RNN with Particle Flow for Probabilistic Spatio-temporal Forecasting - Supplementary Material -

Soumyasundar Pal [1] **Liheng Ma** [1] **Yingxue Zhang** [2] **Mark Coates** [1]

1. Particle flow for Bayesian inference in a state-space model

1.1. Background

Particle flow is an alternative to particle filters for Bayesian filtering in a state-space model. Recall the first order Markov model specified in eqs. (4), (5), and (6) in Section 5.1. of the main paper.

$$\mathbf{x}_1 \sim p_1(\cdot, \mathbf{z}_1, \rho)\,, \tag{1}$$

$$\mathbf{x}_t = g_{\mathcal{G},\psi}(\mathbf{x}_{t-1}, \mathbf{y}_{t-1}, \mathbf{z}_t, \mathbf{v}_t),\ \text{for } t > 1\,, \tag{2}$$

$$\mathbf{y}_t = h_{\mathcal{G},\phi}(\mathbf{x}_t, \mathbf{z}_t, \mathbf{w}_t),\ \text{for } t \geqslant 1\,. \tag{3}$$

Here $\mathbf{y}_t$ is the observation from the state-space model at time t. $\mathbf{x}_t$ and $\mathbf{z}_t$ denote the unobserved state variable and observed covariates at time t respectively. The filtering task is to compute the posterior distribution of the state trajectory $p_\Theta(\mathbf{x}_t|\mathbf{y}_{1:t}, \mathbf{z}_{1:t})$ recursively. Suppose we have a set of N_p samples (particles) $\{\mathbf{x}_{t-1}\}_{j=1}^{N_p}$ which approximates the posterior distribution of $\mathbf{x}_{t-1}$.

$$p_\Theta(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1}, \mathbf{z}_{1:t-1}) \approx \frac{1}{N_p}\sum_{j=1}^{N_p}\delta(\mathbf{x}_{t-1} - \mathbf{x}_{t-1}^j)\,. \tag{4}$$

In the 'predict' step, we approximate the predictive posterior distribution at time t as follows:

$$\begin{aligned} p_\Theta(\mathbf{x}_t|\mathbf{y}_{1:t-1}, \mathbf{z}_{1:t}) &= \int p_{\psi,\sigma}(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{y}_{t-1}, \mathbf{z}_t) \\ &\qquad p_\Theta(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1}, \mathbf{z}_{1:t-1})d\mathbf{x}_{t-1}\,, \\ &\approx \frac{1}{N_p}\sum_{j=1}^{N_p}\delta(\mathbf{x}_t - \tilde{\mathbf{x}}_t^j)\,, \end{aligned} \tag{5}$$

where, the particles $\{\tilde{\mathbf{x}}_t^j\}_{j=1}^{N_p}$ from the predictive posterior distribution $p_\Theta(\mathbf{x}_t|\mathbf{y}_{1:t-1}, \mathbf{z}_{1:t})$ are obtained by propagating $\{\mathbf{x}_{t-1}^j\}_{j=1}^{N_p}$ through the state-transition model specified by eq. (2). Subsequently, the 'update' step applies Bayes' theorem to compute the posterior distribution at time t as follows:

$$p_\Theta(\mathbf{x}_t|\mathbf{y}_{1:t}, \mathbf{z}_{1:t}) \propto p_\Theta(\mathbf{x}_t|\mathbf{y}_{1:t-1}, \mathbf{z}_{1:t})p_{\phi,\gamma}(\mathbf{y}_t|\mathbf{x}_t, \mathbf{z}_t)\,. \tag{6}$$

For non-linear state space models, particle filters (Gordon et al., 1993; Doucet & Johansen, 2009) employ importance sampling to approximate the 'update' step in eq. (6). However, constructing well-matched proposal distributions to the posterior distribution in high-dimensional state-spaces is extremely challenging. A mismatch between the proposal and the posterior leads to weight degeneracy after resampling, which results in poor performance of particle filters in high-dimensional problems (Bengtsson et al., 2008; Snyder et al., 2008; Beskos et al., 2014). Instead of sampling, particle flow filters offer a significantly better solution in complex problems by transporting particles continuously from the prior to the posterior (Daum & Huang, 2007; Ding & Coates, 2012; Daum & Huang, 2014; Daum et al., 2017).

1.2. Particle flow

In a given time step t, particle flow algorithms (Daum & Huang, 2007; Daum et al., 2010) solve differential equations to gradually migrate particles from the predictive distribution such that they represent the posterior distribution for the same time step after the flow. A particle flow can be modelled by a background stochastic process η_λ in a pseudo-time interval $\lambda \in [0, 1]$, such that the distribution of η_0 is the predictive distribution $p_\Theta(\mathbf{x}_t|\mathbf{y}_{1:t-1}, \mathbf{z}_{1:t})$ and the distribution of η_1 is the posterior distribution $p_\Theta(\mathbf{x}_t|\mathbf{y}_{1:t}, \mathbf{z}_{1:t})$. Since particle flow only considers migration of particles within a single time step, we omit the time index t in η_λ, $\mathbf{y}$, and $\mathbf{z}$ to simplify notation.

In (Daum et al., 2010), an ordinary differential equation (ODE) with zero diffusion governs the flow of η_λ:

$$\frac{d\eta_\lambda}{d\lambda} = \varphi(\eta_\lambda, \lambda)\,. \tag{7}$$

If the predictive distribution and the additive measurement noise is Gaussian and the measurement function h is linear,

[1]Department of Electrical and Computer Eningeering, McGill University, Montréal, QC, Canada. [2] Huawei Noah's Ark Lab, Montréal Research Center, Montréal, QC, Canada. Correspondence to: Soumyasundar Pal <soumyasundar.pal@mail.mcgill.ca>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021.

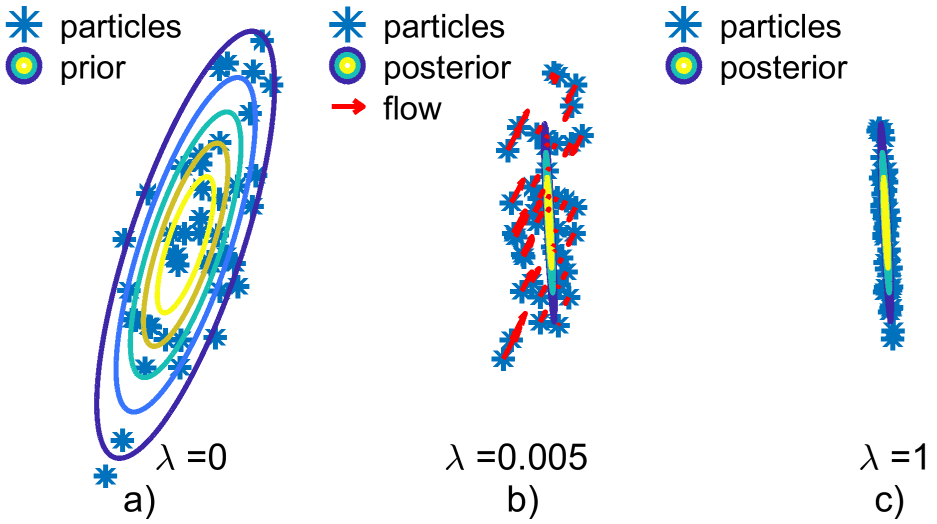


Figure 1. Migration of particles from a 2-d Gaussian prior to a 2-d Gaussian posterior distribution. a) The samples (asterisk) from the prior distribution, b) The contours of the posterior distribution and the direction of flow for the particles at an intermediate step, c) The particles after the flow, approximately distributed according to the posterior distribution.

the Exact Daum-Huang (EDH) flow is given as:

$$\varphi(\eta_\lambda, \lambda) = A(\lambda)\eta_\lambda + b(\lambda) \ , \tag{8}$$

where,

$$A(\lambda) = -\frac{1}{2}\bar{P}H^T(\lambda H\bar{P}H^T + R)^{-1}H \ , \tag{9}$$

$$b(\lambda) = (I + 2\lambda A(\lambda))[(I + \lambda A(\lambda))\bar{P}H^T R^{-1}\mathbf{y} + A(\lambda)\bar{\eta}_0] \ , \tag{10}$$

Here $\bar{\eta}_0$ and $\bar{P}$ are the mean vector and the covariance matrix of the predictive distribution respectively. For a general nonlinear state-space model, we usually a Gaussian approximation of the predictive distribution based on sample estimates of $\bar{\eta}_0$ and $\bar{P}$ or via the Extended Kalman Filter (EKF). $\mathbf{y}$ denotes the new observation at time t. The linear measurement model in $\mathbf{x}$ is specified by the measurement matrix $H = \frac{h_{\mathcal{G},\phi}(\mathbf{x},\mathbf{z},\mathbf{0})}{\partial \mathbf{x}}$, and R denotes the covariance matrix of the zero mean additive Gaussian measurement noise. For a nonlinear measurement model, we use a first order Taylor series approximation at the mean of the particles $\bar{\eta}_\lambda$ and replace H by $H(\lambda) = \frac{\partial h_{\mathcal{G},\phi}(\eta,\mathbf{z},\mathbf{0})}{\partial \eta}\Big|_{\eta=\bar{\eta}_\lambda}$ and $\mathbf{y}$ by $\big(\mathbf{y} - e(\lambda)\big)$, where the linearization error $e(\lambda) = h_{\mathcal{G},\phi}(\bar{\eta}_\lambda, \mathbf{z}, \mathbf{0}) - H(\lambda)\bar{\eta}_\lambda$ in eq. (9) and (10). Similarly, for a zero mean non-Gaussian measurement noise, we use a Gaussian approximation to replace R in eq. (9) and (10) by $R(\lambda) = \text{Cov}\,[\mathbf{y}|\bar{\eta}_\lambda, \mathbf{z}]$. A detailed description of the implementation of the exact Daum-Huang (EDH) filter is provided in (Choi et al., 2011).

Numerical integration is usually used to solve the ODE in Equation (8). The integral between λ_{m-1} and λ_m for $1 \leqslant m \leqslant N_\lambda$, where $\lambda_0 = 0$ and $\lambda_{N_\lambda} = 1$, is approximated via the Euler update rule. For the j-th particle, the EDH flow in the m-th pseudo-time interval becomes:

$$\eta^j_{\lambda_m} = \eta^j_{\lambda_{m-1}} + \epsilon_m\big(A(\lambda_{m-1})\eta^j_{\lambda_{m-1}} + b(\lambda_{m-1})\big) \ , \tag{11}$$

where the step size $\epsilon_m = \lambda_m - \lambda_{m-1}$ and $\sum_{m=1}^{N_\lambda} \epsilon_m = 1$. We start the particle flow from $\eta^j_0 = \tilde{\mathbf{x}}^j_t$ and after the flow is complete, we set $\mathbf{x}^j_t = \eta^j_1$ to approximate the posterior

Algorithm 1 Particle flow

Input: $\{\eta^j_0 = \tilde{\mathbf{x}}^j_t\}_{j=1}^{N_p}$, $\mathbf{y}_t$, $\mathbf{z}_t$, $\{\epsilon_m\}_{m=1}^{N_\lambda}$, and Θ
Output: $\{\mathbf{x}^j_t = \eta^j_1\}_{j=1}^{N_p}$
Compute $\bar{\eta}_0 = \frac{1}{N_p}\sum_{j=1}^{N_p}\eta^j_0$
Compute $\bar{P} = \frac{1}{N_p}\sum_{j=1}^{N_p}\left[(\eta^j_0 - \bar{\eta}_0)(\eta^j_0 - \bar{\eta}_0)^T\right]$
Set $\lambda_0 = 0$
for $m = 1, 2, ..., N_\lambda$ **do**
  $\lambda_m = \lambda_{m-1} + \epsilon_m$
  Linearize the measurement model at $\bar{\eta}_{\lambda_{m-1}} = \frac{1}{N_p}\sum_{j=1}^{N_p}\eta^j_{\lambda_{m-1}}$ to compute $H(\lambda_{m-1})$ and $e(\lambda_{m-1})$.
  Compute $R(\lambda_{m-1}) = \textbf{Cov}\left[\mathbf{y}_t|\bar{\eta}_{\lambda_{m-1}}, \mathbf{z}_t\right]$.
  Compute $A(\lambda_{m-1})$ and $b(\lambda_{m-1})$ using eq. (9) and (10).
  Apply particle flow to all particles: $\eta^j_m = \eta^j_{m-1} + \epsilon_m\big(A(\lambda_{m-1})\eta^j_{m-1} + b(\lambda_{m-1})\big)$
end for
Set $\mathbf{x}^j_t = \eta^j_1$ for $1 \leqslant j \leqslant N_p$

distribution of $\mathbf{x}_t$ as:

$$p_\Theta(\mathbf{x}_t|\mathbf{y}_{1:t}, \mathbf{z}_{1:t}) \approx \frac{1}{N_p}\sum_{j=1}^{N_p}\delta(\mathbf{x}_t - \mathbf{x}_t^j)\,. \tag{12}$$

The overall EDH particle flow algorithm is summarized in Algorithm 1. Figure 1 demonstrates the migration of the particles from the prior to the posterior distribution for a Gaussian predictive distribution and a linear-Gaussian measurement model.

