QUANTIFYING UNCERTAINTY IN DEEP SPATIOTEMPORAL FORECASTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Quantifying uncertainty is critical to risk assessment and decision making in high stakes domains. However, prior works for deep neural network uncertainty estimation have mostly focused on point prediction. A systematic study of uncertainty quantification methods for spatiotemporal forecasting has been missing in the community. In this paper, we analyze forecasting uncertainty in spatiotemporal sequences from both the Bayesian and Frequentist point of view via statistical decision theory. We further conduct case studies to provide an empirical comparison of Bayesian and Frequentist uncertainty estimation techniques. Through experiments on traffic and COVID-19 forecasts, we conclude that, with a limited amount of computation, Bayesian methods are typically more robust in mean prediction, while Frequentist methods are more effective in estimating the confidence levels.

1 INTRODUCTION

Uncertainty quantification (UQ), especially in high-stake domains such as public health and transportation, is critical to assist risk assessment for policy makers. Despite impressive prediction performances, deep neural networks are poor at providing uncertainty and tend to produce overconfident predictions (Amodei et al., 2016). Uncertainty quantification is receiving growing interests in deep learning (Gal & Ghahramani, 2016; Vandal et al., 2018; Qiu et al., 2019). Major attention has been devoted to combining the deep learning's success in accuracy with statistical calibration to generate both accurate and trustworthy predictions.

There are two types of UQ methods in deep learning. One leverages Frequentist thinking and focuses on the robustness. Perturbations are made to the inference procedure in initialization (Lakshminarayanan et al., 2017; Fort et al., 2019), neural network weights (Gal & Ghahramani, 2016), and datasets (Lee et al., 2015; Osband et al., 2016). The other type aims to model posterior beliefs of network parameters given the data (Neal, 2012; Kingma & Welling, 2013; Heek & Kalchbrenner, 2019). A common focal point of both approaches is the generalization ability of the neural networks. We not only wish to explain in-distribution examples, but also generalize to future data with model misspecifications and distribution shifts. Time series data provide the natural laboratory for studying this problem. Unfortunately, most existing UQ methods deal with point predictions instead of sequence predictions.

We focus on probabilistic spatiotemporal forecasting with deep learning. Recent advances in deep spatiotemporal forecasting have greatly improved point prediction accuracy, e.g. (Wang et al., 2017; Yu et al., 2018; Li et al., 2018). However, when developing UQ for spatiotemporal forecasts, many challenges arise. (1) Spatiotemporal dependency: accurate forecasts require models that can capture both spatial and temporal dependency. (2) Evaluation metrics: point estimates use RMSE as a metric. For uncertainty, common metrics such as held-out log-likelihood requires exact likelihood computation. (3) Sample size: deep learning suffers in small sample regime, incurring large uncertainty in predictions. (4) Forecasting horizon: the sequence dependency in time series often leads to error propagation for long-term forecasting, so does uncertainty.

In this paper, we conduct a systematic study of deep uncertainty quantification for spatiotemporal forecasting. We investigate the evaluation metrics, properties of both Frequentist and Bayesian UQ methods, as well as their practical performances. We provide a recipe for practitioners when facing UQ problems in deep spatiotemporal forecasting. We experimented with two real-world applications to validate our hypothesis: traffic forecasting and COVID-19 forecasting. Our study reveals that:

- Posterior sampling excels at mean prediction whereas quantile regression methods excel at confidence level estimation.
- Quantile regression outperforms its approximate counterpart; posterior sampling outperforms simple dropout, signifying the utility of fidelity to the probability scores and the posterior.
- Sample complexity of posterior sampling is lower than that of the bootstrap method, potentially due to posterior contraction (Wilson & Izmailov, 2020).

2 Evaluating Probabilistic Forecasts

We first define the metrics for evaluating probabilistic forecasts. While held-out likelihood is a common metric used in existing UQ literature, it is difficult for deep neural networks as they often do not generate explicit likelihood outputs (Kingma & Dhariwal, 2018). Instead, we revisit a key concept from statistics and econometrics called Mean Interval Score (Gneiting & Raftery, 2007).

2.1 Mean Interval Score (MIS)

Mean Interval Score (MIS) is a scoring function for interval forecasts. It rewards narrower confidence or credible intervals and penalizes intervals that do not include the observations. From a computational perspective, MIS is preferred over other scoring functions such as the Brier score (Brier, 1950), Continuous Ranked Probability Score (CRPS) (Matheson & Winkler, 1976; Hamill & Wilks, 1995; Gneiting & Raftery, 2007) as it is intuitive and easy to compute. From a statistical perspective, MIS does not constrain the model to be parametric, nor does it require explicit likelihood functions. It is therefore better suited for comparison across a wide range of methods. It is also known as Winkler loss (Askanazi et al., 2018).

We formally define MIS for estimated upper and lower confidence bounds. For a one dimensional random variable $Z \sim \mathbb{P}_Z$, if the estimated upper and lower confidence bounds are u and l, where u and l are the $(1 - \frac{\rho}{2})$ and $\frac{\rho}{2}$ quantiles for the $(1 - \rho)$ confidence interval, MIS is defined using samples $z_i \sim \mathbb{P}_Z$:

$$\mathrm{MIS}_N(u, l; \rho) = \frac{1}{N} \sum_{i=1}^{N} \left((u - l) + \frac{2}{\rho}(z_i - u)\mathbb{1}\left\{z_i > u\right\} + \frac{2}{\rho}(l - z_i)\mathbb{1}\left\{z_i < l\right\} \right).$$

In the large sample limit, MIS converges to the following expectation:

$$\mathrm{MIS}_\infty(u, l; \rho) = (u - l) + \frac{2}{\rho} \left(\mathbb{E}\left[Z - u | Z > u\right] + \mathbb{E}\left[l - Z | Z < l\right] \right).$$

We prove in the following two propositions that MIS is a consistent scoring function. The first proposition is concerned with the consistency of MIS in the large sample limit and the second studies the finite sample consistency of it. Proofs for both propositions are deferred to Appendix B.

Proposition 1. *Assume that the distribution $\mathbb{P}_Z$ of Z has a probability density function. Then for $(u^*, l^*) = \arg\min_{u>l; u,l \in \mathbb{R}} \mathrm{MIS}_\infty(u, l; \rho)$, $[l^*, u^*]$ is the $(1 - \rho)$ confidence level.*

Proposition 2. *For $(u^*, l^*) = \arg\min_{u>l; u,l \in \mathbb{R}} \mathrm{MIS}_N(u, l; \rho)$, $[l^*, u^*]$ is the $(1 - \rho)$ quantile of the empirical distribution formed by the samples $\{z_1, \cdots, z_N\}$.*

Note from Proposition 2 that the optimal interval for MIS_N contains ρ portion of the data points, because of the balance it strikes between the reward towards narrower intervals and the penalty towards excluing observations.

