Uncertainty-Aware Temporal Graph Convolutional Network for Traffic Speed Forecasting

Weizhu Qian, Thomas Dyhre Nielsen, Yan Zhao, *Senior Member, IEEE*, Kim Guldstrand Larsen, and James Jianqiao Yu, *Senior Member, IEEE*

***Abstract*— Traffic speed forecasting has been a very active research area as it is essential for Intelligent Transportation Systems. Although a plethora of deep learning methods have been proposed for traffic speed forecasting, the majority of them can only make point-wise prediction, which may not provide enough information for critical real-world scenarios where prediction confidence also need to be estimated, e.g., route planning for ambulances and rescue vehicles. To address this issue, we propose a novel uncertainty-aware deep learning method coined Uncertainty-Aware Temporal Graph Convolutional Network (UAT-GCN). UAT-GCN employs a Graph Convolutional Network and Gated Recurrent Unit based architecture to capture spatio-temporal dependencies. In addition, UAT-GCN consists of a specialized regressor for estimating both epistemic (model-related) and aleatoric (data-related) uncertainty. In particular, UAT-GCN utilizes Monte Carlo dropout and predictive variances to estimate epistemic and aleatoric uncertainty, respectively. In addition, we also consider the recursive dependency between predictions to further improve the forecasting performance. An extensive empirical study with real datasets offers evidence that the proposed model is capable of advancing current state-of-the-arts in terms of point-wise forecasting and quantifying prediction uncertainty with high reliability. The obtained results suggest that, compared to existing methods, the RMSE and MAE of the proposed model on the SZ-taxi dataset are reduced by 2.15% and 7.23%, respectively; the RMSE and MAE of the proposed model on the Los-loop dataset are reduced by 4.17% and 8.53%, respectively.**

***Index Terms*— Traffic speed forecasting, graph convolutional network, gated recurrent unit, spatio-temporal model, uncertainty quantification.**

I. Introduction

INTELLIGENT transportation systems (ITS) are among the core components of modern smart cities, playing a critical role in our daily life.

An essential ITS element is forecasting of traffic speeds, providing assistance to authorities for managing traffic flow and avoiding potential congestion. Recent advances in the domain of deep learning have incentivized the emergence of new methods in traffic speed forecasting [1], [2], [3], [4].

Generally, traffic speed data come as time series, in which the temporal dependency cannot be neglected when predicting future states based on previous states. Furthermore, since roads in transportation systems interact with each other, the spatial relationships are also of utmost importance. In this sense, traffic speed forecasting can be regarded as a multivariate spatio-temporal prediction problem, which needs to be handled accordingly by capturing both spatial and temporal data correlation.

In terms of spatial dependency, the topological structure of a road network can be represented by a graph, in which each node denotes a road segment or traffic sensor, and the corresponding adjacency matrix represents the interactions among multiple segments. These spatial relationships can be captured by Graph Neural Network (GNN) [2]. As for the temporal dependency of traffic speed data, it can be intuitively modelled by a temporal deep learning models, such as Recurrent Neural Networks (RNNs), Convolutional Neural Networks (CNNs), and their variants [1], [2]. However, a particular challenge of modelling and forecasting traffic speed data is to account for spatial and temporal dependencies at the same time. To solve this issue, various spatio-temporal deep learning models have been proposed as will be described below.

Apart from the spatio-temporal correlations that are key for traffic speed prediction, it is often also necessary to not only obtain point estimates for the predictions, as in the majority of existing literature, but also credibility intervals for these predictions, enabling better better decision support and avoiding catastrophic mismanagement. For example, when ambulances and rescue vehicles choose their routes, the drivers may consider different routes balancing expected travel time with the uncertainty associated with the arrival time. In addition, models taking uncertainty into account may have better performance when considering robustness and generalization ability in contrast to deterministic approaches. Uncertainty quantification (UQ) has been a highly popular topic in the field of deep learning [5], however, although a variety of deep learning approaches have been proposed for traffic prediction, uncertainty quantification is an often ignored critical component in this context.

Uncertainty is typically classified into two categories, namely, aleatoric and epistemic uncertainty. In many real-world scenarios, the uncertainty inherent in observations (e.g., sensor noise) is referred to as aleatoric uncertainty, which is

Manuscript received 1 April 2022; revised 14 December 2022, 1 July 2023, and 26 January 2024; accepted 5 February 2024. The Associate Editor for this article was M. Wang. *(Corresponding author: Weizhu Qian.)*

Weizhu Qian is with the School of Computer Science and Technology, Soochow University, Suzhou 215006, China, and also with the Department of Computer Science, Aalborg University, 9220 Aalborg, Denmark (e-mail: wzqian@suda.edu.cn).

Thomas Dyhre Nielsen, Yan Zhao, and Kim Guldstrand Larsen are with the Department of Computer Science, Aalborg University, 9220 Aalborg, Denmark (e-mail: tdn@cs.aau.dk; yanz@cs.aau.dk; kgl@cs.aau.dk).

James Jianqiao Yu is with the Department of Computer Science, University of York, YO10 5DD York, U.K. (e-mail: jqyu@ieee.org).

Digital Object Identifier 10.1109/TITS.2024.3365721

often inevitable. This type of uncertainty cannot be reduced, but can be modelled by probabilistic approaches by capturing, e.g., the mean and the variance of the predictions [6]. On the other hand, when relevant training data is limited, this typically translate into uncertainty about the (true) model structure that generated the data. This is also referred to as epistemic uncertainty. To reflect such uncertainty, one needs to find a distribution over models rather than training a single specific model. Bayesian approximation approaches are commonly adopted techniques, including Bayesian neural networks (BNNs) [7] and Gaussian processes (GPs) [8]. However, these two methods come with their own limitations. BNNs increase the number of model parameters and GPs tend to become highly computationally expensive when the training dataset is large. An alternative approach is Monte Carlo dropout [9], a scheme that is widely regarded as a simple yet efficient Bayesian approximation method. Uncertainty estimation has also recently been explored for spatio-temporal forecasting [10], but the analysis is limited by not considering sampling efficiency nor the recursive dependencies between prediction steps.

In this work, we propose a novel deep learning model for traffic speed prediction, which is capable of both capturing spatio-temporal dependencies and quantifying the prediction uncertainty. In our approach, we employ a Graph Convolutional Network (GCN) [11] for spatial correlation, a Gated Recurrent Unit (GRU) [12] for temporal dependencies and a specialized regression model to quantify both the aleatoric and epistemic uncertainty of the prediction. Finally, we analyze and compare our model on publicly available datasets, where the experimental results show the proposed model achieve better uncertainty-aware forecasting performance compared to current state-of-the-arts.

Our contributions in this work are summarised as follows:

- We propose a novel uncertainty-aware spatial-temporal deep learning model for traffic speed forecasting which can estimate both the epistemic and aleatoric uncertainty in an efficient manner.
- A specialized regressor is devised to enhance the sequential prediction performance by considering the recursive dependency between different prediction steps.
- An efficient sampling strategy is proposed to accelerate the uncertainty estimation process.
- Extensive experiments are conducted to compare the proposed approach with existing state-of-the-art methods. The results suggest the superiority of the proposed model.

The remainder of the paper is organized as follows. Section II surveys the related work, and Section III introduces preliminary concepts and the traffic speed forecasting problem. We then propose the Uncertainty-Aware Temporal Graph Convolutional Network in Section IV, followed by the empirical study in Section V. Section VI concludes the paper.

II. Related Work

In the literature, Graph Neural Networks (GNNs) are commonly used to capture the non-Euclidean information of traffic data [2]. As per previous discussions, traffic speed data are typically multivariate time series. Consequently, apart from modelling the spatial dependencies, we need to take the temporal dependencies into account as well.

For example, Li et al. proposed Diffusion Convolutional Recurrent Neural Network (DCRNN) [13], which utilizes the diffusion convolutional operations to handle directed graphs and Gated Recurrent Units (GRUs) to model the temporal and spatial dependencies, respectively. Similarly, Temporal Graph Convolutional Networks (TGCNs) combine Graph Convolutional Networks and GRU components for traffic forecasting [14]. Alternatively, Wu et al. proposed Graph-Wavenet [15], which utilizes the WaveNet model [16] for capturing temporal information.

The attention Mechanism [17] has also been used to improve traffic forecasting performance. Guo et al. introduced the Attention-based Spatial-Temporal Graph Convolutional Network (ASTGCN) [18], in which spatial and temporal convolutions are combined with spatial and temporal attentions, respectively. Zheng et al. devised a spatio-temporal attention block which combines spatial and temporal attention mechanisms through gated fusion to build-based a graph multi-attention network for traffic forecasting [19]. Yu et al. proposed the Spatio-Temporal Graph Convolutional Networks (STGCN) [20], which uses spatial attention combined with Chebyshev polynomial approximation based spatial convolution with the temporal features being captured through Gated Convolution Neural Networks. Zhang et al. proposed Spatial-Temporal Graph Diffusion Network (ST-GDN) integrated with multi-scale self-attention networks in order to learn multi-level temporal contextual data [21].

Uncertainty quantification in deep learning has been a popular research topic in recent years [5]. Various deep learning-based uncertainty quantification methods have been successfully applied in a number of application domains, e.g., Computer Vision (CV) [6] and Reinforcement Learning (RL) [7]. More specifically, to estimate aleatoric uncertainty, methods including mean-variance estimation (MVE) [22], quantile regression [23], and conformal inference [24], [25] have been proposed. These methods can provide frequentist coverage for testing ground truth data with respect to certain pre-defined significance levels.

