies from empirical data, and then estimate traffic state based on the extracted dependencies and real-time information. Statistical approaches were often used in early studies. For example, several heuristic and statistical data imputation methods were analyzed in [29]; linear regression and Auto-Regressive Integrated Moving Average (ARIMA) models were developed for estimation purposes [30] [31]. Later, ML and Bayesian statistics were introduced in the field of traffic estimation, such as in Bayesian Networks (BN) [32], Kernel Regression (KR) [33], Fuzzy C-Means (FCM) [34], K-Nearest Neighbors (KNN) [35], Principal Component Analysis (PCA) [36] [37], Tucker decomposition (TD) [38], and Bayesian particle filters (BPF) [39]. Recently, a number of attention based models were proposed to model spatiotemporally varying traffic states [40] [41]. A comprehensive summary of traffic estimation methods can be found in [42]

B. Deep Learning on Spatial and Temporal Data

Deep learning is part of a broader family of ML methods. Different types of Deep Neural Networks (DNNs) have been developed for a variety of tasks. For example, Convolutional Neural Networks (CNNs) realize convolutional and pooling operations for processing image-like data [43] [44]; Recurrent Neural Networks (RNNs) introduce recurrent cells to process time series data [45] [46]; Graph Neural Networks (GNNs) capture the dependencies of graphs via message passing between their nodes [47] [48]. These DNNs have been widely used in spatio-temporal traffic estimation and prediction studies, e.g., [49] [50] [41] [51] [52] [53] [54] [55].

A *transformer* is a new type of DNN, which uses self-attention mechanisms to extract intrinsic features [3]. Transformers were first introduced to the field of Natural Language Processing (NLP) where they achieved remarkable success [56] [57]. For example, when Bidirectional Encoder Representations from Transformers (BERT) [56] was published, it achieved state-of-the-art performance on 11 NLP tasks; Generative Pre-trained Transformer 3 (GPT-3) [57] was pre-trained on a large amount of compressed plain text data and showed strong performance on different types of downstream natural language tasks without requiring any fine-tuning. Because of the major success of transformer architectures in processing sequential data, researchers have introduced transformers to the computer vision (CV) field where CNNs were once seen as the fundamental component. Nowadays, the transformer is showing that it is a promising alternative to CNN. A number of visual transformers have been proposed [7] [58] [59] [6], and most of them yielded state-of-the-art performance on multiple image recognition benchmarks. Recently, researchers have explored the potential of transformer-like architectures in video modelling. A few transformer-like architectures have been proposed to model long-range contextual relationships in video, such as Time-Space Transformer (TimeSformer) [60] and Video Vision Transformer (ViViT) [8].

In general, transformers have showed great potential in spatial and temporal data processing. It is interesting to develop suitable transformer-like architectures for spatio-temporal traffic estimation.

III. Problem Formulation

Vehicle trajectories contain instantaneous information of vehicles at each time step, and are usually stored as time series. However, these trajectory series cannot be directly used to model spatial interactions among vehicles. Thus, we first reconstruct trajectory series into video-like data. Given commercial fleet trajectories with instant position p_v, acceleration a_v and velocity v_v at each time step and corresponding vehicle size information, we map these trajectories into a number of video clips, as depicted in Fig.1. Each video with the shape $(T; H; W; C)$ is a collection of T images. A road segment is discretized and divided into H W cells: H is the number of lanes of the segment and W is equal to the segment length divided by the cell width. Each image has C=2 channels providing vehicle velocity and acceleration information.

Spatial discretization enables us to measure space occupancy, o_s, that can provide improved traffic measurement by considering vehicle sizes [61]. o_s is calculated as the percentage of a road segment occupied by vehicles

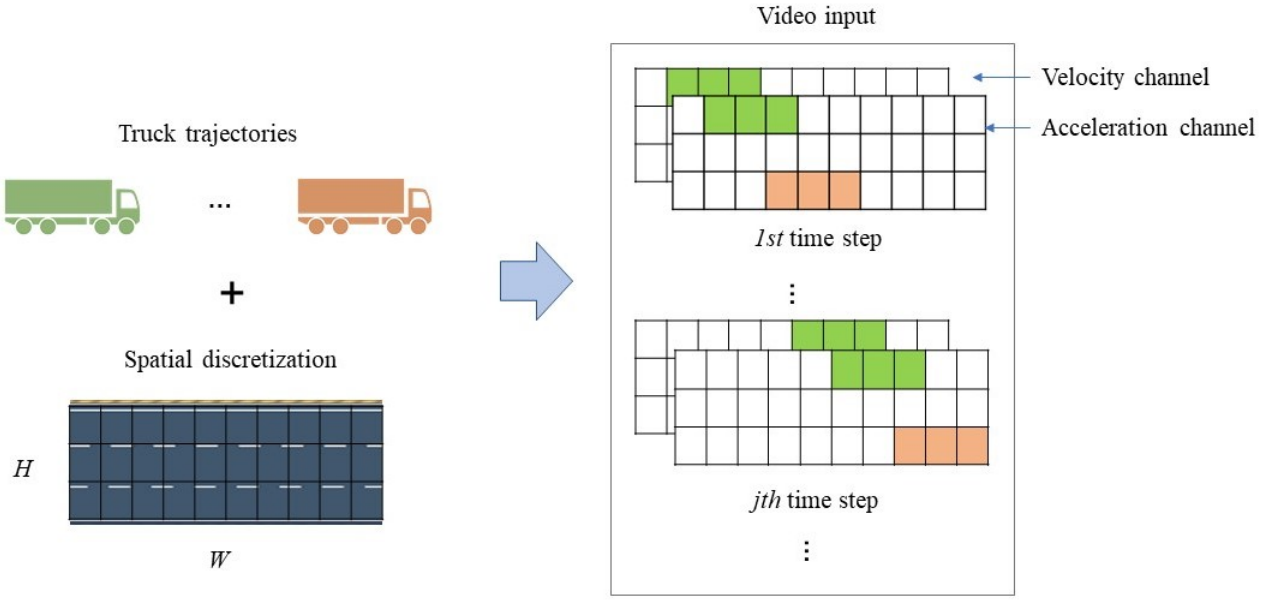


Fig. 1. Video input generation

$$o_s = L_{ocp}/L_{all} \quad 100 \quad N_{ocp}/N_{all} \quad 100 \tag{1}$$

where, L_{ocp} is the occupied road length; L_{all} is the total road length; N_{ocp} is the occupied road cells; and N_{all} is the total road cells.