3 Uncertainty Quantification in Spatiotemporal Forecasts

We first cast uncertainty quantification from Frequentist and Bayesian perspectives into a single framework. Next, we introduce deterministic deep learning models for spatiotemporal forecasting. Finally, we describe both the Bayesian and Frequentist techniques to transform the deterministic forecasts into probabilistic forecasts.

3.1 A UNIFIED FRAMEWORK

We start from a statistical decision theory point of view to examine uncertainty quantification. Assume that each datum $\mathcal{X}$ and its label $\mathcal{Y}$ are generated from a mechanism specified by a probability distribution with parameter θ: $p(\mathcal{Y}|\mathcal{X};\theta)$. We further assume that the parameter θ is distributed according to $p(\theta)$. The goal of a probabilistic inference procedure is to minimize the statistical risk, which is an expectation of the loss function over the distribution of the data as well as $p(\theta)$, the distribution of the class of the models considered. Concretely, consider an estimator $\hat{\theta}$ of θ, which is a measurable function of the dataset $\{\mathcal{X}\} = \{\mathcal{X}_0, \cdots, \mathcal{X}_N\}$. We wish to minimize its risk:

$$\text{Risk}(\hat{\theta}, \theta) = \mathbb{E}\left[\|f(\hat{\theta}) - f(\theta)\|^2\right] = \mathbb{E}_\theta\left[\mathbb{E}\left[\|f(\hat{\theta}) - f(\theta)\|^2 \middle| \theta\right]\right] \tag{1}$$

$$= \mathbb{E}\left[\mathbb{E}_\theta\left[\|f(\hat{\theta}) - f(\theta)\|^2 \middle| \{\mathcal{X}\}\right]\right], \tag{2}$$

where in this paper function f is taken to be the output of the model, i.e., the prediction of the model given its parameters.

When we design an inferential procedure, we can use equation 1 or equation 2. If we take the procedure of equation 1 and be a frequentist, we first seek the estimator $\hat{\theta}$ to minimize $\mathbb{E}[\|f(\hat{\theta}) - f(\theta)\|^2|\theta]$. In practice we minimize the empirical risk over the training dataset $\{\mathcal{X}\}$, $\frac{1}{N}\sum_{i\in\{0,\cdots,N\}}(f(\mathcal{X}_i;\hat{\theta}) - \mathcal{Y}_i)^2$, instead of the original expectation that is agnostic to the algorithms.

If we take the procedure of equation 2 and be Bayesian, we first take the expectation over the distribution of the parameter space θ, conditioning on the training dataset $\{\mathcal{X}\}$. We can find that minimizing $\mathbb{E}_\theta\left[\|f(\hat{\theta}) - f(\theta)\|^2 \middle| \{\mathcal{X}\}\right]$ is equivalent to taking $f(\hat{\theta})$ to be $\mathbb{E}_\theta\left[f(\theta)\right]$. For a proper prior distribution over θ, this boils down to the posterior mean of $f(\theta)$. In this work, we also take function f to be the output of the model.

To quantify uncertainty of the respective approaches, it is important to capture variations in θ that determines the distribution of the data. There are two sources of in-distribution uncertainties corresponding to different modes of variations in θ: effective dimension of the estimator $\hat{\theta}$ being smaller than that of the true θ, oftentimes addressed to as the bias correction problem; and the variance of the estimator $\hat{\theta}$, often approached via parameter calibration. On top of that, we are also faced with distribution shift that poses additional challenges.

Frequentist methods aim to directly capture variations in the data. For example, quantile regression explores the space of predictions within the model class by ensuring proper coverage of the data set. Boostrap method resamples the data set to target both the bias correction and the parameter calibration problems, albeit possessing a potentially high resampling variance. Bayesian methods often leverage prior knowledge about the distribution of θ and use MCMC sampling to integrate over different models to capture both variations in the parameter space.

In practice, it is important to understand the sample complexity (e.g., number of resampling runs in bootstrap or number of MCMC samples) of all the methods and how it scales with the model and data complexity. For deep learning models, it is also crucial to capture the trade off between variance incurred by the algorithms and their computation complexity. To understand the performance of the uncertainty quantification methods, we apply the statistically consistent metric MIS to measure how well the confidence intervals computed from either frequentist or Bayesian methods capture uncertainty of the predictions given the data. In what follows, we introduce the computation methods for uncertainty quantification.

3.2 SPATIOTEMPORAL FORECASTING MODEL

Given multivariate time series $\mathcal{X} = (\mathbf{X}_1, \cdots, \mathbf{X}_t)$ of t time step, with each $\mathbf{X}_t \in \mathbb{R}^{P\times D}$ indicating D features from P locations. We also have spatial information represented as an adjacency matrix A. Spatiotemporal forecasting model learns a function f:

$$f : (\mathcal{X}; A) \to (\mathbf{X}_{t+1}, \cdots, \mathbf{X}_{t+H}; A) \tag{3}$$

where H is the forecasting horizon. When the locations are evenly spaced, popular deep spatiotemporal forecasting models include Convolutional LSTM (ConvLSTM) (Xingjian et al., 2015) and

PredRNN (Wang et al., 2017). The convolution operator extract spatial features and RNN/LSTM models temporal dynamics. When the locations are distributed as a graph, a natural extension is to generalize regular convolution to graph convolution. This type of models include Spatiotemporal Graph CNN (Yu et al., 2018), Diffusion Convolutional RNN (DCRNN) (Li et al., 2018) and etc.

The key idea behind these deterministic models is to replace the matrix multiplication operator in RNN/LSTM with convolution or graph convolution. Simply put, for every feature $\mathbf{x}_t = \mathbf{X}_{:,d}$:

$$\mathbf{h}_{t+1} = \sigma(W \cdot \mathbf{h}_t + W \cdot \mathbf{x}_t) \longrightarrow \mathbf{h}_{t+1} = \sigma(W * \mathbf{h}_t + W * \mathbf{x}_t) \tag{4}$$

where $\mathbf{h}$ are the hidden states and $*$ denotes convolution for regular grids or graphs:

$$\text{Regular Grids} : (W * \mathbf{x})_i = \sum_k W_k \mathbf{x}_{i-k}, \quad \text{Graphs} : W * \mathbf{x} = W \cdot (D^{-1}A) \cdot \mathbf{x} \tag{5}$$

Here $D_{P\times P}$ contains the diagonal element of A. To enable these deterministic models to generate probabilistic forecast, we describe different UQ methods below.

3.3 Frequentist UQ Methods

Bootstrap. The (generalized) bootstrap method (Efron & Hastie, 2016) randomly generates in each round a weight vector over the index set of the data. The data are then resampled according to the weight vector. With every resampled dataset, we retrain our model and make predictions. Using the predictions from different retrained models, we estimate the confidence intervals of our predictions.