2. Model training

Algorithm 2 summarizes the learning of the model parameters Θ, described in Section 5.2.2 of the main paper.

Algorithm 2 Model training and testing

1: **Input:** Training and test data: $\{\mathbf{y}_{1:P+Q}^{(m)}, \mathbf{z}_{1:P+Q}^{(m)}\}_{m\in\mathcal{D}_{trn}}$, $\{\mathbf{y}_{1:P}^{(n)}, \mathbf{z}_{1:P+Q}^{(n)}\}_{n\in\mathcal{D}_{test}}$
2: **Output:** $\{\hat{p}_{\hat{\Theta}}(\mathbf{y}_{P+1:P+Q}^{(n)}|\mathbf{y}_{1:P}^{(n)}, \mathbf{z}_{1:P+Q}^{(n)})\}_{n\in\mathcal{D}_{test}}$
3: **Hyperparameters:** Number of iterations N_{iter}, step-size $\{\zeta_k\}_{k=1}^{N_{iter}}$
4: **Initialization:** random initialization for the system parameters Θ_0
5: **Model training:**
6: Set $k = 1$
7: **while** $k \leqslant N_{iter}$ **do**
8: Sample a minibatch $\mathcal{D} \subset \mathcal{D}_{trn}$.
9: Compute the approximate posterior distribution of the forecasts $\{\hat{p}_{\Theta_{k-1}}(\mathbf{y}_{P+1:P+Q}^{(m)}|\mathbf{y}_{1:P}^{(m)}, \mathbf{z}_{1:P+Q}^{(m)})\}_{m\in\mathcal{D}}$ using Algorithm 1 in the main paper with the current parameters Θ_{k-1}.
10: Compute the gradient of the chosen loss function $\mathcal{L}(\Theta, \mathcal{D})$ w.r.t. model parameters Θ at Θ_{k-1}
11: Update the system parameters using SGD algorithm: $\Theta_k = \Theta_{k-1} - \zeta_k\nabla_\Theta\mathcal{L}(\Theta, \mathcal{D})\big|_{\Theta=\Theta_{k-1}}$
12: $k = k + 1$
13: **end while**
14: Save the estimated model $\widehat{\Theta} = \Theta_{N_{iter}}$
15: **Testing:**
16: Compute the test set forecast posterior distributions $\{\hat{p}_{\hat{\Theta}}(\mathbf{y}_{P+1:P+Q}^{(n)}|\mathbf{y}_{1:P}^{(n)}, \mathbf{z}_{1:P+Q}^{(n)})\}_{n\in\mathcal{D}_{test}}$ using Algorithm 1 in the main paper with the estimated model parameters $\widehat{\Theta}$.

3. Description and statistics of datasets

The statistics of the PeMS datasets and the non-graph datasets used in our experiments are summarized in Tables 1 and 2 respectively. The description of the PeMS datasets are provided in Section 6.1. of the main paper. The Electricity[1] dataset contains electricity consumption for 370 clients. The Traffic[2] dataset is composed of 963 time-series of lane occupancy rates. The Taxi[3] dataset contains counts of taxis on different roads and the Wikipedia[4] dataset specifies clicks to web links.

Table 1. Summary statistics of the PeMS road traffic datasets

Dataset	PeMSD3	PeMSD4	PeMSD7	PeMSD8
No. nodes	358	307	228	170
No. time steps	26208	16992	12672	17856
Interval	5 min	5 min	5 min	5 min

Table 2. Summary statistics of the multivariate non-graph datasets

Dataset	No. time series (N)	Domain	Freq.	No. time steps	Prediction length (Q)
Electricity	370	$\mathbb{R}^+$	Hourly	5833	24
Traffic	963	$(0, 1)$	Hourly	4001	24
Taxi	1214	$\mathbb{N}$	30 Minutes	1488	24
Wikipedia	2000	$\mathbb{N}$	Daily	792	30

4. Definitions of evaluation metrics

The point forecasts are evaluated by computing mean absolute error (MAE), mean absolute percentage error (MAPE), and root mean squared error (RMSE). For the test-set indexed by $\mathcal{D}_{test}$, let $\mathbf{y}_t^{(m)} \in \mathbb{R}^N$ and $\hat{\mathbf{y}}_t^{(m)} \in \mathbb{R}^N$ denote the ground truth and the prediction at horizon t for m-th test example respectively. The average MAE, MAPE, and RMSE at horizon t are defined as follows:

$$\text{MAE}(\mathcal{D}_{test}, t) = \frac{1}{N|\mathcal{D}_{test}|}\sum_{m\in\mathcal{D}_{test}}||\mathbf{y}_t^{(m)} - \hat{\mathbf{y}}_t^{(m)}||_1\,, \tag{13}$$

$$\text{MAPE}(\mathcal{D}_{test}, t) = \frac{1}{N|\mathcal{D}_{test}|}\sum_{m\in\mathcal{D}_{test}}\sum_{i=1}^{N}\frac{|\mathbf{y}_{t,i}^{(m)} - \hat{\mathbf{y}}_{t,i}^{(m)}|}{|\mathbf{y}_{t,i}^{(m)}|}\,, \tag{14}$$

$$\text{RMSE}(\mathcal{D}_{test}, t) = \sqrt{\frac{1}{N|\mathcal{D}_{test}|}\sum_{m\in\mathcal{D}_{test}}||\mathbf{y}_t^{(m)} - \hat{\mathbf{y}}_t^{(m)}||_2^2}\,, \tag{15}$$

For comparison among the probabilistic forecasting mod-

[1] https://archive.ics.uci.edu/ml/datasets/ElectricityLoadDiagrams20112014
[2] https://archive.ics.uci.edu/ml/datasets/PEMS-SF
[3] https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page
[4] https://github.com/mbohlkeschneider/gluon-ts/tree/mv_release/datasets

els, we compute the Continuous Ranked Probability Score (CRPS) (Gneiting & Raftery, 2007), and the P10 and P90 Quantile Losses (QL) (Salinas et al., 2020; Wang et al., 2019). Let $F(\cdot)$ be the Cumulative Distribution Function (CDF) of the forecast of the true value $x \in \mathbb{R}$. We denote by $\mathbf{1}\{x \leqslant z\}$ the indicator function that attains the value 1 if $x \leqslant z$ and the value 0 otherwise. The continuous ranked probability score (CRPS) is defined as:

$$\text{CRPS}(F, x) = \int_{-\infty}^{\infty} \Big(F(z) - \mathbf{1}\{x \leqslant z\}\Big)^2 dz\,. \quad (16)$$

CRPS is a proper scoring function, i.e., it attains its minimum value of zero when the forecast CDF F is a step function at the ground truth x. The average CRPS at horizon t is defined as the average marginal CRPS across different time-series.

$$\text{CRPS}_{avg}(\mathcal{D}_{test}, t) = \frac{1}{N|\mathcal{D}_{test}|} \sum_{m \in \mathcal{D}_{test}} \sum_{i=1}^{N} \text{CRPS}(F_{t,i}^{(m)}, \mathbf{y}_{t,i}^{(m)})\,, \quad (17)$$

where $F_{t,i}^{(m)}(\cdot)$ is the marginal CDF of the forecast at horizon t for i-th time-series in m-th test example.

Let $F_{t,sum}^{(m)}(\cdot)$ is the CDF of the sum of the forecasts of all time-series at horizon t in m-th test example. The (normalized) CRPS_{sum} is defined as:

$$\text{CRPS}_{sum}(\mathcal{D}_{test}) = \frac{\sum_t \sum_{m \in \mathcal{D}_{test}} \text{CRPS}(F_{t,sum}^{(m)}, \sum_{i=1}^{N} \mathbf{y}_{t,i}^{(m)})}{\sum_t \sum_{m \in \mathcal{D}_{test}} |\sum_{i=1}^{N} \mathbf{y}_{t,i}^{(m)}|}\,. \quad (18)$$

For a given quantile $\alpha \in (0, 1)$, a true value x, and an α-quantile prediction $\hat{x}(\alpha) = F^{-1}(\alpha)$, the α-quantile loss is defined as:

$$\text{QL}\big(x, \hat{x}(\alpha)\big) = 2\Bigg(\alpha\big(x - \hat{x}(\alpha)\big)\mathbf{1}\{x > \hat{x}(\alpha)\} + (1-\alpha)\big(\hat{x}(\alpha) - x\big)\mathbf{1}\{x \leqslant \hat{x}(\alpha)\}\Bigg)\,. \quad (19)$$

The average (normalized) quantile loss (QL) is defined as follows:

$$\text{QL}_{avg}(\mathcal{D}_{test}, t, \alpha) = \frac{\sum_{m \in \mathcal{D}_{test}} \sum_{i=1}^{N} \text{QL}\big(\mathbf{y}_{t,i}^{(m)}, \hat{\mathbf{y}}_{t,i}^{(m)}(\alpha)\big)}{\sum_{m \in \mathcal{D}_{test}} \sum_{i=1}^{N} |\mathbf{y}_{t,i}^{(m)}|}\,. \quad (20)$$

The P10QL metric is obtained by setting $\alpha = 0.1$ in eq. (20); the P90QL metric corresponds to $\alpha = 0.9$ and the ND (P50QL) metric is obtained using $\alpha = 0.5$.

5. Detailed experimental results on the PeMS datasets

5.1. Baseline algorithms

For the experiments on the PeMS road traffic datasets, we compare the proposed AGCGRU+flow algorithm with four different classes of forecasting techniques, listed as follows:

Graph agnostic statistical and machine learning based point forecasting models:

- HA (Historical Average): uses the seasonality of the historical data.
- ARIMA (Makridakis & Hibon, 1997): implemented using a Kalman filter.
- Vector Auto-Regressive model (VAR) (Hamilton, 1994): generalization of AR model to multivariate setting.
- Support Vector Regression (SVR) (Chun-Hsin et al., 2004)
- FNN (Feedforward Neural Network).
- FC-LSTM (Sutskever et al., 2014): encoder-decoder architecture for sequence to sequence prediction using fully connected LSTM layers.

Spatio-temporal point forecast models:

- DCRNN (Li et al., 2018): Diffusion Convolutional Recurrent Neural Network, combines diffusion convolution with GRU to form an encoder-decoder architecture for sequence to sequence prediction.
- STGCN (Yu et al., 2018): Spatio-Temporal Graph Convolutional Network, uses gated temporal convolution with graph convolution.
- ASTGCN (Guo et al., 2019): Attention Spatial-Temporal Graph Convolutional Network, spatial and temporal attentions to learn recent and seasonal patterns.
- GWN (Wu et al., 2019): Graph WaveNet, built using graph convolution and dilated causal convolution, provision for learnable graph.
- GMAN (Zheng et al., 2020): Graph Multi-Attention Network, multiple spatio-temporal attention blocks to form an encoder-decoder architecture, transform attention between encoder and decoder.
- AGCRN (Bai et al., 2020): Adaptive Graph Convolutional Recurrent Network, node adaptive parameter learning for graph convolution using adaptive adjacency, combined with GRU.

Table 3. Average MAE, MAPE and RMSE for PeMSD3 dataset for 15/30/45/60 minutes horizons. The best and the second best results in each column are shown in bold and marked with underline respectively. Lower numbers are better.

Algorithm	PeMSD3 (15/ 30/ 45/ 60 min)		
	MAE	MAPE(%)	RMSE
HA	31.58	33.78	52.39
ARIMA	17.31/22.12/27.35/32.47	16.53/20.78/25.66/30.84	26.80/34.60/42.37/49.98
VAR	18.59/20.80/23.06/24.86	19.59/21.81/24.24/26.44	31.05/33.92/36.93/39.32
SVR	16.66/20.33/24.33/28.34	16.07/19.45/23.31/27.57	25.97/32.19/38.30/44.57
FNN	16.87/20.30/23.91/27.74	19.59/23.67/30.09/35.44	25.46/30.97/36.27/41.86
FC-LSTM	19.01/19.46/19.92/20.29	19.77/20.23/20.82/21.30	32.96/33.59/34.24/34.83
DCRNN	14.42/15.87/17.10/18.29	14.57/15.78/16.87/17.95	24.33/27.05/28.99/30.76
STGCN	15.22/17.54/19.74/21.59	16.22/18.44/20.13/21.88	26.20/29.10/32.19/34.83
ASTGCN	17.03/18.50/19.58/20.95	18.02/19.28/20.18/21.61	29.04/31.81/33.98/36.37
GWN	14.63/16.56/18.34/20.08	**13.74**/15.24/16.82/18.75	25.06/28.48/31.11/33.58
GMAN	14.73/15.44/16.15/16.96	15.63/16.25/16.99/17.91	24.48/25.68/26.80/27.99
AGCRN	14.20/15.34/16.28/17.38	13.79/**14.47**/**15.14**/16.25	24.75/26.61/28.06/29.61
LSGCN	14.28/16.08/17.77/19.23	14.80/16.01/17.15/18.21	25.88/28.11/30.31/32.37
DeepGLO	14.79/18.89/19.11/23.53	14.12/16.92/17.75/21.68	22.97/29.17/30.48/35.64
N-BEATS	15.57/18.12/20.50/23.03	15.56/18.05/20.50/23.19	24.44/28.69/32.62/36.72
FC-GAGA	14.68/15.85/16.40/17.04	15.57/15.88/16.32/17.16	24.65/26.85/27.90/28.97
DeepAR	15.84/18.15/20.30/22.64	16.26/18.42/20.19/22.56	26.33/29.96/33.12/36.65
DeepFactors	17.53/20.17/22.78/24.87	19.22/24.42/29.58/34.43	27.62/31.83/35.36/37.91
MQRNN	14.60/16.55/18.34/20.12	15.17/17.34/18.94/20.66	25.35/28.77/31.50/34.40
AGCGRU+flow	**13.79/14.84/15.58/16.06**	14.01/14.75/15.34/**15.80**	**22.08/24.26/25.55/26.43**

Table 4. Average MAE, MAPE and RMSE for PeMSD4 dataset for 15/30/45/60 minutes horizons. The best and the second best results in each column are shown in bold and marked with underline respectively. Lower numbers are better.