Traffic estimation with partially observed trajectories is then formulated as a video computing problem. Given a video input $V^j \in \mathrm{R}^{T\times H\times W\times C}$ generated from commercial fleet trajectories, $fp_v^{j-T}; a_v^{j-T}; v_v^{j-T}; :::; p_v^j; a_v^j; v_v^j g$, between $j \quad T$ and j time steps, we estimate the space occupancy, o_s^j, resulting from all types of vehicles using a video modelling framework $f(\,)$. The problem can be represented by

$$fp_v^{j-T}; a_v^{j-T}; v_v^{j-T}; :::; p_v^j; a_v^j; v_v^j g \;!\; V^{j\ f(\cdot)}\!! \; o_s^j \tag{2}$$

It is noted that the fleet trajectories and video input share the same set of time steps which is also shared by the trajectories of all the vehicles used for model evaluation later.

IV. Revised Video Vision Transformer (RViViT)

In this section, we first provide preliminary information regarding transformer architectures. Then, we describe video embedding strategies for transforming video data into an input sequence. Finally, the architecture of the proposed RViViT is presented.

A. Revisiting Standard and Vision Transformers

A standard transformer consists of an encoder module and a decoder module, which consist of several encoders and decoders, respectively. Each encoder/decoder is composed of a Multi-head Self-Attention (MSA) layer and a feed-forward neural network. MSA is the key component of a transformer. Self-Attention (SA) extracts feature representations Q (query), K (key) and V (value) from an input sequence $z \in \mathrm{R}^{N\times D}$ using three parameter matrices $W_Q \in \mathrm{R}^{D\times D_q}$, $W_K \in \mathrm{R}^{D\times D_k}$, and $W_v \in \mathrm{R}^{D\times D_v}$, where $D; D_q; D_k$ and D_v are the dimensions of input, query, key and value, respectively.

$$Q = zW_q; \;\; K = zW_k; \;\; V = zW_v \tag{3}$$

$$\mathrm{SA}(z) = \mathrm{softmax}(QK^{\mathrm{T}}/\sqrt{D_q})V \tag{4}$$

MSA performs N_{head} SA operations, know as "heads". The outputs of these heads are concatenated and then projected using a parameter matrix $W_{MSA} \in \mathrm{R}^{D_k N_{head}\times D_{out}}$, where D_{out} is the output dimension.

$$\mathrm{MSA}(z) = \mathrm{concat}[\mathrm{SA}^1(z); \mathrm{SA}^2(z); :::; \mathrm{SA}^{N_{head}}(z)]W_{MSA} \tag{5}$$

Inspired by its success in sequence-to-sequence modelling, researchers have applied transformer architecture to learn useful representations from images [7] [58] [59] [6]. Here, we briefly review some preliminaries associated with Video Transformers (ViT) [7]. A standard transformer for NLP receives a sequence of token embeddings as input. To deal with images, ViT reshapes an image $img \in \mathrm{R}^{H\times W\times C}$ into a sequence of flattened patches $img_p \in \mathrm{R}^{N\times(P^2\cdot C)}$, where (H, W) is the resolution of original image, C is the number of channels, (P, P) is the resolution of each image patch, and $N_p = HW/P^2$ is the number of patches. ViT computes patch embeddings by mapping flattened patches to D dimensions with a trainable linear projection:

$$z_0 = [z_{cls}; img_p^1 \mathrm{E}; img_p^2 \mathrm{E}; \ldots; img_p^{N_p} \mathrm{E}] + \mathrm{pos} \quad (6)$$

where the projection by E is equivalent to a 2D convolution. The class token $z_{cls} \in \mathbb{R}^{(P^2 \cdot C) \times D}$ is a trainable vector, which is inherited from BERT [56]. In order to retain positional information, a position embedding, $\mathrm{pos} \in \mathbb{R}^{(N_p+1) \times D}$, is appended to the patch embeddings. z_0 is passed through an encoder including a sequence of L transformer layers, and then fed to a linear classifier. The architecture of ViT enables the self-attention to spread information between the patch embeddings and the class token: during training the supervision signal comes only from the class token, while the patch embeddings are the model's only variable input.

B. Video Embedding

Thanks to its flexible architecture, a transformer can operate on any input sequence $z \in \mathbb{R}^{N \times D}$. Similar to ViT, we need to convert video input into sequence input for transformer layers. In the previous studies [60] [8], two strategies were adopted for tokenising videos. Both strategies map a video $V \in \mathbb{R}^{T \times H \times W \times C}$ to a sequence of tokens $\boldsymbol{z} \in \mathbb{R}^{n_t \times n_h \times n_w \times D}$, where T is the total number of frames. Then, $\boldsymbol{z}$ is combined with positional and class tokens, and reshaped into $\mathbb{R}^{N \times D}$ matrices to obtain the input sequence for transformer layers.