MIS and Quantile Regression For a fixed confidence level ρ, we can directly minimize MIS to obtain estimates of the confidence intervals. Specifically, to use MIS as a loss function for deep neural networks, we use a multi-headed model to jointly output the upper bound $u(x)$, lower bound $l(x)$, and the prediction $f(x)$ for a given input x, and minimize the neural network parameter θ:

$$\begin{aligned} L_{\text{MIS}}(y, u(x), l(x), f(x); \theta, \rho) = \min_\theta \Big\{ &\mathbb{E}_{(x,y)\sim\mathcal{D}}[(u(x) - l(x)) + \frac{2}{\rho}(y - u(x))\mathbb{1}\{y > u(x)\} \\ &+ \frac{2}{\rho}(l(x) - y)\mathbb{1}\{y < l(x)\} + |y - f(x)|] \Big\} \end{aligned} \tag{6}$$

Here $\mathbb{1}\{\}$ is an indicator function, which can be implemented using the identity operator over the larger element in Pytorch.

For quantile regression (Koenker & Bassett Jr, 1978; Koenker, 2005), we can use the one-sided quantile loss function to generate predictions for a fixed confidence level ρ. Given an input x, and the output $f(x)$ of a neural network, parameterized by θ, quantile loss is defined as follows:

$$L_{\text{Quantile}}(y, f(x); \theta, \rho) = \min_\theta \Big\{ \mathbb{E}_{(x,y)\sim\mathcal{D}}[(y - f(x))(\rho - \mathbb{1}\{y < f(x)\}] \Big\} \tag{7}$$

Quantile regression behaves similarly as the MIS regression method. Both methods generate one confidence interval per time. In addition, Kivaranovic et al. (2020); Tagasovska & Lopez-Paz (2019); Pearce et al. (2018) have explored similar ideas of directly optimizing the prediction interval using different variations of quantile loss.

One caveat of these methods is that different predicted quantiles can cross each other due to variations given finite data. This will cause a strange phenomenon when the size of the data set and the model capacity is limited: the higher confidence interval does not contain the interval of lower confidence level or even the point estimate. One remedy for this issue is to add variations in both the data and the parameters to increase the effective data size and model capacity. In particular, during training, we can use different subsets of data and repeat random initialization from a prior distribution to form an ensemble of models. In this way, our modified MIS and quantile regression methods have integrated across different model to quantify the prediction uncertainty and have taken advantage of the Bayesian philosophies.

Another solution to alleviate quantile crossing and unify different confidence levels is to minimize CRPS by assuming the quantile function to be a piecewise linear spline with monotonicity (Gasthaus et al., 2019), a method we call Spline Quantile regression (SQ). In the experiment, we also included this method for comparison.

3.4 BAYESIAN UQ METHODS

Stochastic Gradient MCMC (SG-MCMC) To estimate expectations or quantiles according to the posterior distribution over the parameter space, we use SG-MCMC (Welling & Teh, 2011; Ma et al., 2015). We find in the experiments that the stochastic gradient thermostat method (SGNHT) (Ding et al., 2014; Shang et al., 2015) is particularly useful in controlling the stochastic gradient noise. This is consistent with the observation in (Heek & Kalchbrenner, 2019) where stochastic gradient thermostat method is applied to an i.i.d. image classification task.

To generate samples of model parameters θ (with a slight abuse of notation) according to SGNHT, we first denote the loss function (or the negative log-likelihood) over a minibatch of data as $\widetilde{\mathcal{L}}(\theta)$. We then introduce hyper-parameters including the diffusion coefficients A and the learning rate h and make use of auxiliary variables $p \in \mathbb{R}^d$ and $\xi \in \mathbb{R}$ in the algorithm. We randomly initialize θ, p, and ξ and update according to the following update rule.

$$\begin{cases} \theta_{k+1} = \theta_k + p_k h \\ p_{k+1} = p_k - \nabla\widetilde{\mathcal{L}}(\theta)h - \xi_k p_k h + \mathcal{N}(0, 2Ah) \\ \xi_{k+1} = \xi_k + \left(\frac{p_k^{\mathrm{T}} p_k}{d} - 1\right) h. \end{cases} \tag{8}$$

Upon convergence of the above algorithm at K-th step, θ_K follows the distribution of the posterior. We run parallel chains to generate different samples according to the posterior and quantify the predictions uncertainty.

Approximate Bayesian Inference SG-MCMC can be computationally expensive. There are also approximate Bayesian inference methods introduced to accelerate the inference procedures (Maddox et al., 2019; Dusenberry et al., 2020). In particular, the Monte Carlo (MC) drop out method sets some of the network weights to zero according to a prior distribution (Gal & Ghahramani, 2016; Gal et al., 2017). This method serves as a simple alternative to variational Bayes methods which approximate the posterior (Blundell et al., 2015; Graves, 2011; Louizos & Welling, 2016; Rezende et al., 2014; Qiu et al., 2019). We examine the popular MC dropout method in the experiments for comparison.

3.5 A RECIPE FOR UQ IN SPATIOTEMPORAL FORECASTING

The above-mentioned UQ methods have different properties. Table 3 shows an overview comparison of different uncertainty quantification methods for deep spatiotemporal forecasting. Through our experiments, we provide the following recipe for practitioners.

- **large data, sufficient computation**: We recommend SG-MCMC and bootstrap as both methods generate accurate predictions, high-quality uncertainty quantification, which means the corresponding MIS is small, and have asymptotic consistency.
- **large data, limited computation**: We recommend MIS and quantile regression. Both of them prevail in providing accurate results with high-quality uncertainty quantification.
- **small data**: Bayesian learning with SG-MCMC can have an advantage here. By choosing a proper prior, SG-MCMC can lead to better generalization. Frequentist methods are often inferior with very limited samples, especially for mean prediction, see experiments for more details.
- **asymptotic consistency**: Both bootstrap and SG-MCMC methods are asymptotically consistent, making them the default choice when the computational budget and the dataset are sufficient. When the computational budget is restrictive, the comparison is about sample complexity: the number of bootstrap resampling versus the number of MCMC chains determines which method needs more parallel computing resources. We found in our experiments that the sample complexity of bootstrap is consistently higher than that of posterior sampling.

4 RELATED WORK

Time Series Forecasting Classic time series models such as ARMA or ARIMA were developed for univariate time series (see Hyndman & Athanasopoulos, 2018, and the references therein). For multivariate time series, (Yu et al., 2016) exploit information sharing across variables by applying matrix factorization. (Salinas et al., 2020; Wang et al., 2019b) introduce latent variables and use

RNNs to approximate the parameters in the likelihood functions. These probabilistic forecasting methods provide uncertainty directly, but assumes factorizability over time or exchangeability over space on the model likelihood function, neither holds true in our current setting. (Gasthaus et al., 2019) proposed spline quantile function.