Algorithm	PeMSD4 (15/ 30/ 45/ 60 min)		
	MAE	MAPE(%)	RMSE
HA	3.16	7.00	6.13
ARIMA	1.53/2.01/2.37/2.68	2.92/4.06/4.96/5.73	3.11/4.36/5.25/5.95
VAR	1.66/2.12/2.39/2.57	3.27/4.33/4.95/5.36	3.09/4.02/4.51/4.83
SVR	1.48/1.91/2.23/2.49	2.88/3.97/4.86/5.61	3.11/4.29/5.08/5.66
FNN	1.48/1.90/2.23/2.51	3.04/4.09/4.98/5.80	3.08/4.27/5.08/5.68
FC-LSTM	2.20/2.22/2.23/2.26	4.95/4.97/4.99/5.05	4.89/4.92/4.95/5.01
DCRNN	1.38/1.78/2.06/2.29	2.69/3.72/4.51/5.16	2.95/4.09/4.81/5.34
STGCN	1.42/1.85/2.14/2.39	2.82/3.92/4.71/5.34	2.94/4.03/4.70/5.21
ASTGCN	1.69/2.15/2.40/2.55	3.70/4.85/5.46/5.79	3.54/4.71/5.35/5.62
GWN	1.37/1.76/2.03/2.24	**2.67**/3.73/4.52/5.15	2.94/4.07/4.77/5.28
GMAN	1.38/**1.61**/**1.76**/**1.88**	2.80/**3.42**/**3.84**/4.18	2.98/**3.70**/**4.11**/**4.41**
AGCRN	1.41/1.67/1.84/2.01	2.88/3.55/3.99/4.40	3.04/3.83/4.33/4.73
LSGCN	1.40/1.78/2.03/2.20	2.80/3.71/4.27/4.68	**2.87**/3.90/4.50/4.89
DeepGLO	1.61/1.89/2.25/2.51	3.13/4.06/5.03/5.77	3.06/4.14/4.92/5.55
N-BEATS	1.49/1.90/2.20/2.44	2.93/4.00/4.84/5.48	3.13/4.29/5.05/5.58
FC-GAGA	1.43/1.78/1.95/2.06	2.87/3.80/4.32/4.67	3.06/4.09/4.55/4.82
DeepAR	1.51/2.01/2.38/2.68	3.06/4.41/5.45/6.25	3.11/4.27/5.04/5.60
DeepFactors	1.54/2.01/2.34/2.61	3.07/4.26/5.17/5.90	3.11/4.21/4.90/5.40
MQRNN	1.37/1.76/2.03/2.25	2.68/3.72/4.51/5.17	2.94/4.05/4.73/5.20
AGCGRU+flow	**1.35**/1.63/1.78/**1.88**	**2.67**/3.44/3.87/**4.16**	2.88/3.77/4.20/4.46

Table 5. Average MAE, MAPE and RMSE for PeMSD7 dataset for 15/30/45/60 minutes horizons. The best and the second best results in each column are shown in bold and marked with underline respectively. Lower numbers are better.

Algorithm	PeMSD7 (15/ 30/ 45/ 60 min)		
	MAE	MAPE(%)	RMSE
HA	3.98	10.92	7.20
ARIMA	2.49/3.52/4.32/5.03	5.66/8.30/10.46/12.35	4.53/6.64/8.17/9.42
VAR	2.70/3.71/4.37/4.87	6.23/8.75/10.37/11.56	4.38/5.95/6.89/7.56
SVR	2.43/3.40/4.15/4.78	5.62/8.23/10.38/12.31	4.52/6.53/7.93/9.02
FNN	2.36/3.32/4.06/4.71	5.56/8.20/10.41/12.44	4.45/6.46/7.84/8.90
FC-LSTM	3.55/3.59/3.64/3.70	9.12/9.17/9.25/9.37	6.83/6.91/6.99/7.11
DCRNN	2.23/3.06/3.67/4.18	5.19/7.50/9.31/10.90	4.26/6.05/7.28/8.24
STGCN	2.21/2.96/3.47/3.90	5.20/7.32/8.82/10.09	4.09/5.72/6.76/7.55
ASTGCN	2.71/3.72/4.28/4.60	6.68/9.51/11.06/11.86	4.64/6.53/7.60/8.13
GWN	2.23/3.03/3.56/3.98	5.26/7.63/9.25/10.56	4.27/5.99/7.03/7.76
GMAN	2.40/2.76/**2.98**/**3.16**	5.93/6.96/7.66/**8.16**	4.74/5.57/**6.06**/**6.37**
AGCRN	2.19/2.81/3.15/3.42	5.22/7.09/8.19/9.01	4.12/5.49/6.27/6.79
LSGCN	2.23/2.99/3.50/3.95	5.22/7.18/8.40/9.37	**4.03**/5.59/6.54/7.30
DeepGLO	2.55/3.32/4.16/4.85	6.10/8.31/11.16/13.19	4.53/6.30/7.68/8.84
N-BEATS	2.44/3.34/4.02/4.57	5.75/8.30/10.31/11.94	4.55/6.51/7.84/8.80
FC-GAGA	2.22/2.85/3.18/3.36	5.32/7.09/8.00/8.51	4.29/5.77/6.46/6.82
DeepAR	2.53/3.61/4.48/5.20	6.15/9.30/12.17/14.49	4.55/6.50/7.84/8.87
DeepFactors	2.51/3.47/4.17/4.71	6.14/9.04/11.21/12.93	4.47/6.21/7.30/8.08
MQRNN	2.22/3.03/3.58/4.00	5.26/7.70/9.53/10.97	4.23/5.91/6.98/7.73
AGCGRU+flow	**2.15**/**2.70**/2.99/3.19	**5.13**/**6.75**/**7.61**/8.18	4.11/**5.46**/6.12/6.54

Table 6. Average MAE, MAPE and RMSE for PeMSD8 dataset for 15/30/45/60 minutes horizons. The best and the second best results in each column are shown in bold and marked with underline respectively. Lower numbers are better.

Algorithm	PeMSD8 (15/ 30/ 45/ 60 min)		
	MAE	MAPE(%)	RMSE
HA	2.47	5.66	5.19
ARIMA	1.24/1.61/1.89/2.12	2.33/3.15/3.77/4.31	2.63/3.62/4.28/4.81
VAR	1.37/1.79/2.04/2.23	2.66/3.62/4.23/4.69	2.67/3.53/4.01/4.36
SVR	1.21/1.56/1.80/2.01	2.32/3.12/3.72/4.24	2.64/3.57/4.18/4.63
FNN	1.19/1.54/1.79/2.01	2.27/3.12/3.75/4.30	2.59/3.55/4.17/4.63
FC-LSTM	1.91/1.93/1.94/1.95	4.63/4.66/4.69/4.72	4.71/4.75/4.78/4.81
DCRNN	1.16/1.49/1.70/1.87	2.25/3.16/3.85/4.37	2.54/3.49/4.08/4.49
STGCN	1.22/1.56/1.79/1.98	2.49/3.43/4.06/4.48	2.67/3.65/4.22/4.59
ASTGCN	1.36/1.64/1.81/1.92	3.04/3.79/4.23/4.51	2.98/3.77/4.20/4.47
GWN	**1.11**/1.40/1.59/1.73	**2.14**/**2.94**/3.49/3.90	**2.52**/3.45/4.00/4.38
GMAN	1.23/**1.36**/**1.46**/**1.55**	2.73/3.09/**3.38**/**3.63**	3.05/3.50/**3.82**/**4.06**
AGCRN	1.16/1.39/1.53/1.67	2.49/3.10/3.50/3.84	2.67/**3.44**/3.91/4.25
LSGCN	1.21/1.54/1.75/1.89	2.56/3.44/3.95/4.30	2.71/3.64/4.14/4.46
DeepGLO	1.30/1.75/2.04/2.21	2.48/3.42/4.06/4.50	2.67/3.63/4.24/4.69
N-BEATS	1.33/1.69/1.92/2.12	2.74/3.85/4.45/4.90	2.81/3.94/4.52/4.92
FC-GAGA	1.18/1.47/1.62/1.72	2.37/3.21/3.76/4.11	2.65/3.61/4.10/4.39
DeepAR	1.25/1.61/1.87/2.10	2.53/3.40/4.08/4.67	2.67/3.59/4.17/4.61
DeepFactors	1.26/1.63/1.88/2.07	2.51/3.42/4.08/4.61	2.63/3.54/4.11/4.52
MQRNN	1.13/1.43/1.62/1.77	2.19/2.99/3.56/4.00	2.54/3.48/4.02/4.40
AGCGRU+flow	1.13/1.37/1.49/1.57	2.30/3.01/3.40/3.65	2.59/3.45/3.85/4.09

- LSGCN (Huang et al., 2020): Long Shortterm Graph Convolutional Network, a novel attention mechanism and graph convolution, integrated into a spatial gated block.

Deep-learning based point forecasting methods:

- DeepGLO (Sen et al., 2019): global matrix factorization, regularization using temporal convolution.
- N-BEATS (Oreshkin et al., 2020): Neural Basis Expansion Analysis for Interpretable Time-Series, an univariate model, built using backward and forward residual connections and deep stack of fully-connected layers.
- FC-GAGA (Oreshkin et al., 2021): Fully Connected GAted Graph Architecture, fully connected hard graph gating combined with N-BEATS.

Deep-learning based probabilistic forecasting methods:

- DeepAR (Salinas et al., 2020): RNN based probabilistic method using parametric likelihood for forecasts.
- DeepFactors (Wang et al., 2019): global deep learning component along with a local classical model to account for uncertainty.
- MQRNN (Wen et al., 2017): RNN based multiple quantile regression.

5.1.1. Detailed comparisons with baselines for the PeMS datasets

In Table 1 of the main paper, we report the average MAE of the top 10 algorithms. The detailed comparisons in terms of MAE, MAPE, and RMSE with all the baseline algorithms on the four PeMS datasets are provided in Tables 3, 4, 5, and 6. We observe that statistical models such as HA, ARIMA, and VAR and basic machine learning models such as SVR, FNN, and FC-LSTM show poor predictive performance as they cannot model the complex spatio-temporal patterns present in the real world traffic data well. Graph agnostic deep learning models such as DeepGLO and N-BEATS perform better than the statistical models, but they cannot incorporate the graph structure when learning. FC-GAGA has lower forecasting errors as it is equipped with a graph learning module. The spatio-temporal graph-based models (especially AGCRN, GMAN, GWN, and LSGCN) display better performance. These models either use the observed graph or learn the graph structure from the data. In general, the deep learning based probabilistic forecasting algorithms such as DeepAR, DeepFactors, and MQRNN do not account for the spatial relationships in the data as well as the graph-based models, although MQRNN is among the best performing algorithms. DeepAR and DeepFactors aim to model the forecasting distributions and thus do not perform as well in the point forecasting task. The training loss function (negative log likelihood of the forecasts) does not match the evaluation metric. However, MQRNN shows better performance, possibly because it does target learning the median of the forecasting distribution along with other quantiles. The proposed AGCGRU+flow algorithm demonstrates comparable prediction accuracy to the best-performing spatio-temporal models and achieves the best average ranking across the four datasets. Figure 2 demonstrates that the proposed AGCGRU+flow has lower average MAE in most of the nodes compared to the second best performing AGCRN algorithm, for all four PeMS datasets. Some qualitative visualization of the confidence intervals for 15-minute ahead predictions for the PeMSD3, PeMSD4, PeMSD7, and PeMSD8 datasets are shown in Figures 3, 4, 5, and 6 respectively. We observe that the confidence intervals from the proposed algorithm are considerably tighter compared to its competitors in most cases, whereas the coverage of the ground truth is still ensured.