As for epistemic uncertainty estimation, there are mainly two classes of methods can be used. One is sampling and variational-based methods, which assume that the parameters of models are randomly distributed in weight space (e.g., Bayesian Neural Networks (BNNs) [7], [26], [27], [28], Deep Gaussian Processes (DGPs) [29], and Monte Carlo dropout [9]). The other is deep ensemble methods which average the predictions of a set of different models for inference [30], [31], [32], [33], [34]. Uncertainty quantification covering both aleatoric and epistemic uncertainties has drawn attention of researchers in ITS as well. For example, a Bayesian Graph Neural Network based approach [35] has been proposed for estimating travel time distributions. Uncertainty quantification is also important for traffic speed forecasting, however, it has not yet been well-explored. Therefore, the aim of this paper is to address this issue.

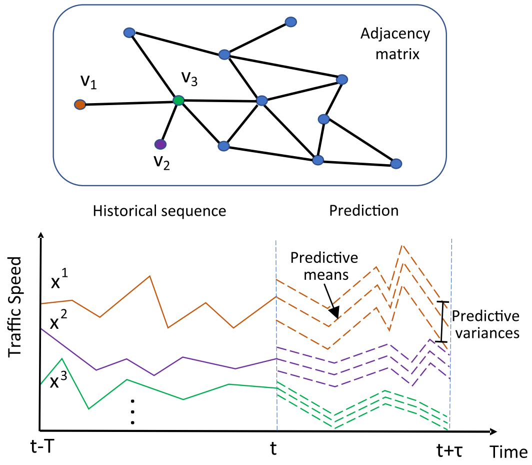


Fig. 1. The uncertainty-aware traffic speed forecasting problem.

III. Traffic Speed Forecasting With Uncertainty Quantification

In a road network, road segments commonly interact with others, thus the topology of a network with N segments can be represented by a graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, where the nodes $\mathcal{V} = \{v_1, v_2, v_3, \cdots, v_N\}$ represent road segments and the set of edges $\mathcal{E}$ represent the spatial interactions between the road segments. The corresponding adjacency matrix of $\mathcal{G}$ is $A \in \mathbb{R}^{N \times N}$. In this work, we focus on time-dependent forecasting of traffic speeds on road segments, but other types of traffic data, e.g., traffic flow, can also be adopted with minuscule modifications. For forecasting, we consider the mean traffic speed at road segment at discrete time points, hence the corresponding temporal traffic speed features for each road segment at time t can be represented by a vector $x_t = \{x_t^1, x_t^2, x_t^3, \cdots, x_t^N\}$.

The objective of traffic speed forecasting is to predict the future traffic speed based on the historical traffic speed, which can be regarded as a multivariate time series prediction problem. Let $X_{<t} = \{x_{t-T}, x_{t-T+1}, x_{t-T+2}, \cdots, x_{t-1}\}$ be the temporal features for every node from time point $t-T$ to $t-1$, where T is the length of the historical data input sequences. Accordingly, $X_{<t}$ and A are the input of the forecasting model. We employ $\hat{X}_{<t+\tau} = \{\hat{x}_t, \hat{x}_{t+1}, \hat{x}_{t+2}, \cdots, \hat{x}_{t+\tau-1}\}$ to denote the prediction of every node in the graph from time t to $t+\tau-1$, where τ is the predicted length.

The traffic speed forecasting problem is illustrated in Fig. 1. Specifically, we want to quantify the uncertainty of our predictions. Since the traffic speed values are continuous in our case, for the sake of simplicity, we assume that the final speed predictive distribution of each node at any particular time point, $P(\hat{X}_{<t+\tau} | X_{<t}, W, A)$, is Gaussian. The model we aim to learn can then be defined as follows:

$$\hat{X}_{<t+\tau} \sim \mathcal{N}(\mu_W(X_{<t}, A), \sigma_W(X_{<t}, A)), \tag{1}$$

where μ and σ are the mean and variance, respectively; W is the model parameters to be learned from the raw data. Note that the Gaussian likelihood assumption is relatively simple, however this assumption is computationally convenient for high dimensional time series forecasting tasks as in our case.

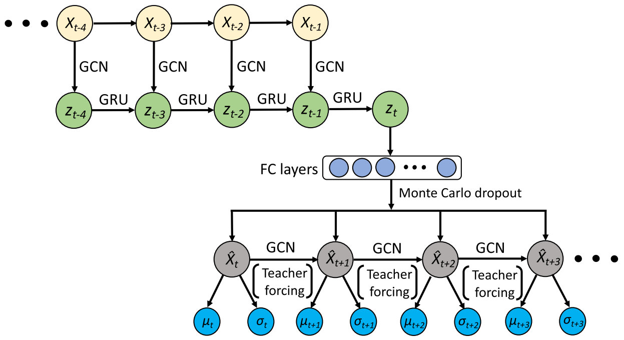


Fig. 2. The structure of the proposed model.

Moreover, in the later experimental section, we can see that the proposed model with the Gaussian likelihood works well in practice.

Finally, please note that the aleatoric and epistemic uncertainties are inevitably entangled during the model training process, especially when the data is very noisy. However, the causes of those two types uncertainties are essentially different. The aleatoric uncertainty represents the randomness of the data distribution, while the epistemic uncertainty represent the randomness of the model parameters distribution. In practice, to model the aleatoric uncertainty, we can first use the means and variances estimated independently by maximizing the corresponding likelihoods of the data distribution. To model the epistemic uncertainty, we sample multiple sets of the model parameters (which account for the randomness of the model parameters distribution), and use the means and variances of those Monte Carlo samples to represent the epistemic uncertainty.

IV. Uncertainty-Aware Temporal Graph Convolutional Network

In this section, we propose the Uncertainty-Aware Temporal Graph Convolutional Network (UAT-GCN) model for traffic speed forecasting with uncertainty quantification. The overall architecture of the proposed model is illustrated in Fig. 2. We first give an overview of the proposed method, and then elaborate the details. With a focus on modeling spatio-temporal dependencies and uncertainty quantification, we design the model by integrating three components: GCN layers that are used to capture the spatial relationship, GRU modules that are used to model the temporal dependency, and a specialized module that is used to quantify the uncertainty. Note that we choose to place the Monte Carlo dropout layer after the recurrent structure for two reasons: one is that dropout applied to the last few layers in our model can capture enough epistemic uncertainty in practice; the other is that such model structure design can accelerate the Monte Carlo simulation process during inference [6], [26], which is especially beneficial for the time series forecasting task in our case.

A. Graphical Convolutional Network

Generally, the output of a typical GNN layer — $H^{(l+1)}$ where l is the index of the hidden layers — can be

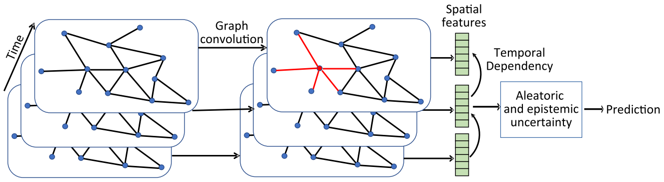


Fig. 3. The spatial-temporal dependency of traffic speed data.

described as

$$H^{(l+1)} = f(H^{(l)}, A), \tag{2}$$

where $H^{(0)} = X_t$. More specifically, the propagation rule is given as

$$f(H^{(l+1)}, A) = \sigma(AH^{(l)}W^{(l)}), \tag{3}$$

where $W^{(l)}$ is the weight matrix and σ is a nonlinear activation function, e.g., ReLU function.

However, there are two limitations that prevent direct usage of Eq. (3). The first is that we only sum up the neighboring nodes of a node without excluding the node itself, and the second one is that multiplication with the adjacent matrix A changes the scale of the feature vectors. To address these two issues, GCNs adopt the following propagation rule that adds an identity matrix to A to enforce self-loops in the graph and use symmetric normalization to normalize A:

$$f(H^{(l+1)}, A) = \sigma(\hat{D}^{-\frac{1}{2}}\hat{A}\hat{D}^{-\frac{1}{2}}, H^{(l)}, W^{(l)}), \tag{4}$$

where $\hat{D}$ is the diagonal node degree matrix of $\hat{A}$, $\hat{A} = A + I$, and I is the identity matrix. In our approach, the spatial relationships among road segments are modelled by a set of GCN layers [11] as shown in Fig. 3.

B. Gated Recurrent Unit

Traffic speed data are essentially time-series, thereby we forecast future traffic speeds by capturing the temporal dependency from the input historical traffic speed data. To this end, one can incorporate a recurrent module, e.g., a Recurrent Neural Network (RNN), into the model. However, as widely recognized, the vanilla RNN suffers from long-term time dependency issues [36]. Instead, Long Short-Term Memory (LSTM) [36] and Gated Recurrent Unit (GRU) [12] can be used to circumvent this problem. The difference between these two models is that, compared to the LSTM, the GRU model has similar performance but a lighter model structure. We therefore employ the GRU model in UAT-GCNs to represent the temporal relationships of traffic speed measurements.

It should be noted that the spatial information within traffic speed data is also important for traffic speed forecasting because it reflects interactions among different road segments. Consequently, as shown in Fig. 3, we combine the GRU module within the GCN layers as follows:

$$u_t = \sigma(W_u[f(H_t^l, A), h_{t-1}] + b_u), \tag{5a}$$

$$r_t = \sigma(W_r[f(H_t^l, A), h_{t-1}] + b_r), \tag{5b}$$

$$c_t = tanh(W_c[f(H_t^l, A), (r_t * h_{t-1})] + b_c), \tag{5c}$$

$$h_t = u_t * h_{t-1} + (1 - u_t) * c_t, \tag{5d}$$

where u is the output gate, r is the reset gate, c is the memory cell, h is the hidden state, W and b stand for the weights and bias, respectively.