In contrast to time series, spatiotemporal forecasting poses additional challenges due to the higher-order dependencies in space, time and variables. (Yu et al., 2014) introduces tensor methods to capture the higher-order dependency. Deep learning models such as Convolutional LSTM (Xingjian et al., 2015), PredNet (Lotter et al., 2016), PredRNN (Wang et al., 2017) are deterministic models. Stochastic videos predictions models such as (Kosiorek et al., 2018) are based on the VAE framework but do not provide explicit uncertainty. Another unique challenge that is different from video prediction is the non-Euclidean geometry. Recently, graph convolutional LSTM (Yu et al., 2018; Li et al., 2018) were proposed to capture the non-Euclidean spatial dependency, but without uncertainty.

Uncertainty Quantification Bayesian neural networks learning (BNN) uses approximate Bayesian inference to improve inference efficiency, see detailed discussion in Section 3.4. Wang et al. (2016) propose natural parameter network using exponential family distributions. Shekhovtsov & Flach (2018) propose new approximations for categorical transformations. However, these BNNs are focused on point estimate rather than time series forecasting. For sequence predictions, DeepAR (Salinas et al., 2020) estimates the forecasting uncertainty by parameterizing the likelihood function with NNs. However, their method relies on structural (factorizable) assumptions on the likelihood function. (Wang et al., 2019a) propose a Bayesian deep learning method for UQ in weather forecasts based on VI but is only a heuristic. Most recently (Alaa & van der Schaar, 2020) propose a frequentist approach for UQ in a multivariate time series forecasting task. They approximate bootstrapping using the influence function to estimate the uncertainty in RNNs. This paper extends previous works and studies the efficacy and efficiency of various methods for UQ in spatio-temporal forecasts.

5 Experiments

We evaluate the performance of both Frequentist and Bayesian UQ methods on the time-series dataset for Traffic forecasting and COVID-19 incident deaths (Sec. 5.1 and Sec. 5.2). The experiments are implemented using pytorch (Paszke et al., 2019). Based on our observations, Bayesian methods typically reach lower error in their mean predictions while Frequentist methods, especially quantile and MIS regression, prevail in estimating the confidence levels.

5.1 Traffic Forecasting

5.1.1 Datasets description and model setup

Datasets. We use the *METR-LA* (Jagadish et al., 2014) dataset which contains information from loop detector sensors in Los Angeles County highway system. The task is to forecast traffic speed for 207 sensors simultaneously. We follow the exact setting and dataset in (Li et al., 2017) and use calibrated network distance to construct the graph. We use 70-10-20 split of data for training, validation and testing. Missing values are excluded. For all training, validation, and testing datasets, we are using an hour's traffic data to predict the next 5, 10, 15, ..., 60 minutes of traffic.

Models and Evaluation Metrics. We implement DCRNN as the base deep learning forecasting model for Frequentist and Bayesian methods. We applied early stopping, curriculum learning, and gradient clipping (Zhang et al., 2019) to improve generalization.

We apply 6 UQ methods to DCRNN: Bootstrap (with 25 resampled datasets), quantile regression, Spline quantile regression (SQ), and MIS regression are Frequentist methods. Monte Carlo Dropout, SG-MCMC (with 25 posterior samples) are Bayesian methods. For MIS regression, we combine the Mean Absolute Error (MAE) score together with the Mean Interval Score (MIS) as the loss function. For SG-MCMC, we apply Stochastic Nosé-Hoover thermostat (Ding et al., 2014) for sampling posterior. See Appendix C.1 for additional details. We report Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) for mean prediction and Mean Interval Score (MIS) to assess the quality of the prediction uncertainty.

5.1.2 Performance Comparison

Table 1 compares UQ methods for 15 minutes, 30 minutes and 1 hour ahead traffic speed forecasting on *METR-LA*. DCRNN is the original deterministic model which we use as a reference. We observe

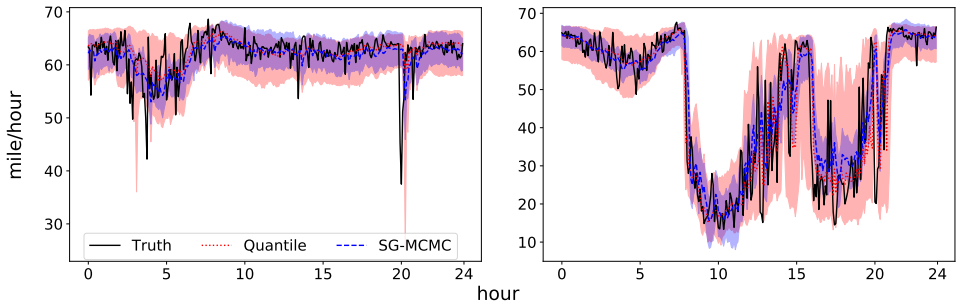

Figure 1: 15 min ahead traffic forecasting over 24 hour time span on two sensors. Left: regular hour traffic with a sudden drop at around 20th hour due to accidents. Right: rush hour traffic forecasting.

that Bayesian methods (SG-MCMC) lead to better prediction accuracy, outperforming the deterministic DCRNN model. Meanwhile, Frequentist methods generally outperform Bayesian methods in 80%, 90%, 95% confidence interval. Directly minimizing MIS score function achieves the best performance in uncertainty. Its overall performance is similar to quantile regression. MC dropout has a relatively good prediction accuracy with a reasonable MIS score, specially for the short-term forecast. The uncertainty estimation is robust within a reasonable range of dropout rate (Zhu & Laptev, 2017). We use 5% dropout rate in this experiment. We note that Bootstrap performs poorly for both metrics, most likely due to limited samples. SQ method is implemented to deal with the quantile crossing problem. We use 5 knots to build the spline quantile functions, which might not be enough for prediction. Both prediction accuracy and MIS are not good for this method. We also separate MIS into the interval and the coverage, where the interval is the length between upper and lower bounds and the coverage is the penalty for confidence bounds not containing the truth. Table 1 shows Bootstrap and SQ methods tend to be overconfident and provide shorter intervals with larger coverage compared with Quantile and MIS methods.

We visualize the predictions from quantile and SG-MCMC to compare their behaviors. As shown in Figure 1, SG-MCMC generates mean predictions closer to the ground truth but with narrower credible intervals. This problem of overconfident credible intervals can be resolved by obtaining more posterior samples. However, it requires sizable parallel computing resource to obtain more posterior samples for large datasets like *METR-LA*. We also see an adversarial effect of our prior here: SG-MCMC is reacting slightly slower to sudden changes in dynamics than quantile regression, due to the agnostic Gaussian prior.

5.2 COVID-19 Forecasting Performance Comparison

We further investigate COVID-19 forecasting. This is highly challenging as the data is very small, highly noisy, and pertains complex spatial dependency. We chose this dataset as its unique challenges make it an ideal task to test and compare different UQ methods for spatiotemporal forecasting.