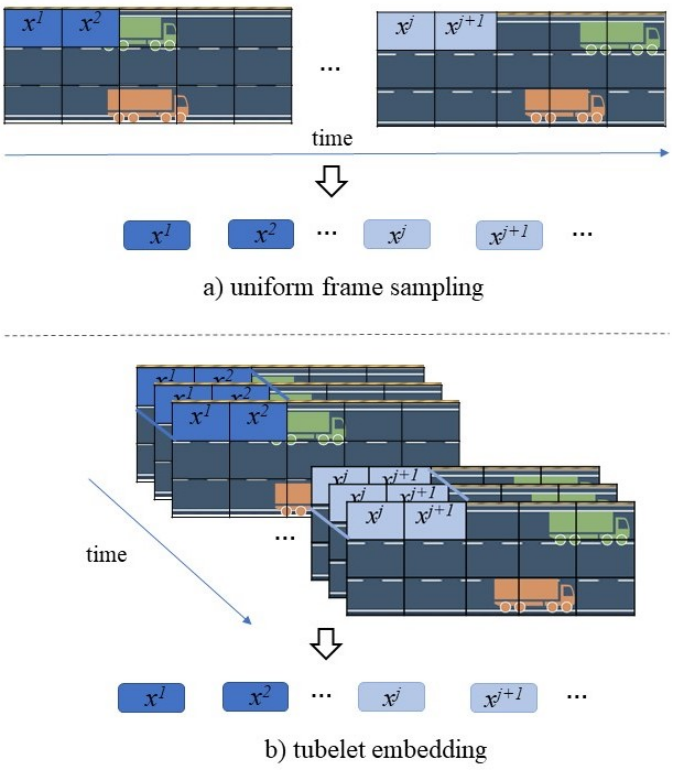


Fig. 2. Video embedding: a) uniform frame sampling, and b) tubelet embedding

As shown in Fig. 2a, a straightforward strategy for tokenising videos is to uniformly sample n_t frames from an input video clip, embed each frame independently, and concatenate all these tokens together. More specifically, n_h n_w image patches are extracted from each frame, and a total of n_t n_h n_w are obtained from a video clip. This process can be seen as simply constructing a large image to be tokenised following ViT. This strategy was adopted in Timesformer [60] which is also selected as a benchmark to compare with our proposed method in the case study section.

Fig. 2b illustrates a tubelet embedding strategy. Here, spatio-temporal "tubes" are extracted from an input video clip and linearly projected to $\mathbb{R}^D$. This strategy extents ViT's embedding to 3D, and therefore performs a 3D convolution. For a tubelet $\in \mathbb{R}^{t \times h \times w}$, $n_t = T/t$, $n_h = H/h$ and $n_w = W/w$ tokens are extracted from temporal, height and width dimensions respectively. This strategy fuses spatio-temporal dependencies during embedding, whereas the strategy illustrated in Fig 2a fuses temporal dependencies using transformer convolution. Note that the tubelet embedding strategy is used for video embedding in the proposed RViViT.

C. RViViT Architecture

Fig. 3 depicts the architecture of the proposed RViViT which consists of a video embedding layer, a spatial transformer encoder, a temporal transformer encoder and a regression head. The temporal encoder and spatial encoder have the same structure which contains a sequence of L transformer layers. Each layer has Multi-headed Self-Attention (MSA), Layer Normalisation (LN), and Multi-Layer Perception (MLP) blocks as follows:

$$z_l' = \mathrm{MSA}(\mathrm{LN}(z_{l-1})) + z_{l-1} \quad (7)$$

$$z_l = \mathrm{MLP}(\mathrm{LN}(z_l')) + z_l' \quad (8)$$

Firstly, the spatial transformer encoder captures dependencies among video embeddings (blue boxes in Fig.3) extracted from the same time step. Here, trainable position embeddings (grey boxes in Fig.3) are added to the video embeddings to retain positional information. This injects information about the relative position of the video embeddings in the sequence. A representation for each time step is obtained after L layers in the spatial encoder. This is the encoded regression token, z_s^i, where $i \in \{1, 2, \ldots, n_t\}$. These spatial representations are concatenated into $Z_s \in \mathbb{R}^{n_t \times D}$, and then fed to the temporal encoder containing L transformer layers to learn features among tokens from different time steps. The original ViViT uses a classification head to perform video classification tasks. In this research, we replace the classification head with a linear regression head. Using the encoded regression token, $z_{rgr} \in \mathbb{R}^D$, from the temporal transformer encoder, the regression head generates continuous traffic state estimations.

V. Case Study

In this section, traffic estimation experiments conducted on a public real-world vehicle trajectory dataset and corresponding analyses of the results are presented.