C. Uncertainty Quantification

In UAT-GCN, we aim to quantify two sources of uncertainty, namely epistemic and aleatoric uncertainty. This also includes taking into account the recurrent dependency between multiple predictions from our time series problem.

1) Epistemic Uncertainty: Epistemic uncertainty is caused by model mis-specification or lack of domain data. Epistemic uncertainty can be reduced if the model is properly defined and training data is sufficient. Epistemic uncertainty can be modelled by various methods, e.g., Bayes By Backprop (BBB) [7], Gaussian Processes (GPs) [8], and Deep Ensembles [34]. However, these methods also exhibit certain shortcomings. For instance, BBB doubles the number of model parameters compared to standard neural networks. While GPs and Deep Ensembles can be computationally expensive when the dataset is large. Spare GPs can mitigate this issue but is too computationally complex [37]. For reasons of simplicity and accuracy, we choose to adopt a simple and flexible, and yet efficient method to model epistemic uncertainty in the model, namely, Monte Carlo dropout [9]. This method can be seen as an approximation to deep Gaussian process [29].

Dropout was originally proposed to prevent overfitting by turning off randomly selected neurons within layers during the training process [38]. By doing so, neural networks are able to learn a set of independent combinations of models to improve the generalization ability. This design principle is, to some extent, aligned with Bayesian approximation. In a Bayesian setting, in order to model epistemic uncertainty, we can put distributions over each weight parameter of the model. Accordingly, when we perform a prediction with a testing point x^*, the posterior predictive distribution is given by

$$p(y^*|x^*, D) = \int p(y^*|x^*, A, w)p(w|D)dw, \tag{6}$$

where y^* is the predictive mean, $D = \{X, Y, A\}$ is the training dataset, w is the model weights, $p(y^*|x^*, A, w)$ is the predictive likelihood, and $p(w|D)$ is the posterior of the weights.

Since the true posterior of the weights with respect to the given dataset $p(w|D)$ in Eq. (6) is typically intractable, it is approximated by a variational distribution $q(w)$ generated by Bernoulli dropout layers [9] in a Monte Carlo dropout approach:

$$q(W_i) = M_i \cdot (diag[z_{i,j}]_{j=1}^{K_i}), \tag{7a}$$

$$z_{i,j} \sim Bernoulli(p_i), \tag{7b}$$

where W_i denotes the masked weight matrices whose units are dropped binarily with respect to the predefined dropout probability p_i in the i-th layer and M_i are the parameters of the neural network. Consequently, the optimization objective is to minimize the Kullback-Leibler (KL) divergence

$D_{\mathrm{KL}}(q(w)||p(w|D))$, which is equivalent to minimizing the following loss function:

$$\begin{aligned}\mathcal{L}_{\text{Dropout}} &= \mathbb{E}_{w\sim q(w)}E[Y, f_W(X, A)] - D_{\mathrm{KL}}(q(w)||p(w)),\\ &\approx \frac{1}{N}\sum_{i=1}^{N} E(y_i, f(x_i, w_i, A)) + \frac{\lambda}{2p}||w||^2, \qquad (8)\end{aligned}$$

where N is the number of data points, E is the loss function, y_i is the target, and λ is the weight decay. The first term in Eq. (8) can be calculated by using the mean squared error (MSE) loss function for a regression task and the second term correspond to L2 regularization. We also need to be aware that the dropout rate p should not be too small, otherwise the model may lack variety. Meanwhile, the dropout rate should not be too large either due to the risk of under-fitting in this case.

At test time, as opposed to the standard dropout method, where dropout is turned off, in the Monte Carlo dropout method, dropout is still performed, we can therefore draw different prediction samples by performing a number of stochastic forward passes through the network and use the averaged results as the final predictions.

Note that inference via recurrent architectures can be more time-consuming than simple feedforward layers, and in real-world traffic speed forecasting scenarios, the inference time should not be too large. Hence, in the proposed model, we only perform Monte Carlo dropout on the regressor layers instead of the spatial and temporal feature extractors, which can reduce the inference time significantly while still maintaining forecasting performance.

2) Aleatoric Uncertainty: As opposed to epistemic uncertainty, aleatoric uncertainty cannot be reduced because it is caused by sensor noise or natural randomness. There are two types of aleatoric uncertainties, namely, homoscedastic and heteroscedastic, where the former is input-invariant and the latter is input-dependent. Here, we focus on heteroscedastic uncertainty. While aleatoric uncertainty cannot be reduced, it can be statistically learned. We address the issue of heteroscedastic uncertainty quantification following the approach proposed in [6].

To this end, we assume that the prediction for each node is a Gaussian distribution $\mathcal{N}(\mu, \sigma)$, whose mean μ and variance σ can be learned by two neural networks, $\mu_W(x_i, A)$ and $\sigma_W(x_i, A)$, respectively. However, it is hard to optimize the model parameters W directly according to the probability density function of the Gaussian distribution $\frac{1}{\sigma_W(x_i,A)\sqrt{2\pi}}e^{-\frac{1}{2}(\frac{y_i-\mu_W(x_i,A)}{\sigma_W(x_i,A)})^2}$. Alternatively, one can maximize the following log likelihood of the Gaussian distribution:

$$\begin{aligned}&\log\ p(y_i|\mu_W(x_i, A), \sigma_W(x_i, A))\\ &= \log(\frac{1}{\sigma_W(x_i, A)\sqrt{2\pi}}e^{-\frac{1}{2}(\frac{y_i-\mu_W(x_i,A)}{\sigma_W(x_i,A)})^2})\\ &= \log(\frac{1}{\sigma_W(x_i, A)\sqrt{2\pi}}) + \log(e^{-\frac{1}{2}(\frac{y_i-\mu_W(x_i,A)}{\sigma_W(x_i,A)})^2})\\ &= -\frac{1}{2}\log(\sigma_W(x_i, A)^2) - \frac{1}{2}\log(2\pi) - \frac{(y_i - \mu_W(x_i, A))^2}{2\sigma_W(x_i, A)^2}. \qquad (9)\end{aligned}$$

Consequently, we use the negative log likelihood as the loss function following Eq. (9). Compared to conventional neural network-based models using mean squared error (MSE) as the training loss without aleatoric uncertainty quantification, the training loss function regarding aleatoric uncertainty in our approach is

$$\begin{aligned}\mathcal{L}_{\text{Aleatoric}} = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{2\sigma(x_i, A)^2}||y_i - \mu(x_i, A)||^2\\ + \frac{1}{2}\log\sigma(x_i, A)^2. \qquad (10)\end{aligned}$$

If both epistemic and aleatoric uncertainty are taken into account, according to Eqs. (8) and (10), we have the following loss function:

$$\begin{aligned}\mathcal{L}_{\text{Combined}} = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{2\sigma(x_i, A)^2}||y_i - \mu(x_i, A)||^2\\ + \frac{1}{2}\log\sigma(x_i, A)^2 + \frac{\lambda}{2p}||w||^2. \qquad (11)\end{aligned}$$

D. Recursive Dependency

Since the output of the forecasting model is sequential, it is necessary to take into account the dependencies within the prediction sequences. In terms of multi-step traffic prediction, predictions on previous steps affect the subsequent steps. To this end, instead of using simple fully connected layers to compute the multi-step prediction directly, we design a specialized uncertainty-aware regressor to produce sequential output in a recursive manner.

Let τ be the prediction length. For taking previous predictions into account when calculating the uncertainty, we define

$$\hat{x}_{t+\tau} \sim p(\hat{x}_{t+\tau}|z_t, \hat{z}_{t+\tau-1}), \qquad (12)$$

where $\hat{x}_{t+\tau}$ is the prediction at the time point $t+\tau$, z_t is the hidden states learned from the historical input sequence x_t, $\hat{z}_{t+\tau-1}$ are the previously predicted hidden states:

$$\hat{z}_{t+\tau-1} = g_\omega(\hat{x}_{t+\tau-1}, A), \qquad (13)$$

where g_ω is a GCN layer parameterized by ω to capture the spatial relationship in $\hat{x}_t$. By combining Eqs. (12) and (13), we have the following regressor for computing the sequential predictions:

$$\begin{aligned}&\hat{x}_t \sim p(\hat{x}_t|z_t), &(14a)\\ &\hat{x}_{t+\tau} \sim p(\hat{x}_{t+\tau}|z_t, g_\omega(\hat{x}_{t+\tau-1}, A)) \text{ when } \tau > 1. &(14b)\end{aligned}$$

Note that when the forecasting step τ is 1, the recursive regressor is reduced to the regressor proposed in Section IV-C.

1) Teacher Forcing: During the training procedure, we can utilize the ground truth as the input to compute $\hat{z}_{t+T-1}$ in selective prediction steps with respect to the dropout probability to improve the training process. This method is referred to as teacher forcing [39]. When we use teacher forcing during training, the actual previous value are employed to guide the prediction process:

$$\hat{x}_{t+\tau} \sim p(\hat{x}_{t+\tau}|z_t, g_\omega(x_{t+\tau-1}, A)). \qquad (15)$$

Otherwise, we use the predicted previous values to compute the future steps:

$$\hat{x}_{t+\tau} \sim p(\hat{x}_{t+\tau}|z_t, g_\omega(\hat{x}_{t+\tau-1}, A)). \tag{16}$$

Naturally, at test time, teacher forcing is disabled because we do not have access to the ground truth anymore.