5.2.1 Datasets description and model setup

Datasets. The COVID-19 dataset contains reported death from Johns Hopkins University (Dong et al., 2020) and the death predictions from a mechanistic, stochastic, and spatial metapopulation epidemic model called Global Epidemic and Mobility Model (GLEAM) (Balcan et al., 2009; 2010; Tizzoni et al., 2012; Zhang et al., 2017b; Chinazzi et al., 2020). Both data are recorded for the 50 US states during the time period from May 24th to Sep 12th 2020. We use the residual between the reported death and the corresponding GLEAM predictions to train the model (`http://covid19.gleamproject.org/`). See Appendix D.1 for details about GLEAM.

We construct the spatial graph—adjacency matrix A in equation 3—using air traffic between different states. The air traffic data is obtained from the Official Aviation Guide (OAG) and the International Air Transportation Association (IATA) databases (updated in 2020). Each directed weighted edge of the graph represents the average number of passenger traveling between two states on a daily basis.

DeepGLEAM. Directly predicting the death number using DCRNN (Deep) suffers from low accuracy as we have limited samples, see Appendix D.3 for quantitative comparisons. Instead, we use the difference between daily death number and GLEAM predictions as input and output of the DCRNN model and name it DeepGLEAM model. We subtract the learned difference from the GLEAM outputs to improve its prediction. Essentially, we use DCRNN (deep neural networks) to learn the correction terms for the mechanistic model GLEAM. See Appendix D.2 for details about

T	Metric	DCRNN	Bootstrap	Quantile	SQ	MIS	MC Dropout	SG-MCMC
15 min	RMSE	4.91	5.35	5.00	5.20	5.02	4.93	**4.80**
	MAE	2.38	2.63	2.43	2.67	2.63	2.47	**2.32**
	MIS (95% CI)	—	39.76	18.32	29.04	**18.26**	27.61	32.21
	Interval	—	5.80	12.42	8.38	12.46	8.99	8.73
	MIS (90% CI)	—	25.28	14.91	19.05	**14.87**	19.14	23.17
	Interval	—	4.69	9.85	7.89	9.89	7.54	6.09
	MIS (80% CI)	—	16.38	11.79	13.53	**11.68**	13.56	15.95
	Interval	—	3.56	6.89	6.92	7.09	5.88	4.33
30 min	RMSE	5.94	6.58	6.00	6.36	5.95	6.08	**5.59**
	MAE	2.73	3.19	2.79	3.10	3.08	2.94	**2.54**
	MIS (95% CI)	—	38.48	21.54	40.93	**21.09**	33.38	31.87
	Interval	—	7.86	13.48	8.37	13.99	11.10	12.62
	MIS (90% CI)	—	25.60	17.62	25.17	**17.16**	23.03	24.18
	Interval	—	6.36	10.51	7.87	10.95	9.32	9.08
	MIS (80% CI)	—	17.32	13.94	16.78	**13.59**	16.25	17.41
	Interval	—	4.85	7.23	6.91	7.56	7.26	6.47
1 hour	RMSE	7.07	7.98	7.05	7.91	6.98	7.76	**6.41**
	MAE	3.14	3.99	3.19	3.75	3.65	3.70	**3.00**
	MIS (95% CI)	—	38.58	25.74	60.56	**24.33**	43.11	30.35
	Interval	—	11.65	14.50	8.38	15.55	14.46	18.79
	MIS (90% CI)	—	27.29	21.18	35.23	**19.95**	29.68	24.59
	Interval	—	9.45	11.06	7.88	11.99	12.14	14.45
	MIS (80% CI)	—	19.58	16.71	22.06	**16.08**	20.88	19.20
	Interval	—	7.28	7.45	6.91	7.88	9.46	10.42

Table 1: Performance comparison of Frequentist (Bootstrap, Quantile, SQ, MIS) and Bayesian (MC Dropout, SG-MCMC) UQ methods applied to DCRNN for *METR-LA* forecasting.

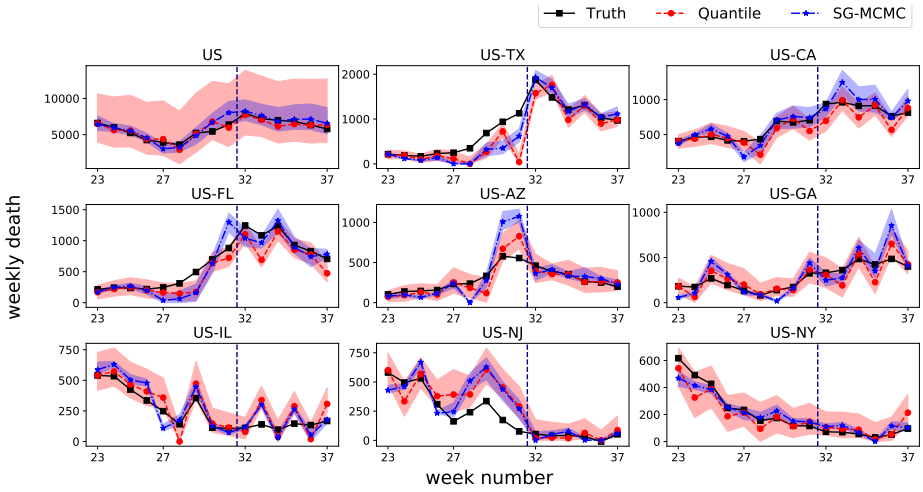

Figure 2: One week ahead COVID-19 prediction in the country level and 8 states with largest death from week 23 to week 37.

the model setup. In the following, we report the average out-of-sample prediction performance of the model.

5.2.2 Performance comparison

Table 2 compares 6 different UQ methods for one week, two weeks, three weeks, and four weeks ahead mortality prediction on COVID-19. We apply DCRNN as the deterministic model. Similar to observations in the experiment on traffic data, we observe that SG-MCMC outperforms other models in RMSE and MAE. With limited amount of samples, introducing prior further helps the model to generalize better. Quantile and MIS regressions achieve the best performance in MIS. We note that Bootstrap still underperforms due to the limited amount of samples we have. Dropping samples from insufficient samples negatively affects the performance of Bootstrap. SQ regression still has large MIS due to the small number of knots for linear spline quantile function construction. For MC dropout, its prediction accuracy is relatively robust but its MIS is bad. The interval in Table 2 shows