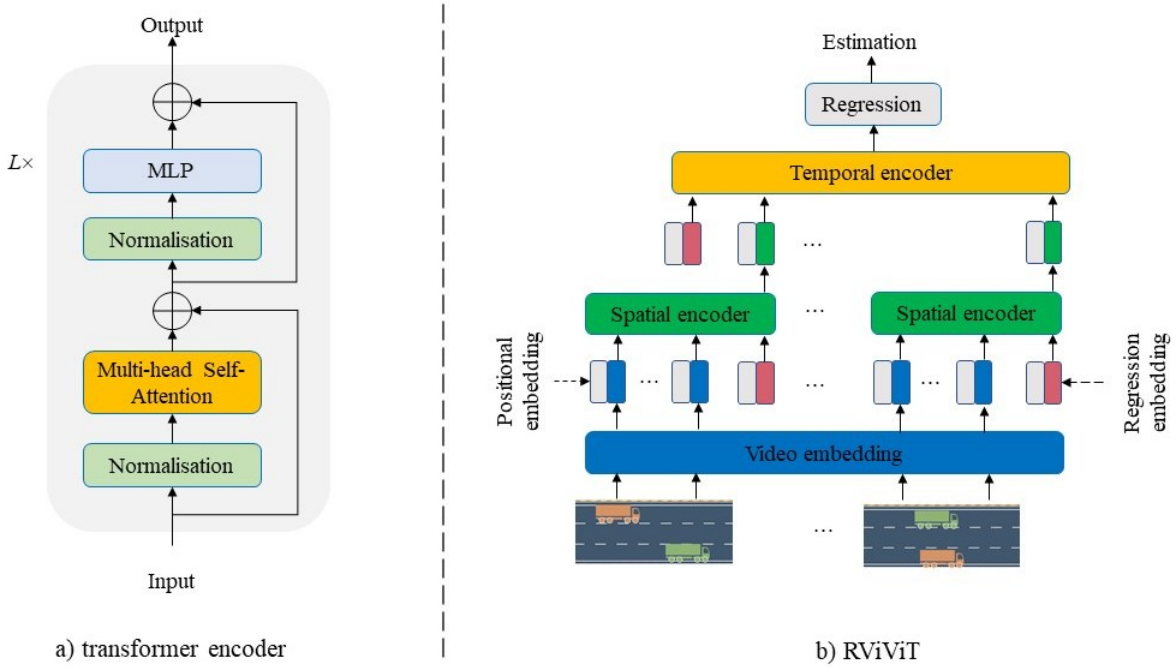


Fig. 3. RViViT architecture: a) transformer encoder, and b) the proposed RViViT

A. Data Preprocessing

We evaluated the proposed RViViT against a public dataset of naturalistic vehicle trajectories - the Highway Drone (HighD) dataset [62]. The HighD dataset provides post-processed trajectories of about 110000 cars and trucks extracted from drone videos of German highways around Cologne during 2017 and 2018. 60 videos were recorded with an average length of 17 minutes at six different sites (see Fig. 4a). For each video recording, three files are provided, which contain information about the site, the vehicles (e.g., type and size) and the extracted trajectories (e.g., velocity and acceleration).

Fig. 4b shows the aerial shot of the selected site which is a road segment of about 420-meter length. This site is numbered 1 in the dataset and has the most trajectory data (85972 trajectories) among all the six sites (110000 trajectories). The number of recorded car and truck trajectories are 69751 and 16221 respectively. Using the truck trajectories, we generated 39851 1-second video clips. Each video contains 25 frames and two channels (vehicle velocity and acceleration). The size of a road cell was set to 2 meters 1 lane. Thus, the width and height of a video is 210 and 3, respectively. Then, we calculated the corresponding space occupancy of the site for every video clip using both car and truck trajectories. Finally, these video clips were divided into training set, validation set and test set with proportions of 60%, 20% and 20%, respectively.

B. Experiment Setting

In this study, all experiments were conducted on GPUs in Google Colaboratory [63]. The patch (tubelet) size was set to 3 3 3. The number of self-attention heads N_{head}, the number of transformer layers L, and the embedding dimension D were set to 4, 6 and 128, respectively. We trained the RViViT using the AdamW optimizer [64] with a batch size of 32 and Mean Square Error (MSE) loss function. The learning rate was varied over the course of training using a linear warm-up with cosine annealing scheduler. More specifically, we increased the learning rate to 0.1 linearly for the 10 first warm-up epochs, and then decreased it to 0.0001 in the following 90 epochs. We also used early-stop and dropout (dropout rate=0.3) in all the experiments to prevent overfitting. A detailed discussion about hyperarameter settings is provided in the following section.

C. Result Analysis

Several spatio-temporal feature learning models were selected to compare with the proposed RViViT. They are:

- 3DCNN (3D Convolutional Neural Network) [65] that extracts features from both spatial and temporal dimensions by performing 3D convolutions;
- ConvLSTM (Convolutional Long Short-Term Memory) [66] that extends the LSTM to have convolutional structures in both the input-to-state and state-to-state transitions;
- SlowFast network [67] that includes a slow pathway to capture spatial semantics, and a fast pathway to capture motion at fine temporal resolution;
- TimeSformer (Time-Space Transformer) [60] that uses the uniform frame sampling described in Section 4.2 to convert input video into a sequence of image patches,

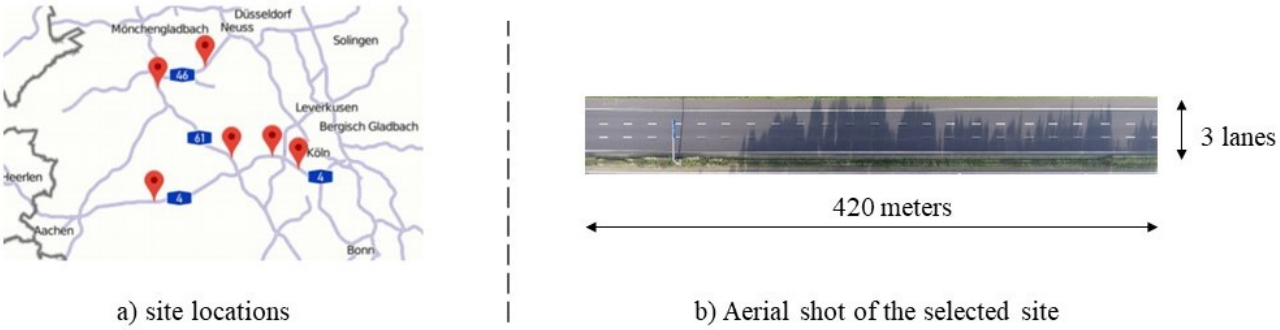


Fig. 4. Recording sites in HighD [62]: a) site locations, and b) aerial shot of the site selected to analyze

and applies MSA mechanism to model spatio-temporal dependencies.