E. Offline Training and Online Inference

The training and inference procedures of our proposed method are summarized in Algorithms 1–3, respectively.

1) Training Procedure: Algorithm 1 presents the training procedure of the proposed method. At the first stage of training procedure, the input sequence x_t passes through the GCN layers g_θ along with the adjacent matrix A to obtain the spatial features z_t. Then we use the GRU module f_ϕ to calculate the future spatial state $z_{t+\tau}$. Afterwards, if we want to quantify the epistemic uncertainty, arbitrarily selected nodes of a set of fully connected layers f_ψ are dropped out with respect to a certain probability; if we want to quantify the aleatoric uncertainty as well, both the predictive mean $\mu_{t+\tau}$ and predictive variance $\sigma_{t+\tau}$ are computed, and accordingly $\mu_{t+\tau}$ is used as the predicted value of $\hat{x}_{t+\tau}$. Moreover, if the recursive dependency between the predictions is considered, we let the previous prediction $x_{t+\tau-1}$ pass through a GCN layer g_w to capture the spatial features, and use the conditional probability $p(\hat{x}_{t+\tau}|z_{t+\tau}, g_\omega(\mu_{t+\tau-1}, A))$ to obtain the prediction $\hat{x}_{t+\tau}$ when the teacher forcing technique is not adopted. Otherwise, we let the ground truth value $x_{t+\tau-1}$ pass through the GCN layer, and use the joint conditional probability $p(x_{t+\tau}|z_{t+\tau}, g_\omega(\mu_{t+\tau-1}, A))$ to obtain the prediction $\hat{x}_{t+\tau}$. Finally, the loss function is computed to optimize the parameters of the model according to the uncertainty type we choose.

Algorithm 1 Offline Model Training

Require: dataset $\{X\}$; model parameters $\theta, \phi, \psi, \omega$; predicting length T.

```
while epoch < max epoch:
    $z_t \leftarrow g_\theta(x_t, A)$; {via GCN}
    $z_{t+\tau} \leftarrow f_\phi(z_t)$; {via GRU}
    if epistemic uncertainty:
        Apply dropout to $f_\psi$;
    if aleatoric uncertainty:
        $\mu_{t+\tau}, \sigma_{t+\tau} \leftarrow f_\psi(z_{t+\tau-1})$;
    else:
        $\mu_{t+\tau} \leftarrow f_\psi(z_{t+\tau})$;
    if recursive dependency:
        if Teacher forcing:
            $\hat{x}_{t+\tau} \leftarrow p(\hat{x}_{t+\tau}|z_{t+\tau}, g_\omega(x_{t+\tau-1}, A))$;
        else:
            $\hat{x}_{t+\tau} \leftarrow p(\hat{x}_{t+\tau}|z_{t+\tau}, g_\omega(\mu_{t+\tau-1}, A))$;
    else:
        $\hat{x}_{<t+\tau} \leftarrow \mu_{t+\tau}$;
    Compute loss: Eq. 8 (epistemic uncertainty) or Eq. 10
    (aleatoric uncertainty) or Eq. 11 (combined uncertainty)
end while
```

2) Regular Inference: Algorithm 2 demonstrates the inference procedure without including epistemic uncertainty. In order to attain the spatial features z_t, we feed the input sequence x_t to the GCN layers g_θ along with the adjacent matrix A, and then the next spatial state $z_{t+\tau}$ is calculated via the GRU module f_ϕ. Afterwards, the predictive mean $\mu_{t+\tau}$ is used as the value of the prediction $\hat{x}_{t+\tau}$. Furthermore, if the recursive dependency between the predictions is considered, we let the previous prediction $x_{t+\tau-1}$ pass through the GCN layer g_w, and use the joint conditional probability $p(\hat{x}_{t+\tau}|z_t, g_\omega(\mu_{t+\tau-1}, A))$ to obtain the prediction $\hat{x}_{t+\tau}$.

Algorithm 2 Regular Online Model Inference

Require: dataset $\{X\}$; model parameters $\theta, \phi, \psi, \omega$; predicting length T.

```
while inferring:
    $z_t \leftarrow f_\theta(X_{<t}, A)$; {via GCN}
    $z_{t+\tau} \leftarrow f_\phi(z_\tau)$; {via GRU}
        if aleatoric uncertainty:
            $\mu_{t+\tau}, \sigma_{t+\tau} \leftarrow f_\psi(z_{t+\tau})$;
        else:
            $\mu_{t+\tau} \leftarrow f_\psi(z_{t+\tau})$;
        if recursive dependency:
            $\hat{x}_{t+\tau} \leftarrow p(\hat{x}_{t+\tau}|z_{t+\tau}, g_\omega(\mu_{t+\tau-1}, A))$;
        else:
            $\hat{x}_{t+\tau} \leftarrow \mu_{t+\tau}$.
end while
```

3) Monte Carlo Inference: Algorithm 3 demonstrates the inference procedure involving epistemic uncertainty. As opposed to the regular reference in Algorithm 2, we need to let the input sequence x_t pass through the model multiple times to do the Monte Carlo simulation. Since the Monte Carlo simulation is time-consuming, we can compute the future spatial state $\hat{z}_{t+\tau}$ in a deterministic manner through the GCN and GRU modules to accelerate inference. Then we can run the Monte Carlo simulation for N_{MC} times, each run of which generates the prediction following Algorithm 2. After finishing Monte Carlo simulation, we average the obtained Monte Carlo samples as the final prediction.

V. Experiments

To validate UAT-GCN, our proposed method is first compared with current state-of-the-art models in terms of point-wise speed forecasting performance. Subsequently, we verify the proposed model's capability of uncertainty quantification. Lastly, we analyze the computational cost of UAT-GCN.

A. Dataset Description

We employ two public traffic speed datasets, the SZ-taxi dataset and the Los-loop dataset [14], to compare the performance of UAT-GCN with existing methods. The SZ-taxi dataset consists of 156 road segments and the spatial interactions among those roads are represented by an 156 × 156 adjacency matrix. The temporal speed features of the SZ-taxi dataset are collected every 15min. The Los-loop

Algorithm 3 Monte Carlo Online Model Inference

Require: dataset $\{X\}$; model parameters θ, ϕ, ψ, ω; predicting length T; Monte Carlo sample number N_{MC}.

```
while inferring:
    $z_t \leftarrow g_\theta(X_{<t}, A)$; {via GCN}
    $z_{t+\tau} \leftarrow f_\phi(z_t)$; {via GRU}
    for $i$ in range($N_{MC}$) :
        Apply dropout to $f_\psi$;
        if aleatoric uncertainty:
            $\mu^i_{t+\tau}, \sigma^i_{t+\tau} \leftarrow f_\psi(z_{t+\tau})$;
        else:
            $\mu^i_{t+\tau} \leftarrow f_\psi(z_{t+\tau})$;
        if recursive dependency:
            $\hat{x}^i_{t+\tau} \leftarrow p(x^i_{t+\tau} | z_{t+\tau}, g_\omega(\mu^i_{t+\tau-1}, A))$;
        else:
            $\hat{x}^i_{t+\tau} \leftarrow \mu^i_{t+\tau}$;
    $\hat{x}_{t+\tau} \leftarrow \frac{1}{N_{MC}} \sum_{i=1}^{N_{MC}} \hat{x}^i_{t+\tau}$.
end while
```

dataset consists of 207 sensors, and the dimension of the adjacency matrix describing the spatial interactions among the sensors is 207 × 207. The temporal speed features of the Los-loop dataset are collected every 5min.

B. Experiment Setup

Our aim is to use 1 hour historical traffic speed recoding for forecasting the next 15min, 30min, 45min and 60min traffic speed, respectively. Therefore, for the SZ-taxi dataset, the input sequence length is 4, and the predicting sequence lengths are 1, 2, 3 and 4, respectively. For the Los-loop dataset, the input sequence length is 12, and the predicting sequence lengths are 3, 6, 9 and 12, respectively. The data of these two datasets are split into two parts, where 80% of the data are used for training and the remaining 20% are used for testing. In terms of data pre-processing, the input data are scaled into $[0, 1]$ by a min-max scaler to feed the models. The training batch size is 64; the optimizer we use is Adam with the learning rate of 0.001 and weight decay of 10^{-6}; the total training epoch is $3,000$; the dropout rate is 0.2; the number of the Monte Carlo samples for evaluating the epistemic uncertainty is 1000; the Teacher Forcing rate is 0.5, which determines the probability of performing the Teacher Forcing operation during training.

C. Models

1) Baseline Models: We use the following state-of-the-art methods as the baseline models for comparisons:

- Graph Convolutional Network (GCN) [11], which considers only the spatial dependency;
- Gated Recurrent Unit (GRU) [12], which considers only the temporal dependency;
- Temporal Graph Convolutional Network (TGCN) [14], which considers the spatio-temporal dependency by combining GCN and GRU;
- Graph-WaveNet (GWN) [15], which considers the spatio-temporal dependency by combining GCN and WaveNet [16];
- Spatio-Temporal Graph Convolutional Network (STGCN) [14], which considers the spatio-temporal dependency by combining GCN and GRU;
- Attention based Spatial-Temporal Graph Convolutional Network (ASTGCN) [18], which leverages attention mechanism to model spatio-temporal correlations for traffic prediction;
- Multivariate Time Series Forecasting with Graph Neural Networks (MTGNN) [40], which consists of a graph learning moduel, a mix-hop propagation layer, and a dilated inception layer.