T	Metric	DCRNN	Bootstrap	Quantile	SQ	MIS	MC Dropout	SG-MCMC
1W	RMSE	66.03	64.63	70.42	71.72	74.07	66.82	**58.30**
	MAE	34.32	32.57	36.63	34.42	39.03	34.27	**29.72**
	MIS (95% CI)	—	856.87	**413.37**	1049.69	427.53	790.24	563.77
	Interval	—	32.48	190.98	23.73	227.19	47.79	47.05
	MIS (90% CI)	—	444.68	302.77	541.61	**285.99**	443.13	365.6
	Interval	—	32.48	129.92	22.62	141.24	40.51	43.04
	MIS (80% CI)	—	252.39	218.73	286.35	**200.32**	254.42	212.97
	Interval	—	24.28	73.23	20.31	84.45	31.94	35.96
2W	RMSE	57.67	54.21	61.35	63.32	64.42	57.63	**46.61**
	MAE	33.72	32.38	36.81	34.03	36.35	33.64	**27.65**
	MIS (95% CI)	—	762.55	**363.14**	1010.97	379.27	686.43	599.14
	Interval	—	36.25	219.06	24.46	260.24	55.93	45.45
	MIS (90% CI)	—	399.40	276.32	522.99	**270.41**	397.52	332.00
	Interval	—	36.25	150.54	23.35	161.9	47.53	42.19
	MIS (80% CI)	—	235.51	196.56	277.86	**185.53**	236.7	197.59
	Interval	—	27.69	85.69	21.00	97.11	37.58	35.75
3W	RMSE	70.12	67.70	72.92	72.52	73.65	70.03	**59.27**
	MAE	41.37	40.33	44.29	41.24	43.15	41.26	**34.62**
	MIS (95% CI)	—	1028.63	411.05	1292.98	**402.46**	905.66	821.71
	Interval	—	39.95	242.47	24.16	291.96	62.39	46.16
	MIS (90% CI)	—	534.29	315.96	664.50	**304.12**	515.15	443.94
	Interval	—	39.95	170.12	23.09	184.03	53.10	43.07
	MIS (80% CI)	—	307.14	237.90	349.00	**220.60**	300.09	254.54
	Interval	—	29.87	96.07	20.81	111.79	42.03	36.63
4W	RMSE	70.75	68.63	73.92	**69.94**	72.44	70.60	70.57
	MAE	42.37	41.71	46.20	41.79	44.45	42.28	**40.66**
	MIS (95% CI)	—	1035.26	455.27	1303.02	**428.82**	891.45	852.26
	Interval	—	43.61	262.09	23.85	316.13	67.52	47.58
	MIS (90% CI)	—	539.43	359.69	669.76	**343.83**	512.72	458.94
	Interval	—	43.61	190.23	22.79	206.27	57.50	44.65
	MIS (80% CI)	—	315.05	262.76	351.96	**252.51**	302.66	261.32
	Interval	—	32.32	105.6	20.58	128.51	45.56	38.03

Table 2: Performance comparison of different approaches on Autoregressive DeepGLEAM Model for COVID-19 mortality forecasting.

Bootstrap, SQ, MC dropout, and SG-MCMC methods tend to make overconfident predictions with shorter intervals and larger coverage compared with Quantile, and MIS methods.

We visualize the predictions between quantile regression and SG-MCMC in Figure 2. We feed data before week 32 into the model for training or validation, and start making predictions after the vertical dashed line. We find mean predictions from SG-MCMC are closer to the ground truth while quantile regression provides better confidence bounds at the state level. For example, the US-TX subplot shows that the SG-MCMC's mean prediction is closer to the ground truth compared with quantile prediction. Meanwhile, the US-GA and US-NY subplots show that the overconfident credible interval of SG-MCMC fails to cover the ground truth. In these cases, quantile regression can provide safer confidence bounds and achieve better MIS. For country-level prediction, as shown in the US subplot, SG-MCMC outperforms quantile regression in both mean prediction accuracy and confidence bounds.

6 Conclusion

We conduct case studies on forecasting uncertainty in spatiotemporal sequences from both Bayesian and Frequentist point of view. Through experiments on both traffic and COVID-19 datasets, we conclude that, with a limited amount of computation, Bayesian methods are typically more robust in mean predictions, while Frequentist methods are more effective in estimating the confidence levels. It is of interest to understand how to best combine Bayesian credible intervals with frequentist confidence intervals to excel in both mean predictions and confidence bounds. Another future direction worth exploring is how to explicitly make use of the spatiotemporal structure of the data in the inference procedures. Current methods we analyze treats minibatches of data as i.i.d. samples. For longer time series with a graph structure, it is interesting how to leverage the spatiotemporal nature of the data to construe efficient inference algorithms (Ma et al., 2017; Aicher et al., 2017; 2019; Alaa & van der Schaar, 2020).

REFERENCES

Socioeconomic Data and Applications Center (SEDAC), Columbia University http://sedac.ciesin.columbia.edu/gpw.

Chris Aicher, Yi-An Ma, Nick Foti, and Emily B. Fox. Stochastic gradient MCMC for state space models. *SIAM Journal on Mathematics of Data Science (SIMODS)*, 1:555–587, 2017.

Christopher Aicher, Srshti Putcha, Christopher Nemeth, Paul Fearnhead, and Emily B. Fox. Stochastic gradient MCMC for nonlinear state space models. arXiv:1901.10568, 2019.

AM Alaa and M van der Schaar. Frequentist uncertainty in recurrent neural networks via blockwise influence functions. In *International Conference on Machine Learning*, 2020.

Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete problems in ai safety. *arXiv preprint arXiv:1606.06565*, 2016.

Ross Askanazi, Francis X Diebold, Frank Schorfheide, and Minchul Shin. On the comparison of interval forecasts. *Journal of Time Series Analysis*, 39(6):953–965, 2018.

Duygu Balcan, Vittoria Colizza, Bruno Gonçalves, Hao Hu, José J Ramasco, and Alessandro Vespignani. Multiscale mobility networks and the spatial spreading of infectious diseases. *Proceedings of the National Academy of Sciences*, 106(51):21484–21489, 2009.

Duygu Balcan, Bruno Gonçalves, Hao Hu, José J Ramasco, Vittoria Colizza, and Alessandro Vespignani. Modeling the spatial spread of infectious diseases: The global epidemic and mobility computational model. *Journal of computational science*, 1(3):132–145, 2010.

Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural network. In *International Conference on Machine Learning*, pp. 1613–1622, 2015.

Glenn W Brier. Verification of forecasts expressed in terms of probability. *Monthly weather review*, 78(1):1–3, 1950.

Matteo Chinazzi, Jessica T. Davis, Marco Ajelli, Corrado Gioannini, Maria Litvinova, Stefano Merler, Ana Pastore y Piontti, Kunpeng Mu, Luca Rossi, Kaiyuan Sun, Cécile Viboud, Xinyue Xiong, Hongjie Yu, M. Elizabeth Halloran, Ira M. Longini, and Alessandro Vespignani. The effect of travel restrictions on the spread of the 2019 novel coronavirus (covid-19) outbreak. *Science*, 368 (6489):395–400, 2020.

Jessica T Davis, Matteo Chinazzi, Nicola Perra, Kunpeng Mu, Ana Pastore y Piontti, Marco Ajelli, Natalie E Dean, Corrado Gioannini, Maria Litvinova, Stefano Merler, Luca Rossi, Kaiyuan Sun, Xinyue Xiong, M. Elizabeth Halloran, Ira M Longini, Cécile Viboud, and Alessandro Vespignani. Estimating the establishment of local transmission and the cryptic phase of the covid-19 pandemic in the usa. *medRxiv*, 2020.