- Space mean speed (SMS) based estimation [68] that employs Van Aerde's traffic flow model [69] to extract traffic state from SMS. Model-based traffic estimation methods that require additional spacing/headway measurement (e.g., [70]) or stationary sensor measurement (e.g., [71]) are not considered. As the proposed RViViT only uses trajectories with instant position, acceeleration and velocity as input.

TABLE I
COMPARISON WITH SPATIO-TEMPORAL LEARNING BENCHMARKS

Model	MAE	MSE	MAPE (%)
3DCNN	2.29	7.65	14.60
ConvLSTM	2.07	6.74	12.98
SMS Estimation	1.90	5.98	11.65
SlowFast Network	1.68	5.17	10.28
TimeSformer	1.46	3.43	8.93
RViViT	**1.43**	**3.35**	**8.41**

Table 1 summarizes the estimation results of the RViViT and aforementioned benchmarks on the HighD dataset. Note that the evaluation metrics, Mean Absolute Error (MAE), Mean Square Error (MSE) and Mean Absolute Percentage Error (MAPE) were calculated using the estimated and observed space occupancy (in %) values:

$$\text{MAE} = \sum_{c=1}^{C} |O^c_{est:} - O^c_{obs:}| \tag{9}$$

$$\text{MSE} = \sum_{c=1}^{C} (O^c_{est:} - O^c_{obs:})^2 \tag{10}$$

$$\text{MAPE} = \frac{1}{C} \sum_{c=1}^{C} \frac{|O^c_{est:} - O^c_{obs:}|}{O^c_{obs:}} \tag{11}$$

where, $O^c_{est:}$ and $O^c_{obs:}$ are the estimated and observed occupancy values of the c^{th} video clip (1-second long), respectively; and C is the total number of video clips in the test dataset.

The results shows that two CNN-based models, i.e., the 3DCNN and ConvLSTM, yielded relatively high estimation errors. The traffic flow model based method (SMS estimation) produced estimation that was better than the 3DCNN and ConvLSTM, but worse than the SlowFast network. Although the SlowFast network is also built based on CNN, it produced much lower MAE and RMSE values compared with the 3DCNN and ConvLSTM methods. This might be attributed to its two-path architecture that is able to learn video features at both low and high frame rates. The RViViT outperformed all the tested spatio-temporal learning models. It should be noted that another transformer-based model, the TimeSformer, produced estimation results close to that of the RViViT, which demonstrates the effectiveness of transformer-like architectures in dealing with spatio-temporal data. When compared with the TimSformer, the better performance yielded by the RViViT might be due to two reasons. Firstly, the RViViT employs a tubelet embedding strategy that fuses spatio-temporal information during tokenisation, in contrast to uniform frame sampling adopted by the TimeSformer, which may be seen as simply constructing a large 2D image to be tokenised following ViT. Secondly, the TimeSformer simply forwards all embeddings through the transformer encoder; whereas the RViViT consists of two separate transformer encoders (spatial encoder and temporal encoder) and perform a "late-fusion" of temporal information.

Fig. 5 shows the estimated and observed occupancy values from a randomly selected recording which is 611-second-long and numbered 11 in the dataset. It can be observed that the 3DCNN and ConvLSTM only modelled the rough trend of the traffic flow evolution. There were obvious deviations between the ground truth and the estimations from these two models. Although the SlowFast network showed improved performance compared to the other two CNN-based models, it still had some difficulties in modelling variations of the traffic flow. For example, when the traffic flow oscillated seriously around 400-500 seconds, the estimations lagged

behind the ground truth, and obvious over-estimations were witnessed. Two transformer-based models produced relatively reliable estimations. Despite slight lags, they showed the capacity of modelling general trend and specific variations of the traffic flow evolution.