2) Proposed Models: Besides the proposed UAT-GCN, we also develop alternative models for ablation tests and for comparison against the aforementioned baselines:

- UAT-GCN/E considers the spatio-temporal dependency as well as epistemic uncertainty;
- UAT-GCN/A considers the spatio-temporal dependency as well as aleatoric uncertainty;
- UAT-GCN/R additionally includes the recursive dependency of UAT-GCN.

D. Evaluation Metrics

We adopt the following metrics for evaluation:

1) Root Mean Squared Error (RMSE):

$$RMSE = \sqrt{\frac{1}{NM}\sum_{i=1}^{N}\sum_{j=1}^{M}(\hat{y}_i^j - y_i^j)^2}, \tag{17}$$

where N is the number of the datapoints and M is number of the road segments, y is the ground truth, and $\hat{y}$ is the prediction.

2) Mean Absolute Error (MAE):

$$MAE = \frac{1}{NM}\sum_{i=1}^{N}\sum_{j=1}^{M}\left|\hat{y}_i^j - y_i^j\right|. \tag{18}$$

3) Mean Absolute Percentage Error (MAPE):

$$MAPE = \frac{1}{NM}\sum_{i=1}^{N}\sum_{j=1}^{M}\frac{\left|\hat{y}_i^j - y_i^j\right|}{y_i^j}. \tag{19}$$

4) Accuracy:

$$Accuracy = 1 - \frac{||\hat{Y} - Y||_F}{Y_F}, \tag{20}$$

where Y denotes the set of y_i^j and $\hat{Y}$ denotes the set of $\hat{y}_i^j$.

5) Coefficient of Determination (R^2):

$$R^2 = 1 - \frac{\sum_{i=1}^{N}\sum_{j=1}^{M}(\hat{y}_i^j - y_i^j)^2}{\sum_{i=1}^{N}\sum_{j=1}^{M}(\hat{y}_i^j - \bar{Y})^2}, \tag{21}$$

where $\bar{Y}$ is the average of y_i^j.

TABLE I
PREDICTION RESULTS ON THE SZ-TAXI DATASET

Time	Metrics	GCN	GRU	TGCN	GWN	STGCN	ASTGCN	MTGNN	UAT-GCN/E	UAT-GCN/A	UAT-GCN
15 min	RMSE	5.9938	4.3331	4.0938	4.5739	**4.0373**	4.1513	4.0710	4.0421	4.1652	4.0394
	MAE	4.4408	3.0765	2.7452	3.2180	2.6946	2.7748	2.6747	2.7289	2.7827	**2.6271**
	Accuracy	0.5824	0.6981	0.7148	0.7111	0.7174	0.7108	0.7164	0.7184	0.7098	**0.7185**
	R^2	0.6704	0.8278	0.8463	0.8422	0.8497	0.8420	0.8480	0.8503	0.8411	**0.8505**
	Variance	0.6704	0.8283	0.8466	0.8423	0.8499	0.8422	0.8483	0.8503	0.8414	**0.8505**
	MAPE	26.23%	16.50%	16.54%	16.41%	15.66%	16.57%	16.30%	16.36%	17.06%	**15.62%**
30 min	RMSE	6.0116	4.4187	4.1007	4.6525	4.0772	4.2010	4.0784	4.0716	4.1092	**4.0449**
	MAE	4.4509	3.1903	2.7480	3.2829	2.7133	2.7605	2.6913	2.7432	2.6625	**2.6396**
	Accuracy	0.5812	0.6921	0.7143	0.7047	0.7149	0.7074	0.7159	0.7163	0.7137	**0.7181**
	R^2	0.6685	0.8210	0.8457	0.8351	0.8486	0.8382	0.8475	0.8481	0.8453	**0.8501**
	Variance	0.6686	0.8217	0.8457	0.8357	0.8470	0.8384	0.8479	0.8481	0.8453	**0.8501**
	MAPE	26.48%	15.87%	16.52%	17.85%	16.91%	16.70%	16.56%	16.16%	16.39%	**15.83%**
45 min	RMSE	6.0312	4.4910	4.1858	4.6923	4.1048	4.3833	4.1186	4.0717	4.1099	**4.0649**
	MAE	4.4668	3.2820	2.8710	3.3305	2.7403	2.9788	2.6867	2.7060	2.6851	**2.6587**
	Accuracy	0.5798	0.6871	0.7084	0.7003	0.7133	0.6946	0.7131	0.7163	0.7136	**0.7167**
	R^2	0.6663	0.8151	0.8393	0.8301	0.8450	0.8239	0.8446	0.8481	0.8452	**0.8486**
	Variance	0.6663	0.8160	0.8395	0.8313	0.8452	0.8245	0.8457	0.8481	0.8454	**0.8487**
	MAPE	26.88%	15.85%	16.55%	17.26%	**15.51%**	16.82%	16.88%	16.38%	16.47%	16.16%
60 min	RMSE	6.0491	4.5662	4.3451	4.7581	4.1320	4.2763	4.1307	4.0796	4.1486	**4.0615**
	MAE	4.4846	3.3732	3.0575	3.3373	2.7638	2.8366	2.6981	2.6944	2.7036	**2.6702**
	Accuracy	0.5786	0.6819	0.6973	0.6967	0.7117	0.7021	0.7123	0.7157	0.7109	**0.7170**
	R^2	0.6644	0.8090	0.8269	0.8260	0.8432	0.8324	0.8438	0.8475	0.8423	**0.8488**
	Variance	0.6644	0.8100	0.8274	0.8270	0.8434	0.8327	0.8451	0.8475	0.8423	**0.8489**
	MAPE	27.04%	**16.00%**	16.31%	16.51%	17.20%	17.17%	17.09%	16.46%	16.09%	16.14%

TABLE II
PREDICTION RESULTS ON THE LOS-LOOP DATASET

Time	Metrics	GCN	GRU	TGCN	GWN	STGCN	ASTGCN	MTGNN	UAT-GCN/E	UAT-GCN/A	UAT-GCN
15 min	RMSE	7.8473	5.6171	**5.1805**	5.1933	5.2465	5.5589	5.3376	5.2504	5.4742	5.2135
	MAE	5.4435	3.1726	3.2428	**2.9611**	3.0862	3.7132	3.6792	3.2328	3.4414	3.2932
	Accuracy	0.8664	0.9043	**0.9118**	0.9116	0.8572	0.9054	0.9091	0.9106	0.9068	0.9112
	R^2	0.6801	0.8365	**0.8605**	0.8599	0.8312	0.8414	0.8539	0.8567	0.8448	0.8593
	Variance	0.6805	0.8393	0.8607	0.8602	0.8589	0.8518	0.8653	0.8567	0.8479	**0.8628**
	MAPE	13.75%	10.64%	8.56%	8.00%	9.41%	8.60%	8.50%	8.58%	8.56%	**7.98%**
30 min	RMSE	8.3534	6.7081	6.1842	6.2508	6.1053	6.4483	5.9894	6.0453	6.5633	**5.9556**
	MAE	5.6882	3.7445	3.8341	3.6583	3.4912	3.6965	3.6649	**3.4814**	3.9377	3.4821
	Accuracy	0.8577	0.8857	0.8946	0.8936	0.8960	0.8902	0.8980	0.8971	0.8882	**0.8986**
	R^2	0.6386	0.7679	0.8019	0.7976	0.8070	0.7849	0.8143	0.8107	0.7773	**0.8164**
	Variance	0.6387	0.7712	0.8022	0.7977	0.8071	0.7858	0.8148	0.8108	0.7787	**0.8166**
	MAPE	14.95%	11.76%	10.52%	9.97%	10.77%	10.12%	9.71%	9.31%	10.59%	**9.24%**
45 min	RMSE	8.8399	7.5317	7.0629	6.8931	6.8479	7.3155	6.9778	6.7687	7.3122	**6.5104**
	MAE	5.8889	4.2619	4.3077	4.0723	3.8667	4.4506	4.9826	4.1869	4.3532	**3.8036**
	Accuracy	0.8494	0.8716	0.8796	0.8826	0.8833	0.8754	0.8811	0.8847	0.8754	**0.8891**
	R^2	0.5971	0.7079	0.7433	0.7553	0.7591	0.7245	0.7531	0.7650	0.7264	**0.7814**
	Variance	0.5977	0.7096	0.7459	0.7571	0.7627	0.7264	0.7667	0.7698	0.7326	**0.7819**
	MAPE	15.90%	12.32%	11.63%	13.05%	12.23%	11.90%	12.47%	10.82%	11.36%	**10.27%**
60 min	RMSE	9.2749	8.2474	7.2390	7.4930	7.4970	7.9033	7.0535	7.2177	7.3463	**6.9177**
	MAE	6.2764	4.7410	4.4560	4.3526	4.2483	4.3825	4.5642	4.4234	4.3875	**3.9112**
	Accuracy	0.8419	0.8594	0.8766	0.8723	0.8722	0.8653	0.8798	0.8770	0.8748	**0.8821**
	R^2	0.5574	0.6507	0.7308	0.7114	0.7134	0.6789	0.7449	0.7320	0.7230	**0.7540**
	Variance	0.5575	0.6520	0.7320	0.7120	0.7199	0.6795	0.7472	0.7320	0.7246	**0.7545**
	MAPE	17.30%	13.87%	12.46%	15.26%	13.45%	13.02%	12.50%	12.71%	12.08%	**11.25%**

6) Explained Variance Score (Variance):

$$Variance = 1 - \frac{Var(\hat{Y} - Y)}{Var(Y)}. \tag{22}$$

Smaller RMSE and MAE means better performance while larger accuracies, coefficients of determination and explained variance scores are better.