Nan Ding, Youhan Fang, Ryan Babbush, Changyou Chen, Robert D Skeel, and Hartmut Neven. Bayesian sampling using stochastic gradient thermostats. In *Advances in neural information processing systems*, pp. 3203–3211, 2014.

Ensheng Dong, Hongru Du, and Lauren Gardner. An interactive web-based dashboard to track covid-19 in real time. *The Lancet infectious diseases*, 20(5):533–534, 2020.

M. Dusenberry, G. Jerfel, Y. Wen, Y.-A. Ma, J. Snoek, K. Heller, B. Lakshminarayanan, and D. Tran. Efficient and scalable Bayesian neural nets with rank-1 factors. In *Proceedings of the 37th International Conference on Machine Learning*, pp. 9823–9833. 2020.

Bradley Efron and Trevor Hastie. *Computer Age Statistical Inference*, volume 5. Cambridge University Press, 2016.

S. Fort, H. Hu, and B. Lakshminarayanan. Deep ensembles: A loss landscape perspective. arXiv:1912.02757, 2019.

Yarin Gal and Zoubin Ghahramani. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pp. 1050–1059, 2016.

Yarin Gal, Jiri Hron, and Alex Kendall. Concrete dropout. In *Advances in Neural Information Processing Systems*, pp. 3581–3590, 2017.

Jan Gasthaus, Konstantinos Benidis, Yuyang Wang, Syama Sundar Rangapuram, David Salinas, Valentin Flunkert, and Tim Januschowski. Probabilistic forecasting with spline quantile function RNNs. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 1901–1910, 2019.

Tilmann Gneiting and Adrian E Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American statistical Association*, 102(477):359–378, 2007.

Alex Graves. Practical variational inference for neural networks. In *Advances in neural information processing systems*, pp. 2348–2356, 2011.

Thomas M Hamill and Daniel S Wilks. A probabilistic forecast contest and the difficulty in assessing short-range forecast uncertainty. *Weather and Forecasting*, 10(3):620–631, 1995.

Jonathan Heek and Nal Kalchbrenner. Bayesian inference for large scale image classification. arXiv:1908.03491, 2019.

Rob J Hyndman and George Athanasopoulos. *Forecasting: principles and practice*. OTexts, 2018.

Hosagrahar V Jagadish, Johannes Gehrke, Alexandros Labrinidis, Yannis Papakonstantinou, Jignesh M Patel, Raghu Ramakrishnan, and Cyrus Shahabi. Big data and its technical challenges. *Communications of the ACM*, 57(7):86–94, 2014.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Diederik P Kingma and Max Welling. Auto-encoding variational Bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In *Advances in neural information processing systems*, pp. 10215–10224, 2018.

Danijel Kivaranovic, Kory D Johnson, and Hannes Leeb. Adaptive, distribution-free prediction intervals for deep networks. In *International Conference on Artificial Intelligence and Statistics*, pp. 4346–4356. PMLR, 2020.

R. Koenker. *Quantile Regression*. Econometric Society Monographs. Cambridge University Press, 2005.

R. Koenker and G. Bassett Jr. Regression quantiles. *Econometrica: Journal of the Econometric Society*, pp. 33–50, 1978.

Adam Kosiorek, Hyunjik Kim, Yee Whye Teh, and Ingmar Posner. Sequential attend, infer, repeat: Generative modelling of moving objects. In *Advances in Neural Information Processing Systems*, pp. 8606–8616, 2018.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*, pp. 6402–6413, 2017.

Stefan Lee, Senthil Purushwalkam, Michael Cogswell, David Crandall, and Dhruv Batra. Why m heads are better than one: Training a diverse ensemble of deep networks. *arXiv:1511.06314*, 2015.

Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. *arXiv preprint arXiv:1707.01926*, 2017.

Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. In *International Conference on Learning Representations (ICLR)*, 2018.

William Lotter, Gabriel Kreiman, and David Cox. Deep predictive coding networks for video prediction and unsupervised learning. *arXiv preprint arXiv:1605.08104*, 2016.

Christos Louizos and Max Welling. Structured and efficient variational deep learning with matrix gaussian posteriors. In *International Conference on Machine Learning*, pp. 1708–1716, 2016.

Yi-An Ma, Tianqi Chen, and Emily Fox. A complete recipe for stochastic gradient mcmc. In *Advances in Neural Information Processing Systems*, pp. 2917–2925, 2015.

Yi-An Ma, Nicholas J. Foti, and Emily B. Fox. Stochastic gradient MCMC methods for hidden Markov models. In *Proceedings of International Conference on Machine Learning 34 (ICML 2017)*, pp. 2265–2274, 2017.

Wesley J Maddox, Pavel Izmailov, Timur Garipov, Dmitry P Vetrov, and Andrew Gordon Wilson. A simple baseline for Bayesian uncertainty in deep learning. In *Advances in Neural Information Processing Systems*, pp. 13153–13164, 2019.

James E Matheson and Robert L Winkler. Scoring rules for continuous probability distributions. *Management science*, 22(10):1087–1096, 1976.

Dina Mistry, Maria Litvinova, Matteo Chinazzi, Laura Fumanelli, Marcelo FC Gomes, Syed A Haque, Quan-Hui Liu, Kunpeng Mu, Xinyue Xiong, M Elizabeth Halloran, et al. Inferring high-resolution human mixing patterns for disease modeling. *arXiv preprint arXiv:2003.01214*, 2020.

Radford M Neal. *Bayesian learning for neural networks*, volume 118. Springer Science & Business Media, 2012.

Ian Osband, Charles Blundell, Alexander Pritzel, and Benjamin Van Roy. Deep exploration via bootstrapped dqn. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 29*, pp. 4026–4034. 2016.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in neural information processing systems*, pp. 8026–8037, 2019.

Tim Pearce, Alexandra Brintrup, Mohamed Zaki, and Andy Neely. High-quality prediction intervals for deep learning: A distribution-free, ensembled approach. In *International Conference on Machine Learning*, pp. 4075–4084. PMLR, 2018.

Xin Qiu, Elliot Meyerson, and Risto Miikkulainen. Quantifying point-prediction uncertainty in neural networks via residual estimation with an i/o kernel. In *International Conference on Learning Representations*, 2019.

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. *arXiv preprint arXiv:1401.4082*, 2014.

David Salinas, Valentin Flunkert, Jan Gasthaus, and Tim Januschowski. Deepar: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting*, 36(3): 1181–1191, 2020.

Xiaocheng Shang, Zhanxing Zhu, Benedict Leimkuhler, and Amos J Storkey. Covariance-controlled adaptive langevin thermostat for large-scale bayesian sampling. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 28*, pp. 37–45. 2015.