5.2. Detailed results for comparison with particle filter

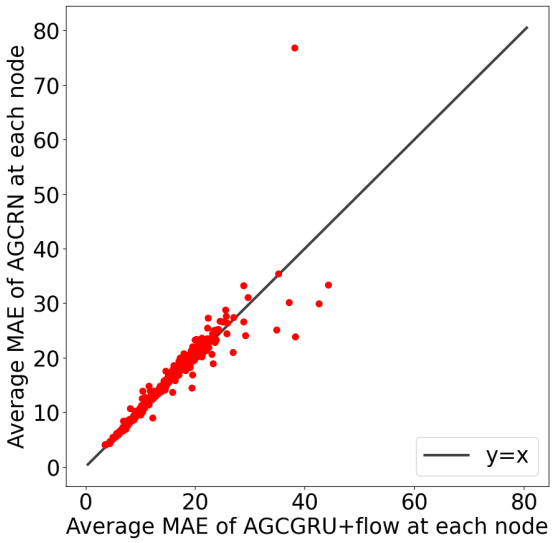
In Table 4 of the main paper, we compare the average MAE and average CRPS of the proposed AGCGRU+flow with a Bootstrap Particle Filter (BPF) (Gordon et al., 1993) based approach. Tables 7 and 8 provide the detailed comparison both in terms of point forecasting and probabilistic forecasting metrics. We observe that the proposed AGCGRU+flow algorithm outperforms the particle filter based approach in most cases.

5.3. Effect of number of particles

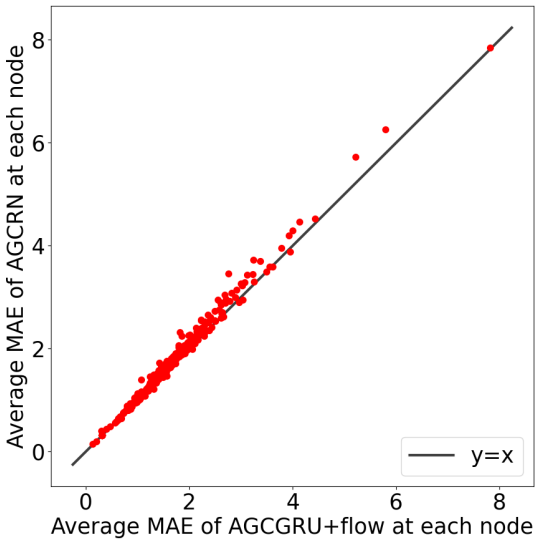
For this experiment, we consider three different settings with varying number of particles $N_p = 1/10/50$ for testing. The model is trained using 1 particle in each case. From Table 9, we observe that increasing the number of particles cannot improve the point forecasting accuracy significantly, whereas the results in Table 10 show that characterization of the prediction uncertainty is improved as more particles are used to form the approximate posterior distribution of the forecasts.

5.4. Effect of different learnable noise variance at each node

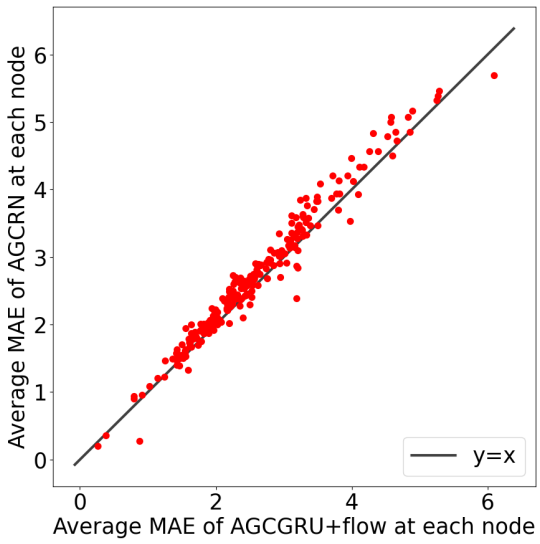
In this experiment, we compare the proposed state-space model with different learnable noise variance at each node (parameterized by the softplus function in eq. (9) in the main paper with fixed and uniform noise standard deviation $\gamma = 0.01/0.05/0.10$ at all nodes. Other hyper-parameters and the training setup remain unchanged. The results in Table 11 demonstrate that the learnable noise variance approach is not particularly beneficial in comparison to a uni-



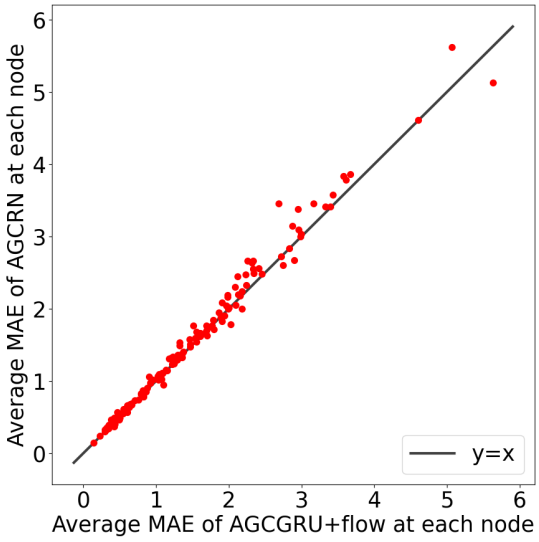
(a) PeMSD3



(b) PeMSD4



(c) PeMSD7



(d) PeMSD8

Figure 2. Scatter-plots of average MAE at each node for AGCGRU+flow v.s. that of AGCRN on PeMS datasets. The AGCGRU+flow has lower average MAE compared to AGCRN at most of the nodes for all four datasets.

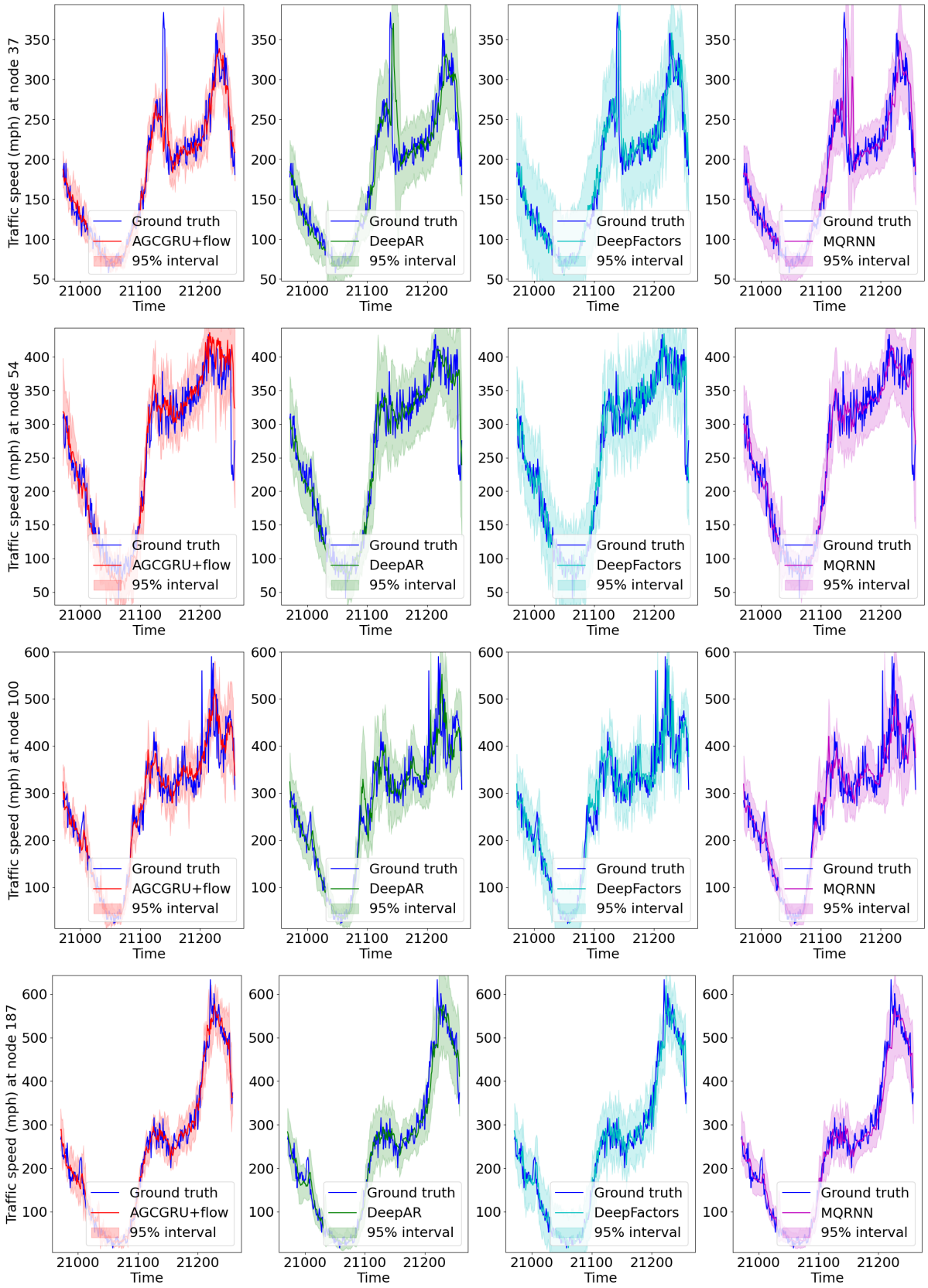


Figure 3. 15 minutes ahead predictions from the probabilistic forecasting algorithms with confidence intervals at nodes 37, 54, 100, and 187 of PeMSD3 dataset for the first day in the test set. The proposed AGCGRU+flow algorithm provides tighter confidence interval than its competitors in most cases, which leads to lower quantile error.

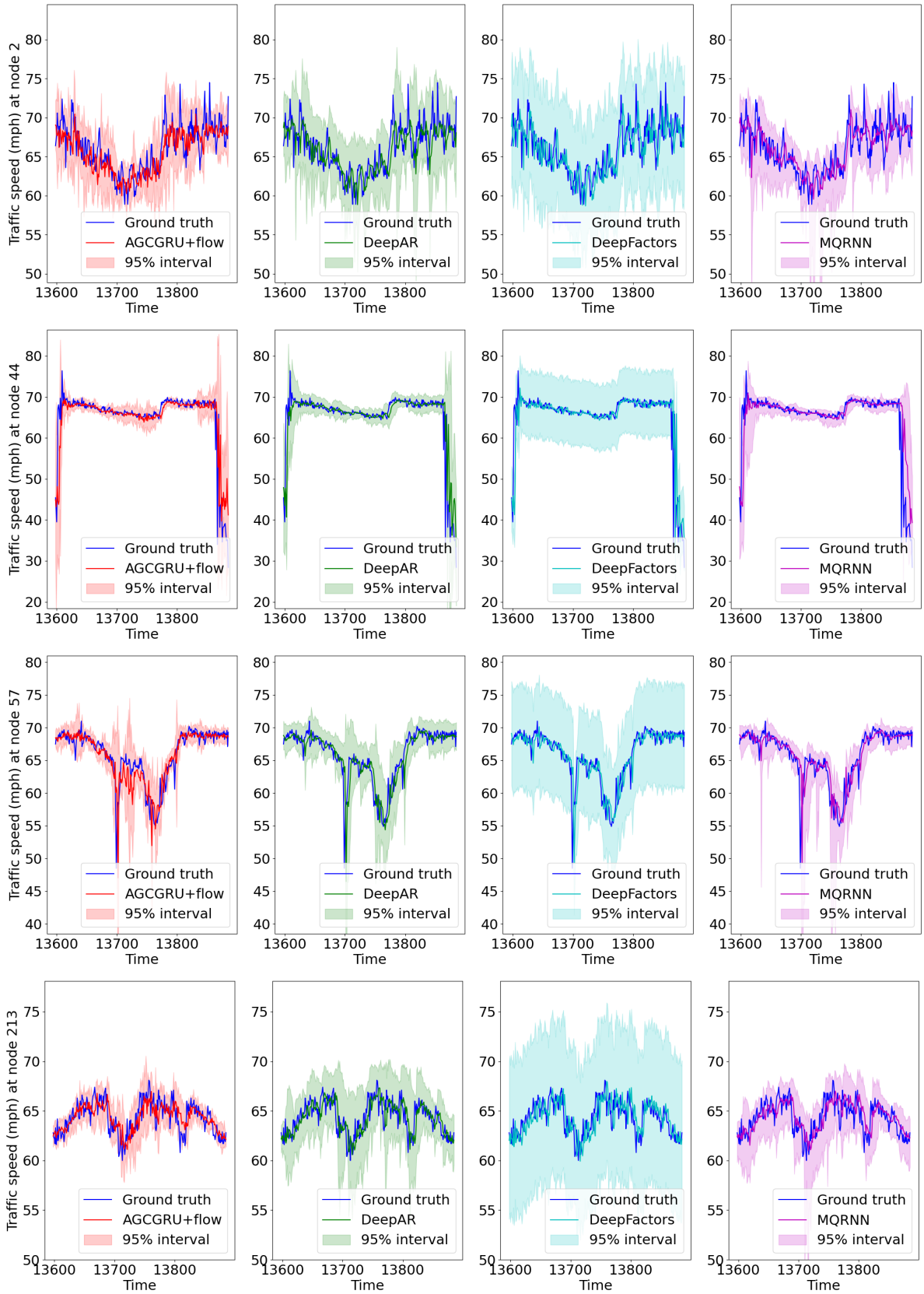


Figure 4. 15 minutes ahead predictions from the probabilistic forecasting algorithms with confidence intervals at nodes 2, 44, 57, and 213 of PeMSD4 dataset for the first day in the test set. The proposed AGCGRU+flow algorithm provides tighter confidence interval than its competitors in most cases, which leads to lower quantile error.