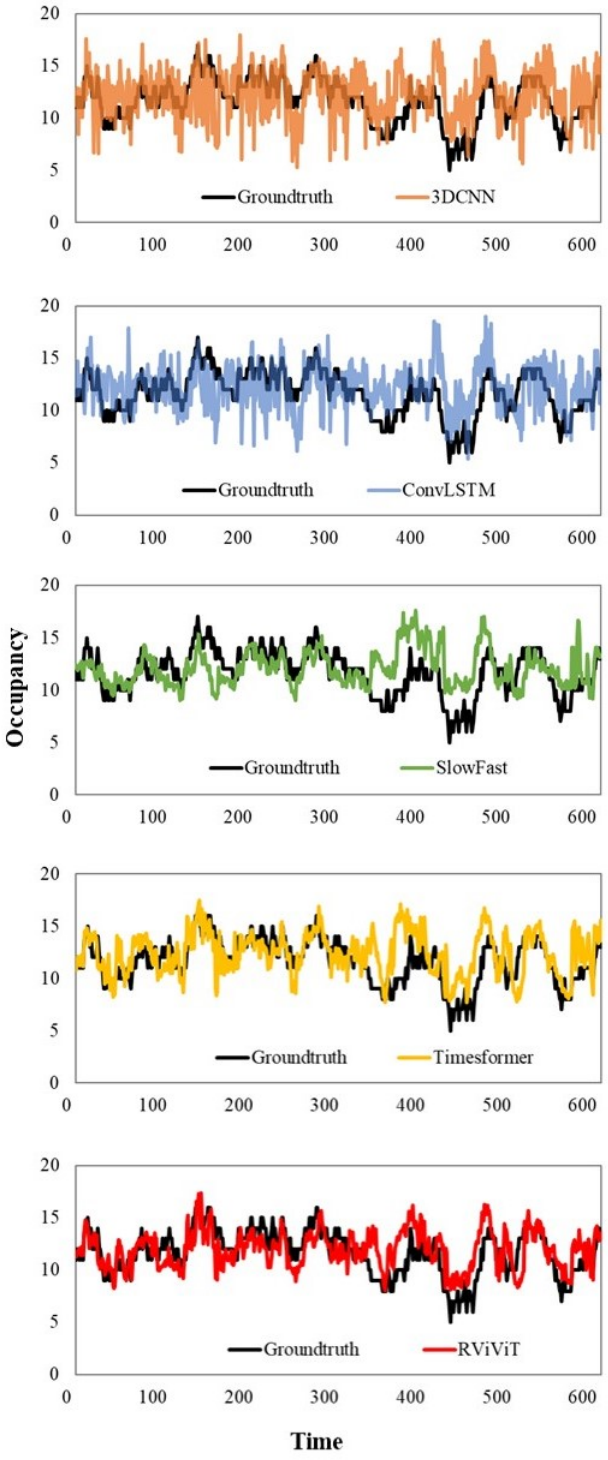


Fig. 5. Estimated and observed occupancy values from recording #11

Input video resolution plays a important role in video computing problems. Therefore, we analyzed different spatial discretization and temporal sampling combinations, as presented in Table 2. For spatial discretization, we kept the default height of a road cell, and changed its width. The results indicate that spatial discretization using smaller cell sizes resulted in better estimation. For temporal sampling, we first kept the default video length (25 frames), and sampled frames from 1 second (25 frame/second), 5 seconds (5 frame/second) and 15 seconds (3 frame/second). The results reveal that the RViViT achieved lower MSE when learning from shorter videos. Then, we sampled frames from the same second using 1, 3 and 24 strides to generate input videos with 25, 9, and 2 frames, respectively. It is observed that the differences among estimations were not significant. This suggests that sampling frames from the same period with a large stride might be a potential way to balance computational cost and model performance.

To investigate the flexibility of the RViViT, we conducted experiments on different dataset sizes and model settings. As shown in Fig.6a, we tested three parameter settings, including 1) less complex settings: D=32, L=3 and N_{head} = 2, 2) default settings: D=128, L=6 and N_{head} = 4, and 3) more complex settings: D=192, L=10 and N_{head} = 6. Floating Point Operations (FLOPs) were used to measure the computational costs required by these settings. It is seen that applying the more complex settings led to an approximate 5-time computational cost increase, but only resulted in a slight improvement in estimation when compared with the default settings. The less complex case requiring only around 1=18 computational cost of the default case can still provide acceptable estimation. In Fig. 6b, the estimation results on different dataset sizes are presented. Here, small, medium, and full datasets contain 1000, 10000, and 39851 video clips, respectively. As expected, we can improve the performance of our model by feeding it with more data. However, the RViViT can produce reliable estimation by just learning from the medium-size dataset.

Fig.7 shows the impact of probe vehicle composition and probe penetration rate on estimation. Note that in random vehicle cases, we randomly selected trucks and cars as probe vehicles; the number of random probe vehicles is the same to its truck-only counterpart. It is clearly shown that the estimation performance improved with the increase in the penetration rate of the probe vehicles that provide their trajectory information. Extracting trajectory information from both trucks and cars resulted in slightly worse estimation results than the cases that were solely based on truck trajectories. An explanation can be that although trucks tend to maintain lower speeds and have lager vehicle sizes in comparison to passenger cars, which may record a lower average speed and a higher occupancy for the segment than its mean, this pattern is easier to be learned by the model when only trucks provide information.

VI. Conclusion

Stationary sensing networks are often associated with high maintenance costs. Using instant vehicle information from

TABLE II
THE IMPACT OF VIDEO RESOLUTION ON ESTIMATION

Cell Width	MSE	Sampling Range	MSE	Video Length	MSE
2-meter	3.35	1-second	3.35	25-frame	3.35
5-meter	3.53	5-second	3.98	9-frame	3.43
10-meter	3.70	15-second	4.29	2-frame	3.69