E. Experiment Results

1) Point-wise Forecasting Results: The prediction results on the SZ-Taxi and Los-loop datasets are reported in Table I and II, respectively. The best results are highlighted in bold. Some of the results are reported from [41].

According to the results demonstrated in Tables I and II, we can see that all spatio-temporal models, including TGCN, Graph-WaveNet, STGCN, UAT-GCN and its variants, outperform the spatio-only baseline GCN, and the temporal-only baseline GRU. This suggests that it is necessary to capture both the temporal and spatial features for traffic prediction. Additionally, the prediction results also indicate that our models capturing epistemic uncertainty are superior than other state-of-the-art methods. Furthermore, the performance of the proposed model can be further improved if we take both the epistemic uncertainty and the aleatoric uncertainty into account.

In addition, compared to T-GCN, the RMSE and MAE of the proposed model considering both epistemic and aleatoric uncertainty on the SZ-taxi dataset are reduced by **2.15**% and **7.23**%, respectively; the RMSE and MAE of the proposed model considering both the epistemic and aleatoric uncertainty on the Los-loop dataset are reduced by **4.17**% and **8.53**%, respectively. This is due to the fact that by capturing both data-related and the model-related uncertainty, UAT-GCN has better generalization ability compared to existing deterministic approaches. The main reason why the model's performance on the Los-loop dataset is better than on the SZ-taxi dataset is that the latter contains more noisy information than the former, which consequently makes the uncertainty quantification tasks more challenging.

2) Recursive Prediction Results: Table III demonstrates the prediction results when we consider the recursive dependency between the prediction steps. According to Table III, when the prediction step is 1, i.e., the forecasting horizons 15 min, UAT-GCN/R has the same the architecture as the vanilla UAT-GCN. When the prediction horizons get larger, e.g., 45 min, 60 min, 75 min, and 90 min, considering the recursive dependency further improves the prediction performance.

3) Uncertainty Quantification Results: For evaluation, Quantile Regression [23], Mean-Variance Estimation (MVE) [22], MC Dropout [9], and Deep Ensembling [30] are compared to the proposed methods. We use the mean test predictive intervals (PI), mean test predictive Gaussian negative log-likelihoods (NLL), and test Prediction Interval Coverage Probability (PICP) to evaluate the uncertainty quantification performance [42]. The predicted lower and upper bounds of the prediction interval are $\hat{y_L}$ and $\hat{y_U}$, and the significance level is set to 95%. Let k_i^j indicate whether the real speed value of a road segment j at time i is captured by the estimated prediction interval:

$$k_i^j = \begin{cases} 1, & \text{if } \hat{y_L}_i^j \le y_i^j \le \hat{y_U}_i^j \\ 0, & \text{else.} \end{cases} \tag{23}$$

The total number of the captured data points is

$$C = \sum_{i=1}^{N} \sum_{j=1}^{M} k_i^j, \tag{24}$$

hence the PICP can be defined as

$$PICP = \frac{C}{NM}. \tag{25}$$

TABLE III

PREDICTION RESULTS ON THE SZ-TAXI DATASET CONSIDERING RECURSIVE DEPENDENCY

Time	Metrics	UAT-GCN	UAT-GCN/R
15 min	RMSE	**4.0394**	**4.0394**
	MAE	**2.6271**	**2.6271**
	Accuracy	**0.7185**	**0.7185**
	R^2	**0.8505**	**0.8505**
	Variance	**0.8505**	**0.8505**
	MAPE	**15.62%**	**15.62%**
30 min	RMSE	**4.0449**	4.0588
	MAE	**2.6396**	2.6703
	Accuracy	**0.7181**	0.7172
	R^2	**0.8501**	0.8491
	Variance	**0.8501**	0.8494
	MAPE	15.83%	**15.32%**
45 min	RMSE	4.0649	**4.0635**
	MAE	2.6587	**2.6568**
	Accuracy	0.7167	**0.7168**
	R^2	0.8486	**0.8487**
	Variance	0.8487	**0.8489**
	MAPE	**16.16%**	16.24%
60 min	RMSE	4.0615	**4.0581**
	MAE	2.6702	**2.6657**
	Accuracy	0.7170	**0.7172**
	R^2	0.8488	**0.8491**
	Variance	0.8489	**0.8491**
	MAPE	16.14%	**15.91%**
75 min	RMSE	4.0874	**4.0835**
	MAE	**2.6927**	2.6974
	Accuracy	0.7152	**0.7155**
	R^2	0.8469	**0.8471**
	Variance	0.8470	**0.8474**
	MAPE	16.41%	**15.63%**
90 min	RMSE	4.0950	**4.0867**
	MAE	**2.7025**	2.7041
	Accuracy	0.7147	**0.7152**
	R^2	0.8463	**0.8470**
	Variance	0.8463	**0.8472**
	MAPE	15.93%	**15.81%**

The test uncertainty quantification results on the SZ-taxi dataset and the Los-loop dataset are illustrated in Table IV. As can be observed from the results, the models considering aleatoric uncertainty, i.e., Quantile regression, MVE, UAT-GCN/A and UAT-GCN, can capture more datapoints but has larger negative log-likelihoods compared to the Epistemic ones. On the contrary, the proposed UAT-GCN considering both epistemic and aleatoric uncertainty has the lowest mean negative log-likelihoods and largest PCIP, developing the best performance on uncertainty quantification.

Fig. 4 and Fig. 5 show the visualized uncertainty quantification results (including the epistemic uncertainty, the aleatoric uncertainty and the combined uncertainty) of an arbitrarily selected road segment selected from the SZ-taxi dataset and the Los-loop dataset, respectively. The aleatoric uncertainty represents the data noise and the process's randomness within the traffic speed data, while the epistemic uncertainty represents the uncertainty of the prediction means raised by data sparsity.

It is clear from the figures that the proposed model can not only forecast the traffic speed accurately but also achieve desirable coverage proportions. From the results shown in Tables I–IV, it can be seen that the improvements of the proposed method are two-fold, i.e., the proposed method can

TABLE IV

Test Uncertainty Quantification Results (horizon: 60 Minutes, Monte Carlo Sample Number: $1,000$)

Model	SZ-taxi			Los-loop		
	Mean PI	Mean NLL	PICP	Mean PI	Mean NLL	PICP
Quantile Regression	6.5040	–	0.9073	11.3428	–	0.9284
MVE	3.5957	3.5746	0.6870	6.3691	2.4514	0.7634
MC Dropout	0.4502	67.4087	0.1523	1.9294	7.5511	0.3129
Deep Ensembling	0.5573	29.2991	0.1918	2.7155	5.3271	0.4317
UAT-GCN/E	0.9171	12.0278	0.4567	2.2256	9.6844	0.4440
UAT-GCN/A	13.2507	1497.7111	0.8585	5.7776	7.2331	0.9366
UAT-GCN	4.4930	2.1647	0.9121	7.2697	2.2635	0.9545

TABLE V

Computation Time on the SZ-Taxi Dataset (horizon: 60 Minutes, Monte Carlo Sample Number: 50)

	Training Time (s/epoch)	Inference Time (s)
TGCN	0.94	0.04
Graph-WaveNet	4.17	0.02
STGCN	2.20	0.28
UAT-GCN/E	1.14	1.94
UAT-GCN/A	1.14	0.04
UAT-GCN	1.13	1.94
UAT-GCN/R	1.18	2.04

TABLE VI

Test RMSE, Test MAE and Inference Time w.r.t. Different Monte Carlo Sample Numbers (horizon: 60 minutes)

Sample Number	SZ-taxi			Los-loop		
	RMSE	MAE	Time (s)	RMSE	MAE	Time (s)
10	4.0693	2.6742	0.39	6.9552	3.9399	0.34
50	4.0595	2.6668	1.94	6.9221	3.9160	1.19
100	4.0588	2.6662	2.79	6.9190	3.9124	5.03
500	4.0583	2.6659	16.52	6.9172	3.9118	15.43
$1,000$	4.0580	2.6657	40.84	6.9171	3.9118	81.16
$1,500$	4.0579	2.6656	125.22	6.9172	3.9118	189.99

improve the point prediction performance and provide predictive distribution estimation compared to existing methods.

4) Computation Time and Ablation Study:

a) Computation time: The computation time of the best-performing baselines TGCN, Graph-WaveNet, STGCN, and the proposed models are illustrated in Table V. From the statistics, we can see that the training time of our model is marginally longer than the TGCN model, and is notably smaller than Graph-WavaNet and STGCN. As for the inference time, compared to other existing approaches, the proposed models involving epistemic uncertainty take more time due to the Monte Carlo simulations, but the step-wise inference time is 0.82 ms, which is still acceptable for real-world applications.