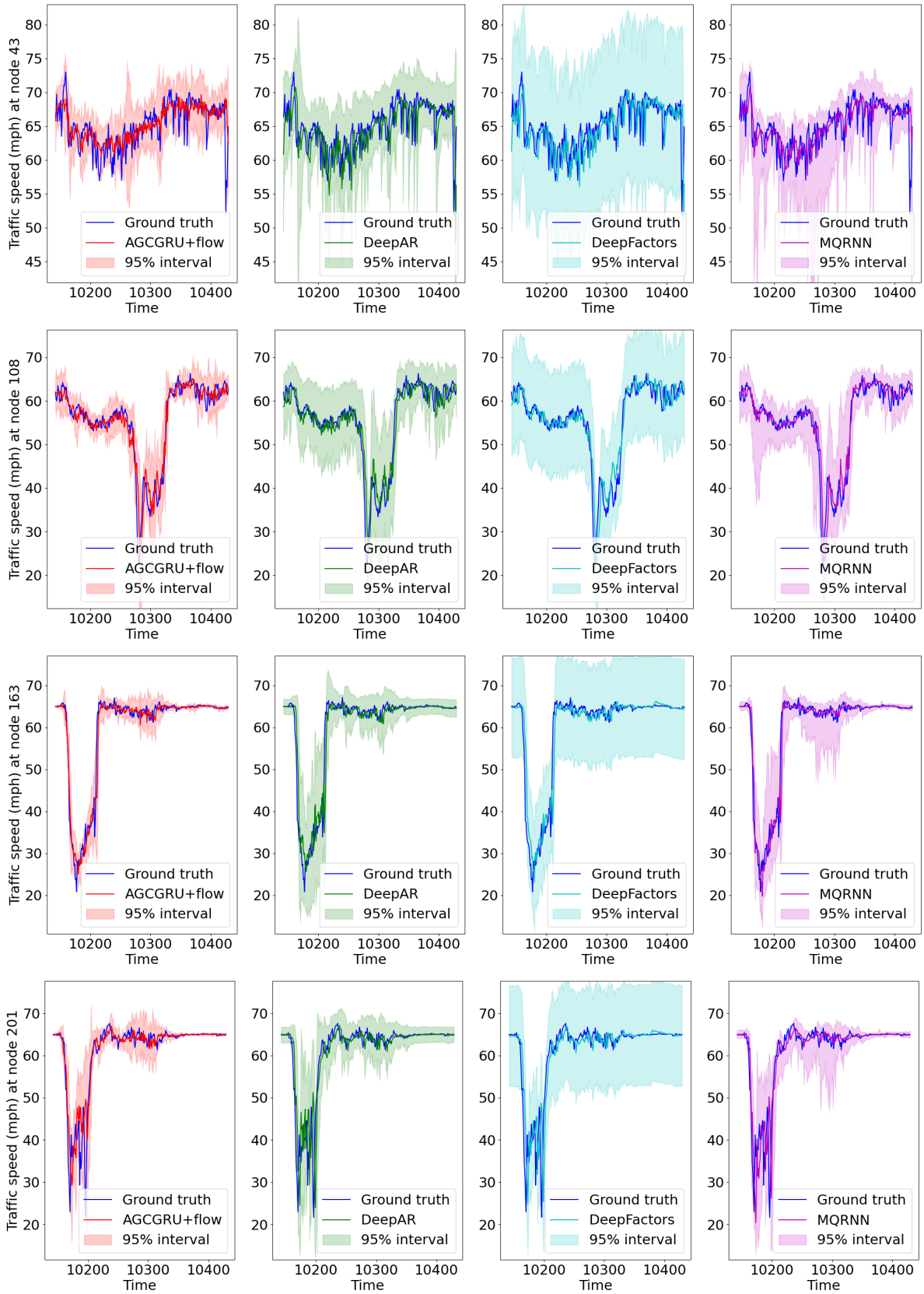


Figure 5. 15 minutes ahead predictions from the probabilistic forecasting algorithms with confidence intervals at nodes 43, 108, 163, and 201 of PeMSD7 dataset for the first day in the test set. The proposed AGCGRU+flow algorithm provides tighter confidence interval than its competitors in most cases, which leads to lower quantile error.

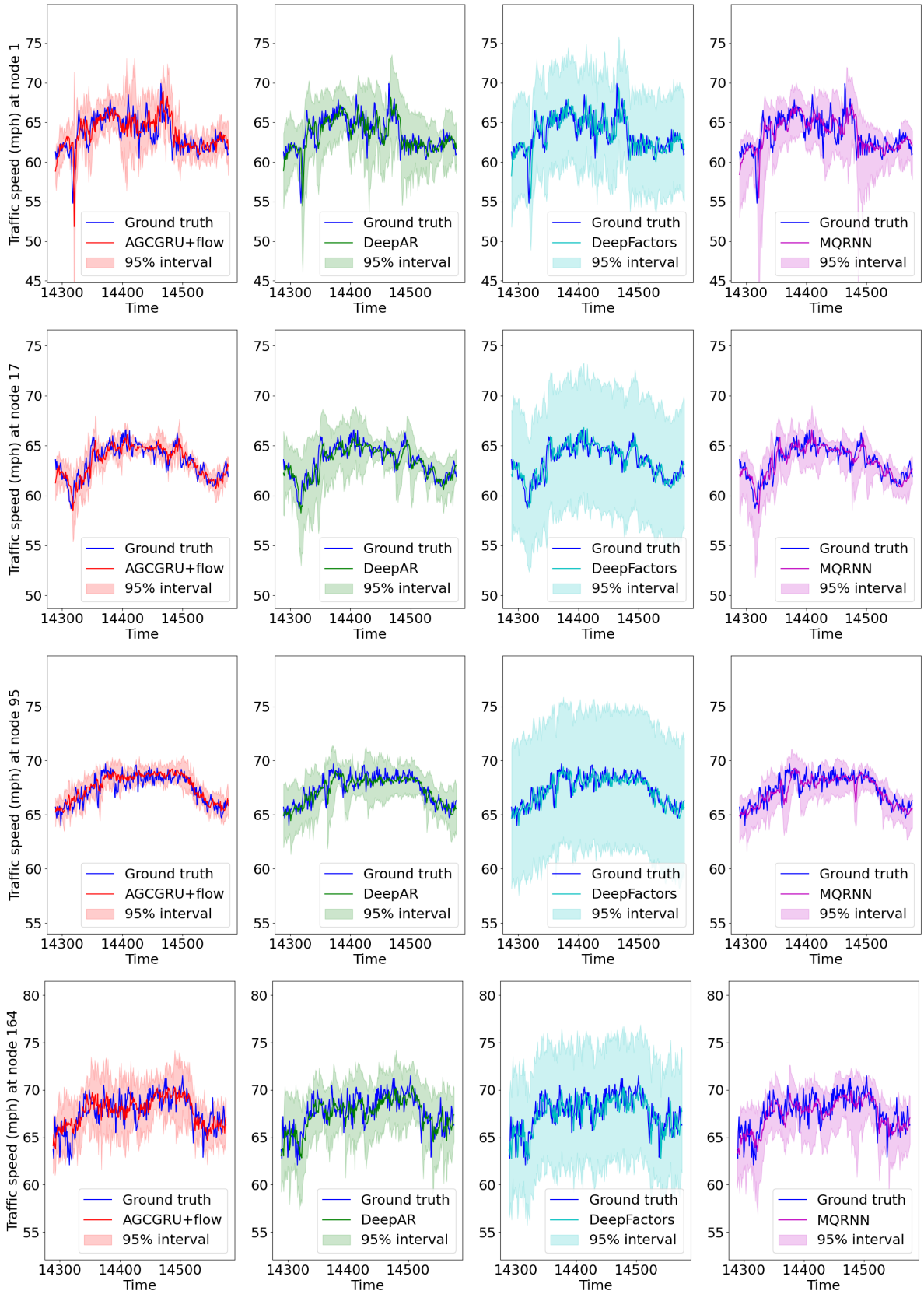


Figure 6. 15 minutes ahead predictions from the probabilistic forecasting algorithms with confidence intervals at nodes 1, 17, 95, and 164 of PeMSD8 dataset for the first day in the test set. The proposed AGCGRU+flow algorithm provides tighter confidence interval than its competitors in most cases, which leads to lower quantile error.

Table 7. Average MAE, MAPE, and RMSE for PeMSD3, PeMSD4, PeMSD7, and PeMSD8 for 15/30/45/60 minutes horizons for AGCGRU+flow and AGCGRU+BPF. Lower numbers are better.

Algorithm	PeMSD3 (15/ 30/ 45/ 60 min)		
	MAE	MAPE(%)	RMSE
AGCGRU+flow	**13.79/14.84/15.58/16.06**	**14.01/14.75/15.34/15.80**	**22.08/24.26/25.55/26.43**
AGCGRU+BPF	14.19/15.13/15.85/16.35	14.21/14.86/15.40/15.82	25.69/27.38/28.51/29.26
Algorithm	PeMSD4 (15/ 30/ 45/ 60 min)		
	MAE	MAPE(%)	RMSE
AGCGRU+flow	**1.35/1.63/1.78/1.88**	**2.67/3.44/3.87/4.16**	**2.88/3.77/4.20/4.46**
AGCGRU+BPF	1.36/1.65/1.80/1.90	2.71/3.46/3.90/4.18	2.91/3.81/4.25/4.52
Algorithm	PeMSD7 (15/ 30/ 45/ 60 min)		
	MAE	MAPE(%)	RMSE
AGCGRU+flow	**2.15/2.70**/2.99/3.19	**5.13/6.75/7.61/8.18**	**4.11/5.46/6.12/6.54**
AGCGRU+BPF	2.19/2.73/2.99/**3.17**	5.27/6.86/7.69/8.21	4.18/5.52/6.16/6.53
Algorithm	PeMSD8 (15/ 30/ 45/ 60 min)		
	MAE	MAPE(%)	RMSE
AGCGRU+flow	**1.13/1.37/1.49/1.57**	**2.30/3.01/3.40/3.65**	**2.59/3.45/3.85/4.09**
AGCGRU+BPF	1.18/1.41/1.52/1.59	2.47/3.13/3.50/3.74	2.69/3.53/3.92/4.15

Table 8. Average CRPS, P10QL, and P90QL for PeMSD3, PeMSD4, PeMSD7, and PeMSD8 for 15/30/45/60 minutes horizons for AGCGRU+flow and AGCGRU+BPF. Lower numbers are better.

Algorithm	PeMSD3 (15/ 30/ 45/ 60 min)		
	CRPS	P10QL(%)	P90QL(%)
AGCGRU+flow	**10.53/11.39/12.03/12.47**	**4.01/4.44/4.76/4.97**	**4.06/4.38/4.63/4.82**
AGCGRU+BPF	11.32/11.94/12.55/12.92	4.36/4.66/4.98/5.13	4.39/4.65/4.88/5.07
Algorithm	PeMSD4 (15/ 30/ 45/ 60 min)		
	CRPS	P10QL(%)	P90QL(%)
AGCGRU+flow	**1.08**/1.32/1.46/1.56	**1.28**/1.62/1.82/1.97	**1.05**/1.26/1.37/1.45
AGCGRU+BPF	1.10/1.32/**1.45/1.54**	1.29/**1.60/1.79/1.92**	1.06/1.26/1.37/1.45
Algorithm	PeMSD7 (15/ 30/ 45/ 60 min)		
	CRPS	P10QL(%)	P90QL(%)
AGCGRU+flow	**1.73/2.18/2.43/2.58**	**2.27/2.97/3.36/3.60**	**1.83/2.25/2.48/2.62**
AGCGRU+BPF	1.79/2.24/2.49/2.66	2.35/3.02/3.40/3.67	1.86/2.29/2.53/2.69
Algorithm	PeMSD8 (15/ 30/ 45/ 60 min)		
	CRPS	P10QL(%)	P90QL(%)
AGCGRU+flow	**0.90/1.10/1.20**/1.28	**1.10/1.43/1.61/1.73**	**0.87/1.01**/1.09/1.14
AGCGRU+BPF	0.96/1.13/1.22/1.28	1.19/1.47/1.63/1.74	0.91/1.03/1.09/**1.13**

Table 9. Average MAE, MAPE, and RMSE for PeMSD3, PeMSD4, PeMSD7, and PeMSD8 for 15/30/45/60 minutes horizons for AGCGRU+flow with different number of particles. Lower numbers are better.