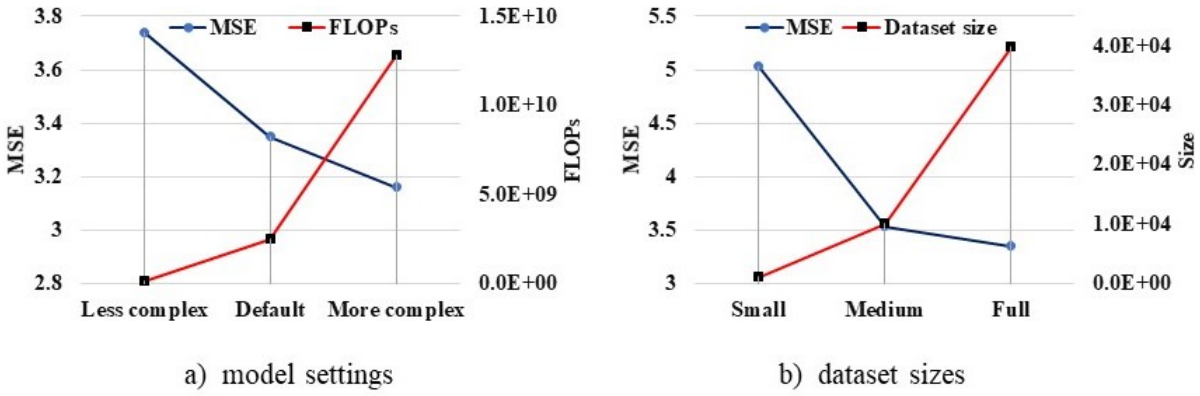


Fig. 6. Experiments on different a) model settings and b) dataset sizes

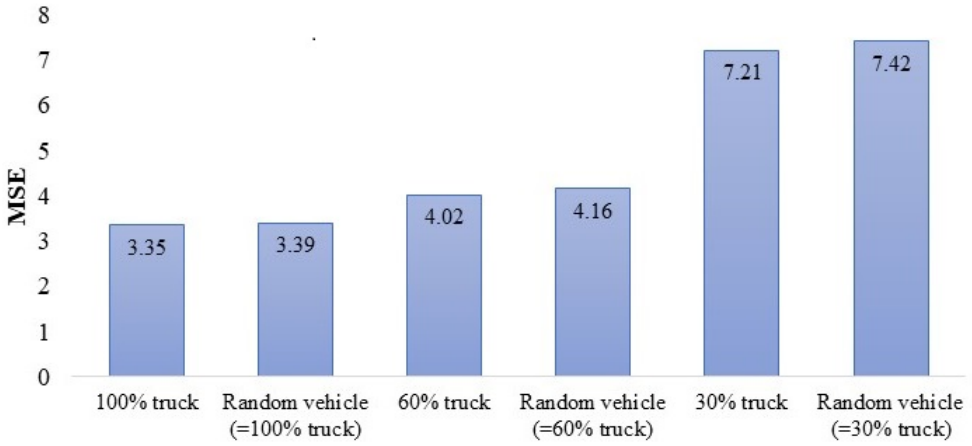


Fig. 7. The impact of probe penetration rate on estimation

GPS devices is a viable alternative for traffic monitoring. In this work, we first formulated trajectory-based traffic estimation as a video computing problem. Next, we reconstructed trajectory series into video-like data by performing spatial discretization. Following this, video input was embedded using a tubelet embedding strategy. Finally, a Revised Video Vision Transformer (RViViT) was proposed to estimate traffic state from video embeddings.

We tested the proposed RViViT on a public dataset of naturalistic vehicle trajectories. Four spatio-temporal deep learning models were chosen to compared with the proposed method, namely, 3DCNN, ConvLSTM, SlowFast network and TimeSformer. The results showed that the RViViT outperformed all these models. We further analyzed the performance of the proposed method on different video resolutions, model complexities and training data sizes. The analysis results revealed that increasing video resolution, model parameter and training data size had positive impacts on model performance. Nevertheless, even using relatively lower resolution, fewer parameters, and smaller training data size, the RViViT was able to yield acceptable estimation performance.

It should be noted that the proposed architecture can also be applied to traffic prediction. Since recordings in the HighD dataset were made with an average length of 17 minutes, we cannot generate sufficient data to train and test a prediction

model. We will test the prediction performance of the proposed RViViT once a suitable dataset is available. In addition, the RViViT is designed for estimating traffic state of a single road segment. In the future, we will extend this work to multi-segment traffic estimation.

ACKNOWLEDGMENT

This study was sponsored by the Engineering and Physical Sciences Research Council (EPSRC) (Project No.EP/R035199/1).

REFERENCES

[1] R. C. Lanctot, "A 5g perspective on connecting cars," in *5GAA C-V2X Workshop and Demonstration for North American Transportation Planning and Road Operator Communities*, 2018.

[2] T. Navman, "9 must-know stats on the state of fleet management," https://www.teletracnavman.com/resources/blog/9-must-know-stats-on-the-state-of-fleet-management, accessed: 2021-07-20.

[3] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[4] M. Ott, S. Edunov, D. Grangier, and M. Auli, "Scaling neural machine translation," *arXiv preprint arXiv:1806.00187*, 2018.

[5] M. X. Chen, O. Firat, A. Bapna, M. Johnson, W. Macherey, G. Foster, L. Jones, N. Parmar, M. Schuster, Z. Chen *et al.*, "The best of both worlds: Combining recent advances in neural machine translation," *arXiv preprint arXiv:1804.09849*, 2018.

[6] D. Zhou, B. Kang, X. Jin, L. Yang, X. Lian, Z. Jiang, Q. Hou, and J. Feng, "Deepvit: Towards deeper vision transformer," *arXiv preprint arXiv:2103.11886*, 2021.

[7] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[8] A. Arnab, M. Dehghani, G. Heigold, C. Sun, M. Lučić, and C. Schmid, "Vivit: A video vision transformer," *arXiv preprint arXiv:2103.15691*, 2021.

[9] T. Seo, Y. Kawasaki, T. Kusakabe, and Y. Asakura, "Fundamental diagram estimation by using trajectories of probe vehicles," *Transportation Research Part B: Methodological*, vol. 122, pp. 40–56, 2019.

[10] Z. Sun, W.-L. Jin, and S. G. Ritchie, "Simultaneous estimation of states and parameters in newell's simplified kinematic wave model with eulerian and lagrangian traffic data," *Transportation research part B: methodological*, vol. 104, pp. 106–122, 2017.

[11] V. L. Knoop and W. Daamen, "Automatic fitting procedure for the fundamental diagram," *Transportmetrica B: Transport Dynamics*, vol. 5, no. 2, pp. 129–144, 2017.

[12] T. Seo, T. Kusakabe, and Y. Asakura, "Traffic state estimation with the advanced probe vehicles using data assimilation," in *2015 IEEE 18th International Conference on Intelligent Transportation Systems*. IEEE, 2015, pp. 824–830.

[13] M. J. Lighthill and G. B. Whitham, "On kinematic waves ii. a theory of traffic flow on long crowded roads," *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, vol. 229, no. 1178, pp. 317–345, 1955.

[14] P. I. Richards, "Shock waves on the highway," *Operations research*, vol. 4, no. 1, pp. 42–51, 1956.

[15] H. J. Payne, "Freflo: A macroscopic simulation model of freeway traffic," *Transportation Research Record*, no. 722, 1979.