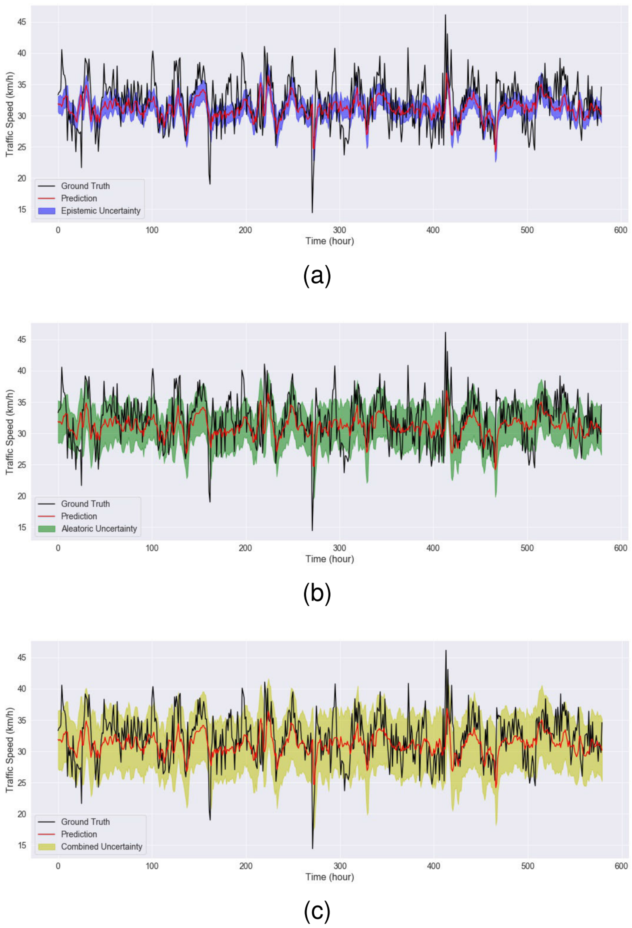
b) Effect of monte carlo sample number: Table VI illustrates the inference time with respect to various Monte Carlo sample numbers on both datasets. From the reported RMSE and MAE results, it can be seen that we can obtain relatively good results just a few Monte Carlo samples, e.g., 50, reduce the inference time. The performance converges when the sample number approaches $1,000$.

c) Comparison of different sampling methods: We also compare two different sampling methods for Monte Carlo simulation, namely the one proposed in Algorithm 3 (*Fast* method) and running Mote Carlo simulation by letting the input pass through the whole model architecture (*Regular* method). From the results demonstrated in Table VII, the second inference method is much more time consuming but does not increase the performance significantly. This result

TABLE VII

Comparison of Different Sampling Methods (horizon: 60 Minutes, Monte Carlo Sample Number: $1,000$)

Metrics	SZ-taxi		Los-loop	
	Fast	Regular	Fast	Regular
RMSE	4.0615	4.0616	6.9177	6.9171
MAE	2.6702	2.6703	3.9112	3.9113
Accuracy	0.7170	0.7170	0.8821	0.8821
R^2	0.8488	0.8488	0.7540	0.7540
Variance	0.8489	0.8489	0.7545	0.7545
MAPE	16.41%	16.14%	11.25%	11.25%
Time (s)	40.84	123.52	81.86	195.07



(a)

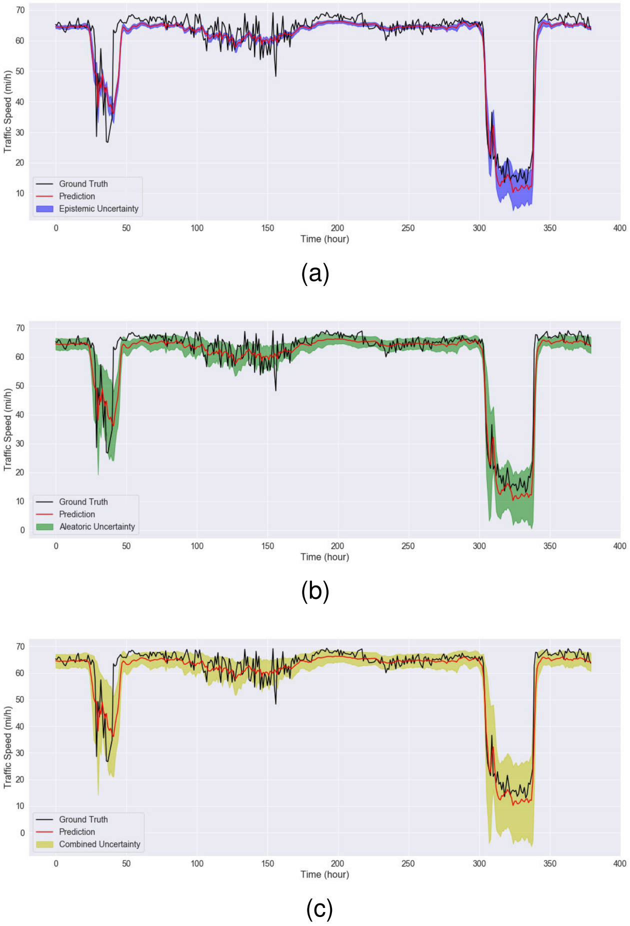
(b)

(c)

Fig. 4. Uncertainty quantification results on the SZ-taxi dataset with the prediction horizon of 60 minutes. (a) Epistemic uncertainty. (b) Aleatoric uncertainty. (c) Combined uncertainty.

TABLE VIII
ABLATION STUDY OF DIFFERENT DROPOUT RATES ON SZ-TAXI DATASET

Dropout rate	RMSE	MAE	Accuracy	R^2	Variance	MAPE	NLL	MPIW	PICP
0.1	4.0837	2.6611	0.7155	0.8473	0.8480	16.83%	2.0538	12.1001	0.9235
0.2	4.0394	2.6271	0.7185	0.8505	0.8595	15.62%	2.0330	12.9058	0.9370
0.5	4.5299	3.1272	0.6844	0.8159	0.8335	16.41%	1.8694	19.4392	0.9762



(a)

(b)

(c)

Fig. 5. Uncertainty quantification results on the Los-loop dataset with the prediction horizon of 60 minutes. (a) Epistemic uncertainty. (b) Aleatoric uncertainty. (c) Combined uncertainty.

indicates that for our model, a large number of stochastic layers in a deep spatio-temporal model may be redundant for Bayesian approximation, and thereby increase the inference time drastically. In contrast, the devised architecture can be regarded as a shallow BNN, which uses only a few stochastic layers at the end of the model to perform stochastic approximation. The weights of non-stochastic layers of the GCN and GRU modules in our model can be regarded as both prior of and posterior parameters for variational inference to some extent. While the last few dropout layers in the proposed model can yield sufficient epistemic/model uncertainty and reduce the inference time significantly by avoid passing through the GRU modules multiple times, which is particularly beneficial for potential future practical applications.

TABLE IX
TEST RESULTS UNDER ROAD CONGESTION

Time	Metrics	SZ-taxi	Los-loop
15 min	RMSE	6.9161	4.1202
	MAE	4.8648	2.7286
	Accuracy	0.6737	0.7293
	R^2	0.7976	0.9125
	Variance	0.7980	0.9125
	MAPE	13.15%	4.43.%
30 min	RMSE	6.8030	4.3113
	MAE	4.7611	2.5863
	Accuracy	0.6731	0.6817
	R^2	0.7971	0.8793
	Variance	0.7980	0.8819
	MAPE	12.86%	4.24%
45 min	RMSE	6.6890	4.6396
	MAE	4.6392	2.7178
	Accuracy	0.6706	0.6526
	R^2	0.7941	0.8562
	Variance	0.7953	0.8590
	MAPE	12.56%	4.45%
60 min	RMSE	7.0203	4.4559
	MAE	5.0071	2.6717
	Accuracy	0.6717	0.5917
	R^2	0.7951	0.8032
	Variance	0.7953	0.8138
	MAPE	13.48%	4.38%

d) Effect of different dropout rates: From the experimental results shown in Table VIII, it can be seen that when the dropout rate is too small (0.1), the model's uncertainty estimation performance decreases; when the dropout rate is too large (0.5), the model's point prediction performance decreases.

e) Performance under road congestion: To test the proposed method's performance under traffic congestion, we conduct the corresponding study. The congestion thresholds for the SZ-taxi and Los-loop datasets are under 30 km/h and under 50 mi/h, respectively. From the results shown in Table IX, we can see that the proposed method still has relative good performance under traffic congestion.

VI. CONCLUSION

In this paper, we introduce a novel uncertainty-aware deep learning model for traffic speed forecasting, namely, Uncertainty-Aware Temporal Graph Convolutional Network (UAT-GCN). In the proposed model, the spatial characteristics are modelled by a Graphical Convolutional Network and the temporal features are captured by a Gated Recurrent Unit, respectively. Particularly, we devise an uncertainty-aware regressor to estimate the prediction uncertainty. The proposed model is capable of estimating both data-related and model-related uncertainty, i.e., epistemic and aleatoric, respectively. We use Monte Carlo dropout to compute the epistemic uncertainty and the predictive mean and variance to estimate the aleatoric uncertainty. We validate the proposed UAT-GCN on two traffic speed datasets, and the obtained results suggest that our approach outperform state-of-the-art methods.

In future work, we plan to explore the possibility of integrating other deep learning techniques into our model, e.g, attention mechanism and graph structure learning, to further enhance the model. Additionally, relevant meterological features, e.g., weather, can also be included to improve the forecasting performance. Moreover, we plan to explore the potential of the proposed method for anomaly and outlier detection through aleatoric uncertainty estimation. We will also explore other techniques, e.g., adversarial learning and Bayesian variational learning, to enhance the robustness of uncertainty quantification methods against noise.

References

[1] M. Veres and M. Moussa, “Deep learning for intelligent transportation systems: A survey of emerging trends,” *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 8, pp. 3152–3168, Aug. 2020.

[2] W. Jiang and J. Luo, “Graph neural network for traffic forecasting: A survey,” 2021, *arXiv:2101.11174*.

[3] A. Haydari and Y. Yilmaz, “Deep reinforcement learning for intelligent transportation systems: A survey,” *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 1, pp. 11–32, Jan. 2022.

[4] S. Wang, J. Cao, and P. S. Yu, “Deep learning for spatio-temporal data mining: A survey,” *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 8, pp. 3681–3700, Aug. 2022.