Algorithm	PeMSD3 (15/ 30/ 45/ 60 min)		
	MAE	MAPE(%)	RMSE
AGCGRU+flow ($N_p = 1$)	13.82/14.87/15.60/16.08	14.04/14.78/15.36/15.82	22.33/24.41/25.70/26.54
AGCGRU+flow ($N_p = 10$)	**13.79/14.84/15.58/16.06**	**14.01**/14.75/15.34/15.80	22.08/24.26/**25.55**/26.43
AGCGRU+flow ($N_p = 50$)	**13.79/14.84/15.58/16.06**	**14.01/14.74/15.33/15.79**	**22.02/24.20/25.55/26.42**
Algorithm	PeMSD4 (15/ 30/ 45/ 60 min)		
	MAE	MAPE(%)	RMSE
AGCGRU+flow ($N_p = 1$)	1.35/1.63/1.78/1.88	2.68/3.45/3.89/4.18	2.89/3.78/4.22/4.47
AGCGRU+flow ($N_p = 10$)	1.35/1.63/1.78/1.88	**2.67/3.44/3.87/4.16**	**2.88/3.77/4.20**/4.46
AGCGRU+flow ($N_p = 50$)	1.35/1.63/1.78/1.88	**2.67/3.44/3.87/4.16**	**2.88/3.77/4.20/4.45**
Algorithm	PeMSD7 (15/ 30/ 45/ 60 min)		
	MAE	MAPE(%)	RMSE
AGCGRU+flow ($N_p = 1$)	2.16/2.71/3.00/3.20	5.14/6.77/7.63/8.20	4.12/5.47/6.14/6.56
AGCGRU+flow ($N_p = 10$)	**2.15/2.70/2.99/3.19**	5.13/**6.75/7.61/8.18**	**4.11/5.46/6.12/6.54**
AGCGRU+flow ($N_p = 50$)	**2.15/2.70/2.99/3.19**	**5.12/6.75/7.61/8.18**	**4.11/5.46/6.12/6.54**
Algorithm	PeMSD8 (15/ 30/ 45/ 60 min)		
	MAE	MAPE(%)	RMSE
AGCGRU+flow ($N_p = 1$)	1.14/1.38/1.50/1.57	2.31/3.02/3.41/3.67	2.60/3.46/3.87/4.11
AGCGRU+flow ($N_p = 10$)	**1.13/1.37/1.49**/1.57	**2.30/3.01/3.40/3.65**	**2.59**/3.45/**3.85/4.09**
AGCGRU+flow ($N_p = 50$)	**1.13/1.37/1.49**/1.57	**2.30/3.01/3.40/3.65**	**2.59/3.44/3.85/4.09**

Table 10. Average CRPS, P10QL, and P90QL for PeMSD3, PeMSD4, PeMSD7, and PeMSD8 for 15/30/45/60 minutes horizons for AGCGRU+flow with different number of particles. Lower numbers are better.

Algorithm	PeMSD3 (15/ 30/ 45/ 60 min)		
	CRPS	P10QL(%)	P90QL(%)
AGCGRU+flow ($N_p = 1$)	19.34/20.44/21.24/21.80	11.79/12.80/13.46/13.91	10.46/10.72/10.98/11.18
AGCGRU+flow ($N_p = 10$)	10.53/11.39/12.03/12.47	4.01/4.44/4.76/4.97	4.06/4.38/4.63/4.82
AGCGRU+flow ($N_p = 50$)	**10.02/10.86/11.49/11.92**	**3.67/4.05/4.33/4.53**	**3.83/4.15/4.41/4.59**
Algorithm	PeMSD4 (15/ 30/ 45/ 60 min)		
	CRPS	P10QL(%)	P90QL(%)
AGCGRU+flow ($N_p = 1$)	1.95/2.34/2.58/2.73	3.11/3.75/4.16/4.47	3.00/3.59/3.92/4.10
AGCGRU+flow ($N_p = 10$)	1.08/1.32/1.46/1.56	1.28/1.62/1.82/1.97	1.05/1.26/1.37/1.45
AGCGRU+flow ($N_p = 50$)	**1.03/1.26/1.40/1.49**	**1.21/1.54/1.73/1.87**	**0.98/1.17/1.27/1.35**
Algorithm	PeMSD7 (15/ 30/ 45/ 60 min)		
	CRPS	P10QL(%)	P90QL(%)
AGCGRU+flow ($N_p = 1$)	3.18/3.95/4.35/4.61	5.57/6.96/7.67/8.15	5.38/6.63/7.29/7.69
AGCGRU+flow ($N_p = 10$)	1.73/2.18/2.43/2.58	2.27/2.97/3.36/3.60	1.83/2.25/2.48/2.62
AGCGRU+flow ($N_p = 50$)	**1.64/2.09/2.32/2.47**	**2.16/2.83/3.20/3.44**	**1.71/2.10/2.31/2.45**
Algorithm	PeMSD8 (15/ 30/ 45/ 60 min)		
	CRPS	P10QL(%)	P90QL(%)
AGCGRU+flow ($N_p = 1$)	1.63/1.90/2.07/2.18	2.73/3.28/3.63/3.87	2.38/2.68/2.86/2.98
AGCGRU+flow ($N_p = 10$)	0.90/1.10/1.20/1.28	1.10/1.43/1.61/1.73	0.87/1.01/1.09/1.14
AGCGRU+flow ($N_p = 50$)	**0.86/1.05/1.16/1.22**	**1.04/1.35/1.52/1.63**	**0.83/0.95/1.03/1.08**

form, fixed variance approach in most cases. However, we note that the probabilistic metrics reported in Table 12 are the lowest for the learnable noise variance model in all cases. This suggests that different time-series in these road traffic datasets have different degrees of uncertainty which cannot be effectively modelled by the uniform, fixed noise variance approach.

5.5. Detailed comparison with deterministic encoder-decoder models

In Table 2 of the main paper, we compare the average MAE of the proposed flow based approaches with those of the deterministic encoder-decoder based sequence to sequence prediction models for three different RNN architectures. In Table 13, we report the MAPE and RMSE, in addition to the MAE. We see that the particle flow based RNN models outperform the corresponding deterministic encoder-decoder models in most cases.

5.6. Detailed results for comparison to ensembles

In Table 5 of the main paper, we compare the average CRPS of the proposed AGCGRU+flow algorithm with ensembles of AGCRN and GAMN. From Table 14, we observe that our approach is comparable or slightly worse compared to the ensembles in terms of the MAE, MAPE and RMSE of the point forecasts. However, the proposed AGCGRU+flow shows better characterization of the prediction uncertainty compared to the ensemble methods in almost all cases, as shown in Table 15.

5.7. Comparison with a Variational Inference (VI) based approach

Although there is no directly applicable baseline forecasting method in the literature that incorporates VI, RNNs, and GNNs, we can derive a variational approach using equivalent GNN-RNN architectures and compare it to the particle flow approach. We wish to approximate $p_\Theta(\mathbf{y}_{P+1:P+Q}|\mathbf{y}_{1:P}, \mathbf{z}_{1:P+Q})$. So, the ELBO is defined as follows:

$$\mathcal{L}(\Theta, \Omega) = \mathbb{E}_{q_\Omega}\Big[\log p_\Theta(\mathbf{y}_{P+1:P+Q}, \mathbf{x}_{1:P}|\mathbf{y}_{1:P}, \mathbf{z}_{1:P+Q}) - \log q_\Omega(\mathbf{x}_{1:P}|\mathbf{y}_{1:P+Q}, \mathbf{z}_{1:P+Q})\Big]. \quad (21)$$

Now, we approximate

$$\begin{aligned} &p_\Theta(\mathbf{y}_{P+1:P+Q}, \mathbf{x}_{1:P}|\mathbf{y}_{1:P}, \mathbf{z}_{1:P+Q}) \\ &= \int \prod_{t=P+1}^{P+Q} \Big(p_{\phi,\gamma}(\mathbf{y}_t|\mathbf{x}_t, \mathbf{z}_t) \\ &\qquad p_{\psi,\sigma}(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{y}_{t-1}, \mathbf{z}_t)\Big) d\mathbf{x}_{P+1:P+Q}\,, \\ &\approx \prod_{t=P+1}^{P+Q} \left[\frac{1}{N_p}\sum_{j=1}^{N_p} p_{\phi,\gamma}(\mathbf{y}_t|\mathbf{x}_t^j, \mathbf{z}_t)\right], \end{aligned} \quad (22)$$

where, in the decoder, we first sample $\mathbf{x}_t^j$ from $p_{\psi,\sigma}(\mathbf{x}_t|\mathbf{x}_{t-1}^j, \mathbf{y}_{t-1}^j, \mathbf{z}_t)$ (for $t > P+1$) or from $p_{\psi,\sigma}(\mathbf{x}_t|\mathbf{x}_{t-1}^j, \mathbf{y}_{t-1}, \mathbf{z}_t)$ (for $t = P+1$) for $1 \leqslant j \leqslant N_p$ and then sample $\mathbf{y}_t^j$ from $p_{\phi,\gamma}(\mathbf{y}_t|\mathbf{x}_t^j, \mathbf{z}_t)$ for $1 \leqslant j \leqslant N_p$ to form the MC approximation. This decoder is initialized using the output of the encoder, i.e., we sample $\mathbf{x}_{1:P}^j$ from the inference distribution $q_\Omega(\mathbf{x}_{1:P}|\mathbf{y}_{1:P+Q}, \mathbf{z}_{1:P+Q})$ for $1 \leqslant j \leqslant N_p$, which is assumed to be factorized as follows:

$$\begin{aligned} &q_\Omega(\mathbf{x}_{1:P}|\mathbf{y}_{1:P+Q}, \mathbf{z}_{1:P+Q}) \\ &\quad = q_\Omega(\mathbf{x}_{1:P}|\mathbf{y}_{1:P}, \mathbf{z}_{1:P})\,, \\ &\quad = q_1(\mathbf{x}_1, \mathbf{z}_1, \rho)\prod_{t=2}^{P} q_{\psi',\sigma'}(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{y}_{t-1}, \mathbf{z}_t)\,. \end{aligned} \quad (23)$$

Here, we set $q_1(\mathbf{x}_1, \mathbf{z}_1, \rho) = p_1(\mathbf{x}_1, \mathbf{z}_1, \rho)$ for simplicity and we use the same RNN architecture (i.e. AGCGRU) for $q_{\psi',\sigma'}$ and $p_{\psi,\sigma}$.

Experimental details : We treat ρ, σ and σ' as hyperparameters and set $\rho = 1$ and $\sigma = \sigma' = 0$. This implies that $q_{\psi',\sigma'}$ is a Dirac-delta function and the maximization of ELBO (in eq. (21)) using SGD (SGVI) amounts to mimization of the same cost function as defined in eq. (14) in the main paper. The only difference is that now a) we have two separate AGCGRUs for encoder and decoder and b) there is no particle flow in the forward pass. We call this model AGCGRU+VI and compare it to AGCGRU+flow. The other hyperparameters are set to the same values as for the AGCGRU+flow algorithm. From Table 16, we observe that for comparable RNN architectures, the flow based algorithm significantly outperforms the variational inference based approach in the point forecasting task. The results in Table 17 indicate that in the probabilistic forecasting task, both particle flow and VI approaches show comparable performance despite AGCGRU+flow having approximately half of the learnable parameters of the AGCGRU+VI model.

5.8. Comparison of execution time, GPU memory usage and model size

Table 18 summarizes the run time, GPU usage during training, and the size of the learned model for AGCRN-ensemble, GMAN-ensemble, and the proposed AGCGRU+flow for the four PeMS datasets. We observe that if we choose the ensemble size so that the algorithms have an approximately equal execution time, then the model-size of the ensemble algorithms are comparable to our approach as well. However, our method requires more GPU memory compared to the ensembles during training because of the particle flow in the forward pass.

Table 11. Average MAE, MAPE, and RMSE for PeMSD3, PeMSD4, PeMSD7, and PeMSD8 for 15/30/45/60 minutes horizons for AGCGRU+flow with learnable and fixed noise variance settings. The best result in each column is shown in bold. Lower numbers are better.