[16] G. B. Whitham, *Linear and nonlinear waves*. John Wiley & Sons, 2011, vol. 42.

[17] 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.

[18] 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.

[19] H. Haj-Salem and J. Lebacque, "Reconstruction of false and missing data with first-order traffic flow model," *Transportation Research Record*, vol. 1802, no. 1, pp. 155–165, 2002.

[20] L. Muñoz, X. Sun, R. Horowitz, and L. Alvarez, "Traffic density estimation with the cell transmission model," in *Proceedings of the 2003 American Control Conference, 2003.*, vol. 5. IEEE, 2003, pp. 3750–3755.

[21] C. G. Claudel and A. M. Bayen, "Guaranteed bounds for traffic flow parameters estimation using mixed lagrangian-eulerian sensing," in *2008 46th Annual Allerton Conference on Communication, Control, and Computing*. IEEE, 2008, pp. 636–645.

[22] Y. Yuan, J. Van Lint, R. E. Wilson, F. van Wageningen-Kessels, and S. P. Hoogendoorn, "Real-time lagrangian traffic state estimator for freeways," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 1, pp. 59–70, 2012.

[23] Y. Wang, M. Papageorgiou, and A. Messmer, "Renaissance–a unified macroscopic model-based approach to real-time freeway network traffic surveillance," *Transportation Research Part C: Emerging Technologies*, vol. 14, no. 3, pp. 190–212, 2006.

[24] M. Treiber and D. Helbing, "Reconstructing the spatio-temporal traffic dynamics from stationary detector data," *Cooper@ tive Tr@ nsport@ tion Dyn@ mics*, vol. 1, no. 3, pp. 3–1, 2002.

[25] Y. Zhao, S. Yin, D. Li, Q. Yu, and P. Ranjitkar, "Improving motorway mobility and environmental performance via vehicle trajectory data-based control," *IEEE Access*, vol. 8, pp. 86 862–86 869, 2020.

[26] D. Li, X. Zhao, and P. Cao, "An enhanced motorway control system for mixed manual/automated traffic flow," *IEEE Systems Journal*, vol. 14, no. 4, pp. 4726–4734, 2020.

[27] D. Li and P. Wagner, "A novel approach for mixed manual/connected automated freeway traffic management," *Sensors*, vol. 20, no. 6, p. 1757, 2020.

[28] D. Chen, S. Ahn, Z. Zheng, and J. Laval, "Traffic hysteresis and the evolution of stop-and-go oscillations," in *Transportation Research Board 92nd Annual Meeting*, 2013.

[29] B. L. Smith, W. T. Scherer, and J. H. Conklin, "Exploring imputation techniques for missing data in transportation management systems," *Transportation Research Record*, vol. 1836, no. 1, pp. 132–142, 2003.

[30] C. Chen, J. Kwon, J. Rice, A. Skabardonis, and P. Varaiya, "Detecting errors and imputing missing data for single-loop surveillance systems," *Transportation Research Record*, vol. 1855, no. 1, pp. 160–167, 2003.

[31] M. Zhong, P. Lingras, and S. Sharma, "Estimation of missing traffic counts using factor, genetic, neural, and regression techniques," *Transportation Research Part C: Emerging Technologies*, vol. 12, no. 2, pp. 139–166, 2004.

[32] D. Ni and J. D. Leonard, "Markov chain monte carlo multiple imputation using bayesian networks for incomplete intelligent transportation systems data," *Transportation research record*, vol. 1935, no. 1, pp. 57–67, 2005.

[33] W. Yin, P. Murray-Tuite, and H. Rakha, "Imputing erroneous data of single-station loop detectors for nonincident conditions: Comparison between temporal and spatial methods," *Journal of Intelligent Transportation Systems*, vol. 16, no. 3, pp. 159–176, 2012.

[34] J. Tang, G. Zhang, Y. Wang, H. Wang, and F. Liu, "A hybrid approach to integrate fuzzy c-means based imputation method with genetic algorithm for missing traffic volume data estimation," *Transportation Research Part C: Emerging Technologies*, vol. 51, pp. 29–40, 2015.

[35] S. Tak, S. Woo, and H. Yeo, "Data-driven imputation method for traffic data in sectional units of road links," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 6, pp. 1762–1771, 2016.

[36] L. Li, Y. Li, and Z. Li, "Efficient missing data imputing for traffic flow by considering temporal and spatial dependence," *Transportation research part C: emerging technologies*, vol. 34, pp. 108–120, 2013.

[37] H. Tan, Y. Wu, B. Cheng, W. Wang, and B. Ran, "Robust missing traffic flow imputation considering nonnegativity and road capacity," *Mathematical Problems in Engineering*, vol. 2014, 2014.

[38] H. Tan, G. Feng, J. Feng, W. Wang, Y.-J. Zhang, and F. Li, "A tensor-based method for missing traffic data completion," *Transportation Research Part C: Emerging Technologies*, vol. 28, pp. 15–27, 2013.

[39] N. Polson and V. Sokolov, "Bayesian particle tracking of traffic flows," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 2, pp. 345–356, 2017.

[40] Y. Lin, Y. Zhou, S. Yao, F. Ding, and P. Wang, "Real-time fine-grained freeway traffic state estimation under sparse observation," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2020, pp. 561–577.

[41] D. Li and J. Lasenby, "Spatiotemporal attention-based graph convolution network for segment-level traffic prediction," *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[42] T. Seo, A. M. Bayen, T. Kusakabe, and Y. Asakura, "Traffic state estimation on highway: A comprehensive survey," *Annual reviews in control*, vol. 43, pp. 128–151, 2017.

[43] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, pp. 1097–1105, 2012.