[5] M. Abdar et al., “A review of uncertainty quantification in deep learning: Techniques, applications and challenges,” *Inf. Fusion*, vol. 76, pp. 243–297, Dec. 2021.

[6] A. Kendall and Y. Gal, “What uncertainties do we need in Bayesian deep learning for computer vision?” 2017, *arXiv:1703.04977*.

[7] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, “Weight uncertainty in neural network,” in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 1613–1622.

[8] C. E. Rasmussen, “Gaussian processes in machine learning,” in *Summer School on Machine Learning*. Berlin, Germany: Springer, 2003, pp. 63–71.

[9] Y. Gal and Z. Ghahramani, “Dropout as a Bayesian approximation: Representing model uncertainty in deep learning,” in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 1050–1059.

[10] D. Wu et al., “Quantifying uncertainty in deep spatiotemporal forecasting,” in *Proc. 27th ACM SIGKDD Conf. Knowl. Discovery Data Mining*, Aug. 2021, pp. 1841–1851.

[11] T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” 2016, *arXiv:1609.02907*.

[12] K. Cho, B. van Merrienboer, D. Bahdanau, and Y. Bengio, “On the properties of neural machine translation: Encoder-decoder approaches,” 2014, *arXiv:1409.1259*.

[13] Y. Li, R. Yu, C. Shahabi, and Y. Liu, “Diffusion convolutional recurrent neural network: Data-driven traffic forecasting,” in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–16.

[14] L. Zhao et al., “T-GCN: A temporal graph convolutional network for traffic prediction,” *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 9, pp. 3848–3858, Aug. 2019.

[15] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang, “Graph WaveNet for deep spatial–temporal graph modeling,” in *Proc. 28th Int. Joint Conf. Artif. Intell.*, 2019, pp. 1907–1913.

[16] A. Van Den Oord et al., “WaveNet: A generative model for raw audio,” in *Proc. SSW*, vol. 125, 2016, p. 2.

[17] A. Vaswani et al., “Attention is all you need,” in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 5998–6008.

[18] S. Guo, Y. Lin, N. Feng, C. Song, and H. Wan, “Attention based spatial–temporal graph convolutional networks for traffic flow forecasting,” in *Proc. AAAI Conf. Artif. Intell.*, 2019, vol. 33, no. 1, pp. 922–929.

[19] C. Zheng, X. Fan, C. Wang, and J. Qi, “GMAN: A graph multi-attention network for traffic prediction,” in *Proc. AAAI Conf. Artif. Intell.*, Apr. 2020, vol. 34, no. 1, pp. 1234–1241.

[20] B. Yu, H. Yin, and Z. Zhu, “Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting,” in *Proc. 27th Int. Joint Conf. Artif. Intell.*, Jul. 2018, pp. 3634–3640.

[21] X. Zhang et al., “Traffic flow forecasting with spatial–temporal graph diffusion network,” in *Proc. AAAI Conf. Artif. Intell.*, 2021, vol. 35, no. 17, pp. 15008–15015.

[22] D. A. Nix and A. S. Weigend, “Estimating the mean and variance of the target probability distribution,” in *Proc. IEEE Int. Conf. Neural Netw. (ICNN)*, vol. 1, Jun./Jul. 1994, pp. 55–60.

[23] R. Koenker and K. Hallock, “Quantile regression,” *J. Econ. Perspect.*, vol. 15, no. 4, pp. 143–156, 2001.

[24] J. Lei, M. G’Sell, A. Rinaldo, R. J. Tibshirani, and L. Wasserman, “Distribution-free predictive inference for regression,” *J. Amer. Stat. Assoc.*, vol. 113, no. 523, pp. 1094–1111, Jul. 2018.

[25] K. Stankeviciute, A. M. Alaa, and M. van der Schaar, “Conformal time-series forecasting,” in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 6216–6228.

[26] L. V. Jospin, W. Buntine, F. Boussaid, H. Laga, and M. Bennamoun, “Hands-on Bayesian neural networks—A tutorial for deep learning users,” 2020, *arXiv:2007.06823*.

[27] P. Izmailov, S. Vikram, M. D. Hoffman, and A. G. G. Wilson, “What are Bayesian neural network posteriors really like?” in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 4629–4640.

[28] M. Welling and Y. W. Teh, “Bayesian learning via stochastic gradient Langevin dynamics,” in *Proc. 28th Int. Conf. Mach. Learn. (ICML)*, 2011, pp. 681–688.

[29] A. Damianou and N. D. Lawrence, “Deep Gaussian processes,” in *Proc. Artif. Intell. Stat.*, 2013, pp. 207–215.

[30] B. Lakshminarayanan, A. Pritzel, and C. Blundell, “Simple and scalable predictive uncertainty estimation using deep ensembles,” in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 6402–6413.

[31] T. Garipov, P. Izmailov, D. Podoprikhin, D. P. Vetrov, and A. G. Wilson, “Loss surfaces, mode connectivity, and fast ensembling of DNNs,” in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 31, 2018, pp. 8803–8812.

[32] S. Fort, H. Hu, and B. Lakshminarayanan, “Deep ensembles: A loss landscape perspective,” 2019, *arXiv:1912.02757*.

[33] P. Izmailov, D. Podoprikhin, T. Garipov, D. Vetrov, and A. Wilson, “Averaging weights leads to wider optima and better generalization,” in *Proc. 34th Conf. Uncertainty Artif. Intell.*, vol. 2, 2018, pp. 876–885.

[34] A. G. Wilson and P. Izmailov, “Bayesian deep learning and a probabilistic perspective of generalization,” in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 4697–4708.

[35] J. J. Q. Yu, “Citywide estimation of travel time distributions with Bayesian deep graph learning,” *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 3, pp. 2366–2378, Mar. 2023.

[36] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[37] E. Snelson and Z. Ghahramani, “Local and global sparse Gaussian process approximations,” in *Proc. Artif. Intell. Statist.*, 2007, pp. 524–531.

[38] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A simple way to prevent neural networks from overfitting,” *J. Mach. Learn. Res.*, vol. 15, pp. 1929–1958, Sep. 2014.

[39] R. J. Williams and D. Zipser, “A learning algorithm for continually running fully recurrent neural networks,” *Neural Comput.*, vol. 1, no. 2, pp. 270–280, Jun. 1989.

[40] Z. Wu, S. Pan, G. Long, J. Jiang, X. Chang, and C. Zhang, “Connecting the dots: Multivariate time series forecasting with graph neural networks,” in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2020, pp. 753–763.

[41] J. Zhu, Q. Wang, C. Tao, H. Deng, L. Zhao, and H. Li, “AST-GCN: Attribute-augmented spatiotemporal graph convolutional network for traffic forecasting,” *IEEE Access*, vol. 9, pp. 35973–35983, 2021.

[42] T. Pearce, A. Brintrup, M. Zaki, and A. Neely, “High-quality prediction intervals for deep learning: A distribution-free, ensembled approach,” in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 4075–4084.

Weizhu Qian received the Ph.D. degree in computer science from Université Bourgogne-Franche-Comté in 2021. He was a Post-Doctoral Researcher with Aalborg University. He is currently a Lecturer with Soochow University, China. His research interests include deep learning, probabilistic inference, time series analysis, and data science.

Thomas Dyhre Nielsen received the Ph.D. degree in computer science from Aalborg University, Denmark, in 2001. He is currently a Professor MSO with the Department of Computer Science, Aalborg University, where he is also the Deputy Head of Research. He is the author/coauthor of more than 110 research articles and a textbook on *Bayesian Networks and Decision Graphs*. His main research interests include probabilistic machine learning, latent variable models, and uncertainty quantification. He is Senior Area Editor of the *International Journal of Approximate Reasoning*.

Yan Zhao (Senior Member, IEEE) received the Ph.D. degree in computer science from Soochow University in 2020. She is currently an Associate Professor with Aalborg University. Her research interests include spatial database and trajectory computing.

Kim Guldstrand Larsen received the Ph.D. degree in computer science from The University of Edinburgh, Edinburgh, U.K., in 1986. He is currently a Professor of computer science with Aalborg University, Aalborg, Denmark, and the Director of the ICT-Competence Center CISS, Center for Embedded Software Systems. He has been the Director of the Danish Innovation Network (InfinIT) and the Innovation Fund Denmark Research Center, DiCyPS. He is the holder of the VILLUM Investigator grant S4OS and held the ERC Advanced Grant LASSO. He is a member of the Academia Europaea and the Royal Danish Academy of Sciences and Letter and the Digital Wiseman of the Academy of Technical Sciences.

James Jianqiao Yu (Senior Member, IEEE) received the B.Eng. and Ph.D. degrees in electrical and electronic engineering from The University of Hong Kong, Pokfulam, Hong Kong, in 2011 and 2015, respectively. He was a Post-Doctoral Fellow with The University of Hong Kong from 2015 to 2018. He was an Assistant Professor with the Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen, China, from 2018 to 2023. He is currently a Lecturer with the Department of Computer Science, University of York. He has published over 90 academic papers in top international journals and conferences. His research interests include intelligent transportation systems, privacy computing, deep learning, and smart cities. His work is now mainly on forecasting, decision making, and privacy preservation of future transportation systems, and artificial intelligence techniques for industrial applications. His representative papers have been selected as ESI highly cited papers. He was the World's Top 2% Scientists in 2020 and in 2023 by Stanford University and ranked at top 0.4% of all artificial intelligence scholars and top 0.2% of transportation. He is an Editor of the *IET Smart Cities*.