Algorithm	PeMSD3 (15/ 30/ 45/ 60 min)		
	MAE	MAPE(%)	RMSE
AGCGRU+flow (learnable)	13.79/14.84/15.58/16.06	**14.01/14.75/15.34/15.80**	22.08/24.26/25.55/26.43
AGCGRU+flow ($\gamma = 0.01$)	**13.68/14.75/15.49/16.02**	14.57/15.37/16.02/16.57	**21.74/23.95/25.27/26.21**
AGCGRU+flow ($\gamma = 0.05$)	13.96/15.05/15.76/16.25	15.87/16.66/17.23/17.62	22.08/24.33/25.64/26.54
AGCGRU+flow ($\gamma = 0.10$)	13.86/14.91/15.68/16.17	14.42/15.20/15.87/16.39	22.04/24.25/25.60/26.41
Algorithm	PeMSD4 (15/ 30/ 45/ 60 min)		
	MAE	MAPE(%)	RMSE
AGCGRU+flow (learnable)	**1.35/1.63/1.78/1.88**	**2.67/3.44/3.87/4.16**	**2.88**/3.77/**4.20/4.46**
AGCGRU+flow ($\gamma = 0.01$)	**1.35/1.63**/1.79/1.89	2.68/3.45/3.89/4.20	**2.88**/3.77/**4.20**/4.47
AGCGRU+flow ($\gamma = 0.05$)	1.36/1.65/1.80/1.91	2.69/3.47/3.91/4.21	**2.88/3.76/4.20/4.46**
AGCGRU+flow ($\gamma = 0.10$)	1.36/1.65/1.80/1.90	2.70/3.47/3.89/4.18	2.92/3.81/4.24/4.49
Algorithm	PeMSD7 (15/ 30/ 45/ 60 min)		
	MAE	MAPE(%)	RMSE
AGCGRU+flow (learnable)	2.15/2.70/2.99/3.19	5.13/6.75/7.61/8.18	4.11/5.46/6.12/6.54
AGCGRU+flow ($\gamma = 0.01$)	**2.14/2.69/2.98/3.16**	**5.07/6.66/7.47/8.00**	4.10/5.43/6.09/6.49
AGCGRU+flow ($\gamma = 0.05$)	2.16/2.71/3.00/3.20	5.13/6.74/7.61/8.19	**4.09/5.41/6.06/6.48**
AGCGRU+flow ($\gamma = 0.10$)	2.16/2.73/3.01/3.20	5.15/6.77/7.62/8.15	4.12/5.48/6.15/6.54
Algorithm	PeMSD8 (15/ 30/ 45/ 60 min)		
	MAE	MAPE(%)	RMSE
AGCGRU+flow (learnable)	1.13/**1.37/1.49/1.57**	2.30/3.01/3.40/3.65	2.59/3.45/3.85/4.09
AGCGRU+flow ($\gamma = 0.01$)	1.13/**1.37/1.49/1.57**	2.31/3.03/3.44/3.71	2.60/3.43/3.84/4.09
AGCGRU+flow ($\gamma = 0.05$)	1.13/**1.37/1.49/1.57**	**2.26/2.95/3.35/3.62**	**2.53/3.34/3.75/4.01**
AGCGRU+flow ($\gamma = 0.10$)	1.13/1.38/1.51/1.60	2.31/3.04/3.49/3.80	2.57/3.41/3.86/4.14

Table 12. Average CRPS, P10QL, and P90QL for PeMSD3, PeMSD4, PeMSD7, and PeMSD8 for 15/30/45/60 minutes horizons for AGCGRU+flow with learnable and fixed noise variance settings. The best result in each column is shown in bold. Lower numbers are better.

Algorithm	PeMSD3 (15/ 30/ 45/ 60 min)		
	CRPS	P10QL(%)	P90QL(%)
AGCGRU+flow (learnable)	**10.53/11.39/12.03/12.47**	**4.01/4.44/4.76/4.97**	**4.06/4.38/4.63/4.82**
AGCGRU+flow ($\gamma = 0.01$)	12.83/13.90/14.63/15.17	7.26/8.10/8.46/8.77	6.68/7.08/7.55/7.86
AGCGRU+flow ($\gamma = 0.05$)	11.58/12.61/13.28/13.74	5.78/6.52/6.99/7.25	5.14/5.54/5.81/6.06
AGCGRU+flow ($\gamma = 0.10$)	13.14/14.18/14.95/15.43	7.79/8.57/9.22/9.53	6.64/7.05/7.28/7.53
Algorithm	PeMSD4 (15/ 30/ 45/ 60 min)		
	CRPS	P10QL(%)	P90QL(%)
AGCGRU+flow (learnable)	**1.08/1.32/1.46/1.56**	**1.28/1.62/1.82/1.97**	**1.05/1.26/1.37/1.45**
AGCGRU+flow ($\gamma = 0.01$)	1.28/1.55/1.70/1.81	2.09/2.58/2.87/3.08	1.74/2.08/2.26/2.38
AGCGRU+flow ($\gamma = 0.05$)	1.19/1.47/1.62/1.72	1.82/2.30/2.57/2.77	1.48/1.84/2.04/2.15
AGCGRU+flow ($\gamma = 0.10$)	1.32/1.60/1.75/1.85	2.19/2.68/2.95/3.15	1.84/2.23/2.43/2.54
Algorithm	PeMSD7 (15/ 30/ 45/ 60 min)		
	CRPS	P10QL(%)	P90QL(%)
AGCGRU+flow (learnable)	**1.73/2.18/2.43/2.58**	**2.27/2.97/3.36/3.60**	**1.83/2.25/2.48/2.62**
AGCGRU+flow ($\gamma = 0.01$)	2.02/2.55/2.82/3.01	3.59/4.57/5.05/5.35	3.00/3.77/4.22/4.54
AGCGRU+flow ($\gamma = 0.05$)	1.90/2.42/2.70/2.90	3.18/4.20/4.76/5.15	2.56/3.27/3.65/3.91
AGCGRU+flow ($\gamma = 0.10$)	2.09/2.65/2.94/3.12	3.80/4.87/5.41/5.77	3.18/4.04/4.47/4.73
Algorithm	PeMSD8 (15/ 30/ 45/ 60 min)		
	CRPS	P10QL(%)	P90QL(%)
AGCGRU+flow (learnable)	**0.90/1.10/1.20/1.28**	**1.10/1.43/1.61/1.73**	**0.87/1.01/1.09/1.14**
AGCGRU+flow ($\gamma = 0.01$)	1.07/1.29/1.41/1.49	1.81/2.29/2.56/2.75	1.35/1.57/1.67/1.73
AGCGRU+flow ($\gamma = 0.05$)	1.00/1.23/1.35/1.43	1.58/2.04/2.31/2.50	1.21/1.43/1.52/1.58
AGCGRU+flow ($\gamma = 0.10$)	1.10/1.34/1.47/1.56	1.88/2.41/2.72/2.93	1.47/1.71/1.81/1.87

Table 13. Average MAE, MAPE, and RMSE for PeMSD3, PeMSD4, PeMSD7, and PeMSD8 for 15/30/45/60 minutes horizons for the proposed flow based approach and deterministic encoder-decoder models. Lower numbers are better.

Algorithm	PeMSD3 (15/ 30/ 45/ 60 min)		
	MAE	MAPE(%)	RMSE
AGCGRU+flow	**13.79/14.84/15.58/16.06**	**14.01/14.75/15.34/15.80**	**22.08/24.26/25.55/26.43**
FC-AGCGRU	13.96/15.37/16.52/17.45	14.26/15.61/16.69/17.37	25.28/27.43/29.09/30.43
DCGRU+flow	14.48/**15.67/16.52/17.36**	15.06/16.06/16.91/**17.84**	**23.86/26.12/27.54/28.76**
FC-DCGRU	**14.42**/15.87/17.10/18.29	**14.57/15.78/16.87**/17.95	24.33/27.05/28.99/30.76
GRU+flow	**14.40/16.10/17.63/19.18**	**14.56/15.99/17.33/18.89**	**23.06/26.15/28.64/30.97**
FC-GRU	15.82/18.37/20.61/22.93	15.87/18.82/21.32/23.75	25.85/30.09/33.37/36.94
Algorithm	PeMSD4 (15/ 30/ 45/ 60 min)		
	MAE	MAPE(%)	RMSE
AGCGRU+flow	**1.35/1.63/1.78/1.88**	**2.67/3.44/3.87/4.16**	**2.88/3.77/4.20/4.46**
FC-AGCGRU	1.37/1.74/2.00/2.20	2.69/3.67/4.41/5.00	2.92/3.96/4.62/5.09
DCGRU+flow	**1.38/1.71/1.92/2.08**	2.72/**3.63/4.23/4.67**	**2.93/3.93/4.49/4.87**
FC-DCGRU	**1.38**/1.78/2.06/2.29	**2.69**/3.72/4.51/5.16	2.95/4.09/4.81/5.34
GRU+flow	**1.37/1.76/2.02/2.23**	**2.70/3.74/4.52/5.15**	**2.95/4.05/4.74/5.23**
FC-GRU	1.46/1.91/2.25/2.54	2.84/3.97/4.88/5.66	3.10/4.35/5.20/5.85
Algorithm	PeMSD7 (15/ 30/ 45/ 60 min)		
	MAE	MAPE(%)	RMSE
AGCGRU+flow	**2.15/2.70/2.99/3.19**	**5.13/6.75/7.61/8.18**	**4.11/5.46/6.12/6.54**
FC-AGCGRU	2.21/2.99/3.56/4.05	5.18/7.39/9.12/10.64	4.18/5.88/7.03/7.94
DCGRU+flow	**2.19/2.87/3.29/3.61**	**5.16/7.17/8.48/9.42**	**4.16/5.66/6.54/7.14**
FC-DCGRU	2.23/3.06/3.67/4.18	5.19/7.50/9.31/10.90	4.26/6.05/7.28/8.24
GRU+flow	**2.24/3.02/3.55/3.96**	**5.27/7.58/9.30/10.60**	**4.28/5.97/7.00/7.73**
FC-GRU	2.41/3.40/4.17/4.84	5.60/8.27/10.47/12.40	4.56/6.68/8.17/9.34
Algorithm	PeMSD8 (15/ 30/ 45/ 60 min)		
	MAE	MAPE(%)	RMSE
AGCGRU+flow	**1.13/1.37/1.49/1.57**	**2.30/3.01/3.40/3.65**	2.59/**3.45/3.85/4.09**
FC-AGCGRU	1.16/1.48/1.70/1.87	**2.30**/3.17/3.78/4.25	**2.58**/3.53/4.12/4.54
DCGRU+flow	1.17/**1.44/1.58/1.70**	2.35/**3.12/3.57/3.87**	2.64/3.54/**4.00/4.28**
FC-DCGRU	**1.16**/1.49/1.70/1.87	**2.25**/3.16/3.85/4.37	**2.54/3.49**/4.08/4.49
GRU+flow	**1.12/1.41/1.59/1.74**	**2.17/2.94/3.50/3.92**	**2.55/3.47/4.02/4.40**
FC-GRU	1.20/1.56/1.81/2.02	2.29/3.09/3.70/4.22	2.63/3.61/4.24/4.73

Table 14. Average MAE, MAPE, and RMSE for PeMSD3, PeMSD4, PeMSD7, and PeMSD8 for 15/30/45/60 minutes horizons for AGCRN-ensemble, GMAN-ensemble, and AGCGRU+flow. The best and the second best results in each column are shown in bold and marked with underline respectively. Lower numbers are better.

Algorithm	PeMSD3 (15/ 30/ 45/ 60 min)		
	MAE	MAPE(%)	RMSE
AGCRN-ensemble	14.21/15.12/15.73/16.22	**13.91/14.56/14.93/15.38**	25.49/27.16/28.20/28.90
GMAN-ensemble	14.48/15.20/15.90/16.66	15.01/15.64/16.41/17.36	23.96/25.20/26.31/27.44
AGCGRU+flow	**13.79/14.84/15.58/16.06**	14.01/14.75/15.34/15.80	**22.08/24.26/25.55/26.43**
Algorithm	PeMSD4 (15/ 30/ 45/ 60 min)		
	MAE	MAPE(%)	RMSE
AGCRN-ensemble	1.35/1.61/1.76/1.91	2.75/3.40/3.79/4.17	2.89/3.65/4.09/4.47
GMAN-ensemble	**1.33/1.57/1.72/1.84**	**2.64/3.27/3.70/4.04**	2.89/**3.62/4.04/4.33**
AGCGRU+flow	1.35/1.63/1.78/1.88	2.67/3.44/3.87/4.16	**2.88**/3.77/4.20/4.46
Algorithm	PeMSD7 (15/ 30/ 45/ 60 min)		
	MAE	MAPE(%)	RMSE
AGCRN-ensemble	2.17/**2.69/2.95**/3.20	5.25/**6.75/7.55**/8.22	**4.09/5.29/5.94/6.45**
GMAN-ensemble	2.42/2.80/3.08/3.35	6.08/7.18/8.00/8.74	4.68/5.54/6.08/6.51
AGCGRU+flow	**2.15**/2.70/2.99/**3.19**	**5.13/6.75**/7.61/**8.18**	4.11/5.46/6.12/6.54
Algorithm	PeMSD8 (15/ 30/ 45/ 60 min)		
	MAE	MAPE(%)	RMSE
AGCRN-ensemble	1.19/1.36/1.46/1.58	2.67/3.10/3.38/3.68	2.88/3.41/3.76/4.06
GMAN-ensemble	**1.13/1.28/1.39/1.49**	2.37/**2.78/3.10/3.37**	2.71/**3.25/3.61/3.87**
AGCGRU+flow	**1.13**/1.37/1.49/1.57	**2.30**/3.01/3.40/3.65	**2.59**/3.45/3.85/4.09

Table 15. Average CRPS, P10QL, and P90QL for PeMSD3, PeMSD4, PeMSD7, and PeMSD8 for 15/30/45/60 minutes horizons for AGCRN-ensemble, GMAN-ensemble, and AGCGRU+flow. The best and the second best results in each column are shown in bold and marked with underline respectively. Lower numbers are better.

Algorithm	PeMSD3 (15/ 30/ 45/ 60 min)		
	CRPS	P10QL(%)	P90QL(%)
AGCRN-ensemble	12.64/13.44/13.96/14.27	6.90/7.40/7.54/7.53	6.10/6.43/6.79/6.96
GMAN-ensemble	12.79/13.49/14.13/14.77	7.17/7.67/8.08/8.45	5.86/6.16/6.44/6.68
AGCGRU+flow	**10.53/11.39/12.03/12.47**	**4.01/4.44/4.76/4.97**	**4.06/4.38/4.63/4.82**
Algorithm	PeMSD4 (15/ 30/ 45/ 60 min)		
	CRPS	P10QL(%)	P90QL(%)
AGCRN-ensemble	1.20/1.44/1.56/1.68	1.82/2.21/2.39/2.57	1.53/1.82/1.93/2.08
GMAN-ensemble	1.16/1.38/1.51/1.62	1.73/2.11/2.35/2.54	1.45/1.70/1.82/1.92
AGCGRU+flow	**1.08/1.32/1.46/1.56**	**1.28/1.62/1.82/1.97**	**1.05/1.26/1.37/1.45**
Algorithm	PeMSD7 (15/ 30/ 45/ 60 min)		
	CRPS	P10QL(%)	P90QL(%)
AGCRN-ensemble	1.90/2.39/2.60/2.81	3.22/4.15/4.55/4.89	2.55/3.19/3.35/3.58
GMAN-ensemble	1.96/2.31/2.53/2.73	3.16/3.83/4.23/4.53	2.20/2.59/2.81/3.00
AGCGRU+flow	**1.73/2.18/2.43/2.58**	**2.27/2.97/3.36/3.60**	**1.83/2.25/2.48/2.62**
Algorithm	PeMSD8 (15/ 30/ 45/ 60 min)		
	CRPS	P10QL(%)	P90QL(%)
AGCRN-ensemble	1.03/1.20/1.28/1.38	1.63/1.97/2.14/2.34	1.18/1.34/1.39/1.48
GMAN-ensemble	0.95/**1.10/1.19/1.28**	1.40/1.68/1.88/2.04	1.12/1.26/1.34/1.41
AGCGRU+flow	**0.90/1.10**/1.20/**1.28**	**1.10/1.43/1.61/1.73**	**0.87/1.01/1.09/1.14**

Table 16. Average MAE, MAPE, and RMSE for PeMSD3, PeMSD4, PeMSD7, and PeMSD8 for 15/30/45/60 minutes horizons for AGCGRU+flow and AGCGRU+VI. Lower numbers are better.

Algorithm	PeMSD3 (15/ 30/ 45/ 60 min)		
	MAE	MAPE(%)	RMSE
AGCGRU+flow	**13.79/14.84/15.58/16.06**	**14.01/14.75/15.34/15.80**	**22.08/24.26/25.55/26.43**
AGCGRU+VI	15.08/16.10/16.83/17.53	15.26/16.10/16.74/17.43	26.17/28.02/29.13/30.17
Algorithm	PeMSD4 (15/ 30/ 45/ 60 min)		
	MAE	MAPE(%)	RMSE
AGCGRU+flow	**1.35/1.63/1.78/1.88**	**2.67/3.44/3.87/4.16**	**2.88/3.77/4.20/4.46**
AGCGRU+VI	1.46/1.76/1.94/2.06	2.94/3.73/4.20/4.52	2.97/3.78/4.22/4.48
Algorithm	PeMSD7 (15/ 30/ 45/ 60 min)		
	MAE	MAPE(%)	RMSE
AGCGRU+flow	**2.15/2.70/2.99/3.19**	**5.13/6.75/7.61/8.18**	**4.11/5.46**/6.12/6.54
AGCGRU+VI	2.33/2.92/3.23/3.45	5.59/7.26/8.16/8.78	4.22/5.48/**6.10/6.50**
Algorithm	PeMSD8 (15/ 30/ 45/ 60 min)		
	MAE	MAPE(%)	RMSE
AGCGRU+flow	**1.13/1.37/1.49/1.57**	**2.30/3.01/3.40/3.65**	**2.59/3.45/3.85/4.09**
AGCGRU+VI	1.29/1.52/1.65/1.74	2.94/3.51/3.86/4.10	2.96/3.59/3.94/4.17

Table 17. Average CRPS, P10QL, and P90QL for PeMSD3, PeMSD4, PeMSD7, and PeMSD8 for 15/30/45/60 minutes horizons for AGCGRU+flow and AGCGRU+VI. Lower numbers are better.

Algorithm	PeMSD3 (15/ 30/ 45/ 60 min)		
	CRPS	P10QL(%)	P90QL(%)
AGCGRU+flow	**10.53/11.39/12.03/12.47**	**4.01/4.44/4.76/4.97**	**4.06/4.38/4.63/4.82**
AGCGRU+VI	11.00/11.80/12.38/12.94	4.14/4.53/4.82/5.10	4.27/4.58/4.81/5.02
Algorithm	PeMSD4 (15/ 30/ 45/ 60 min)		
	CRPS	P10QL(%)	P90QL(%)
AGCGRU+flow	1.08/1.32/1.46/1.56	1.28/1.62/1.82/1.97	1.05/1.26/1.37/1.45
AGCGRU+VI	1.08/**1.31/1.45/1.54**	**1.26/1.59/1.79/1.93**	**1.04/1.25/1.36**/1.45
Algorithm	PeMSD7 (15/ 30/ 45/ 60 min)		
	CRPS	P10QL(%)	P90QL(%)
AGCGRU+flow	1.73/2.18/2.43/**2.58**	2.27/2.97/**3.36/3.60**	1.83/2.25/2.48/**2.62**
AGCGRU+VI	**1.72**/2.18/**2.42**/2.60	**2.25**/2.97/3.39/3.66	**1.80/2.24/2.47**/2.63
Algorithm	PeMSD8 (15/ 30/ 45/ 60 min)		
	CRPS	P10QL(%)	P90QL(%)
AGCGRU+flow	**0.90/1.10/1.20/1.28**	**1.10/1.43/1.61/1.73**	**0.87/1.01/1.09/1.14**
AGCGRU+VI	0.95/1.13/1.24/1.31	1.15/1.44/1.62/1.76	0.90/1.03/1.10/1.15

Table 18. Execution time, memory consumption (during training) and model size for AGCRN-ensemble, GMAN-ensemble and AGCGRU+flow for the four PeMS datasets. Lower numbers are better.

Algorithm	Execution time (minutes)			
	PEMS03	PEMS04	PEMS07	PEMS08
AGCRN-ensemble	369	243	183	224
GMAN-ensemble	444	224	195	185
AGCGRU+flow	**325**	**205**	**154**	**177**
Algorithm	GPU memory (GB)			
	PEMS03	PEMS04	PEMS07	PEMS08
AGCRN-ensemble	**6.55**	**5.19**	**4.09**	**3.47**
GMAN-ensemble	15.45	9.45	8.46	4.45
AGCGRU+flow	25.27	18.76	12.45	8.45
Algorithm	Model Size (MB)			
	PEMS03	PEMS04	PEMS07	PEMS08
AGCRN-ensemble	11.52	11.52	11.45	11.45
GMAN-ensemble	**9.54**	**9.51**	**9.45**	**9.35**
AGCGRU+flow	12.88	12.86	12.86	12.85

References

Bai, L., Yao, L., Li, C., Wang, X., and Wang, C. Adaptive graph convolutional recurrent network for traffic forecasting. In *Proc. Adv. Neural Info. Process. Systems*, 2020.

Bengtsson, T., Bickel, P., and Li, B. Curse-of-dimensionality revisited: Collapse of the particle filter in very large scale systems. In *Probability and Statistics: Essays in Honor of David A. Freedman*, volume 2, pp. 316–334. Institute of Mathematical Statistics, Beachwood, OH, USA, Apr. 2008.

Beskos, A., Crisan, D., and Jasra, A. On the stability of sequential Monte Carlo methods in high dimensions. *Ann. Appl. Prob.*, 24(4):1396–1445, 2014.

Choi, S., Willett, P., Daum, F., and Huang, J. Discussion and application of the homotopy filter. In *Proc. SPIE Conf. Signal Process., Sensor Fusion, Target Recog.*, pp. 805021, Orlando, FL, USA, May 2011.

Chun-Hsin, W., Jan-Ming, H., and T., L. D. Travel-time prediction with support vector regression. *IEEE Trans. Intell. Transport. Systems*, 5(4):276–281, 2004.

Daum, F. and Huang, J. Nonlinear filters with log-homotopy. In *Proc. SPIE Signal and Data Process. Small Targets*, pp. 669918, San Diego, CA, USA, Sep. 2007.

Daum, F. and Huang, J. Seven dubious methods to mitigate stiffness in particle flow with non-zero diffusion for nonlinear filters, Bayesian decisions, and transport. In *Proc. SPIE Conf. Signal Process., Sensor Fusion, Target Recog.*, pp. 90920C, Baltimore, MD, USA, May 2014.

Daum, F., Huang, J., and Noushin, A. Exact particle flow for nonlinear filters. In *Proc. SPIE Conf. Signal Process., Sensor Fusion, Target Recog.*, pp. 769704, Orlando, FL, USA, Apr. 2010.

Daum, F., Huang, J., and Noushin, A. Generalized Gromov method for stochastic particle flow filters. In *Proc. SPIE Conf. Signal Process., Sensor Fusion, Target Recog.*, pp. 102000I, Anaheim, CA, USA, May 2017.

Ding, T. and Coates, M. J. Implementation of the daum-huang exact-flow particle filter. In *Proc. IEEE Statist. Signal Process. Workshop (SSP)*, pp. 257–260, Ann Arbor, MI, USA, Aug. 2012.

Doucet, A. and Johansen, A. M. A tutorial on particle filtering and smoothing: Fifteen years later. In *The Oxford Handbook of Nonlinear Filtering*, chapter 24, pp. 656–704. Oxford University Press, Oxford, UK, 2009.

Gneiting, T. and Raftery, A. E. Strictly proper scoring rules, prediction, and estimation. *J. Amer. Statist. Association*, 102(477):359–378, 2007.

Gordon, N., Salmond, D., and Smith, A. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proc. F Radar and Signal Process.*, 140:107–113, Apr. 1993.

Guo, S., Lin, Y., Feng, N., Song, C., and Wan, H. Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In *Proc. AAAI Conf. Artificial Intell.*, 2019.

Hamilton, J. D. *Time Series Analysis*. Princeton University Press, 1 edition, January 1994.

Huang, R., Huang, C., Liu, Y., Dai, G., and Kong, W. LSGCN: Long short-term traffic prediction with graph convolutional networks. In *Proc. Int. Joint Conf. Artificial Intell.*, pp. 2355–2361, 7 2020.

Li, Y., Yu, R., Shahabi, C., and Liu, Y. Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting. In *Proc. Int. Conf. Learning Rep.*, 2018.

Makridakis, S. and Hibon, M. Arma models and the box-jenkins methodology. *J. Forecasting*, 16(3):147–163, January 1997.

Oreshkin, B. N., Carpov, D., Chapados, N., and Bengio, Y. N-BEATS: Neural basis expansion analysis for interpretable time series forecasting. In *Proc. Int. Conf. Learning Rep.*, 2020.

Oreshkin, B. N., Amini, A., Coyle, L., and Coates, M. J. FC-GAGA: Fully Connected Gated Graph Architecture for spatio-temporal traffic forecasting. In *Proc. AAAI Conf. Artificial Intell.*, Jan. 2021.

Salinas, D., Flunkert, V., Gasthaus, J., and Januschowski, T. DeepAR: Probabilistic forecasting with autoregressive recurrent networks. *Int. J. Forecasting*, 36(3):1181 – 1191, 2020.

Sen, R., Yu, H.-F., and Dhillon, I. S. Think globally, act locally: A deep neural network approach to high-dimensional time series forecasting. In *Proc. Adv. Neural Info. Process. Systems*, volume 32, pp. 4837–4846, 2019.

Snyder, C., Bengtsson, T., Bickel, P., and Anderson, J. Obstacles to high-dimensional particle filtering. *Mon. Weather Rev.*, 136(12):4629–4640, Dec. 2008.

Sutskever, I., Vinyals, O., and Le, Q. V. Sequence to sequence learning with neural networks. In *Proc. Adv. Neural Info. Process. Systems*, volume 27, pp. 3104–3112, 2014.

Wang, Y., Smola, A., Maddix, D., Gasthaus, J., Foster, D., and Januschowski, T. Deep factors for forecasting. In *Proc. Int. Conf. Machine Learning*, Long Beach, California, USA, Jun 2019.

Wen, R., Torkkola, K., Narayanaswamy, B., and Madeka, D. A Multi-Horizon Quantile Recurrent Forecaster. *arXiv e-prints, arXiv:1711.11053*, 2017.

Wu, Z., Pan, S., Long, G., Jiang, J., and Zhang, C. Graph WaveNet for Deep Spatial-Temporal Graph Modeling. In *Proc. Int. Joint Conf. Artificial Intell.*, 2019.

Yu, B., Yin, H., and Zhu, Z. Spatio-Temporal Graph Convolutional Networks: A Deep Learning Framework for Traffic Forecasting. In *Proc. Int. Joint Conf. Artificial Intell.*, 2018.

Zheng, C., Fan, X., Wang, C., and Qi, J. GMAN: A Graph Multi-Attention Network for Traffic Prediction. In *Proc. AAAI Conf. Artificial Intell.*, 2020.