Alexander Shekhovtsov and Boris Flach. Feed-forward propagation in probabilistic neural networks with categorical and max layers. In *International Conference on Learning Representations*, 2018.

Filippo Simini, Marta C González, Amos Maritan, and Albert-László Barabási. A universal model for mobility and migration patterns. *Nature*, 484(7392):96–100, 2012.

Natasa Tagasovska and David Lopez-Paz. Single-model uncertainties for deep learning. In *Advances in Neural Information Processing Systems*, pp. 6417–6428, 2019.

Michele Tizzoni, Paolo Bajardi, Chiara Poletto, José J Ramasco, Duygu Balcan, Bruno Gonçalves, Nicola Perra, Vittoria Colizza, and Alessandro Vespignani. Real-time numerical forecast of global epidemic spreading: case study of 2009 a/h1n1pdm. *BMC medicine*, 10(1):165, 2012.

Thomas Vandal, Evan Kodra, Jennifer Dy, Sangram Ganguly, Ramakrishna Nemani, and Auroop R Ganguly. Quantifying uncertainty in discrete-continuous and skewed data with bayesian deep learning. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 2377–2386. ACM, 2018.

Bin Wang, Jie Lu, Zheng Yan, Huaishao Luo, Tianrui Li, Yu Zheng, and Guangquan Zhang. Deep uncertainty quantification: A machine learning approach for weather forecasting. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 2087–2095. ACM, 2019a.

Hao Wang, Xingjian Shi, and Dit-Yan Yeung. Natural-parameter networks: A class of probabilistic neural networks. *Advances in Neural Information Processing Systems*, 29:118–126, 2016.

Yunbo Wang, Mingsheng Long, Jianmin Wang, Zhifeng Gao, and S Yu Philip. Predrnn: Recurrent neural networks for predictive learning using spatiotemporal lstms. In *Advances in Neural Information Processing Systems*, pp. 879–888, 2017.

Yuyang Wang, Alex Smola, Danielle Maddix, Jan Gasthaus, Dean Foster, and Tim Januschowski. Deep factors for forecasting. In *International Conference on Machine Learning*, pp. 6607–6617, 2019b.

Max Welling and Yee W Teh. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pp. 681–688, 2011.

Andrew Gordon Wilson and Pavel Izmailov. Bayesian deep learning and a probabilistic perspective of generalization. *arXiv:2002.08791*, 2020.

Shi Xingjian, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In *Advances in neural information processing systems*, pp. 802–810, 2015.

Bing Yu, Haoteng Yin, and Zhanxing Zhu. Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pp. 3634–3640, 2018.

Hsiang-Fu Yu, Nikhil Rao, and Inderjit S Dhillon. Temporal regularized matrix factorization for high-dimensional time series prediction. In *Advances in neural information processing systems*, pp. 847–855, 2016.

Rose Yu, Mohammad Taha Bahadori, and Yan Liu. Fast multivariate spatio-temporal analysis via low-rank tensor learning. In *Advances in Neural Information Processing Systems*, pp. 3491–3499, 2014.

Jingzhao Zhang, Tianxing He, Suvrit Sra, and Ali Jadbabaie. Why gradient clipping accelerates training: A theoretical justification for adaptivity. *arXiv:1905.11881*, 2019.

Qian Zhang, Nicola Perra, Daniela Perrotta, Michele Tizzoni, Daniela Paolotti, and Alessandro Vespignani. Forecasting seasonal influenza fusing digital indicators and a mechanistic disease model. In *Proceedings of the 26th international conference on world wide web*, pp. 311–319, 2017a.

Qian Zhang, Kaiyuan Sun, Matteo Chinazzi, Ana Pastore y Piontti, Natalie E Dean, Diana Patricia Rojas, Stefano Merler, Dina Mistry, Piero Poletti, Luca Rossi, et al. Spread of Zika virus in the Americas. *Proceedings of the National Academy of Sciences*, 114(22):E4334–E4343, 2017b.

Lingxue Zhu and Nikolay Laptev. Deep and confident prediction for time series at uber. In *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*, pp. 103–110. IEEE, 2017.

A COMPARISON AMONG METHODS

Method	Parallel computing resource	Small sample	Asymptotic consistency	Accuracy	Uncertainty
`Bootstrap`	25		✓	✓	
`Quantile`	1			✓	✓
`MIS`	1			✓	✓✓
`MC Dropout`	1			✓	
`SG-MCMC`	25	✓	✓	✓✓	✓

Table 3: Comparison of different deep uncertainty quantification methods for forecasts. Double check marks represent robustly highest performance in experiments.

B THEORETICAL ANALYSIS

Proof of Proposition 1. Since the distribution $\mathbb{P}_Z$ of Z has probability density p_Z,

$$\mathrm{MIS}_\infty(u, l; \rho) = (u - l) + \frac{2}{\rho}\left(\int_u^\infty (z - u)p_Z(z)\mathrm{d}z + \int_{-\infty}^l (l - z)p_Z(z)\mathrm{d}z\right).$$

We demonstrate in the following that the minimum of $\mathrm{MIS}_\infty(u, l; \rho)$ is achieved when u and l define the $(1 - \rho)$ confidence level.

For simplicity of exposition, we let p_Z be symmetric around 0 and let $l = -u$. Then

$$\mathrm{MIS}_\infty(u; \rho) = 2u + \frac{4}{\rho}\left(\int_u^\infty (z - u)p_Z(z)\mathrm{d}z\right).$$

Setting

$$0 = \frac{\mathrm{d}}{\mathrm{d}u}\mathrm{MIS}_\infty(u; \rho) = 2 - \frac{4}{\rho}\left(\int_u^\infty p_Z(z)\mathrm{d}z\right),$$

we reach the conclusion that the upper bound u^* that achieves the minimum MIS satisfies: $\int_{u^*}^\infty p_Z(z)\mathrm{d}z = \frac{\rho}{2}$. Therefore, $[l^*, u^*]$ defines the $(1 - \rho)$ confidence level. □

Proof of Proposition 2. We think about minimizing the MIS score over lower and upper bounds l and u, given samples $z_1, \cdots, z_N$ from the posterior distribution:

$$(u, l) = \arg\min_{u>l;u,l\in\mathbb{R}} \mathrm{MIS}_N(u, l) \tag{9}$$

$$= \arg\min_{u>l;u,l\in\mathbb{R}} \left((u - l) + \frac{1}{N} \cdot \frac{2}{\rho}\sum_{i=1}^N ((z_i - u)\mathbb{1}\{z_i > u\} + (l - z_i)\mathbb{1}\{z_i < l\})\right). \tag{10}$$

We prove in the following that the minimum of MIS score is achieved when we sort $\{z_1, \cdots, z_N\}$ in an increasing order and take $l = z_{\lceil\rho\cdot N/2\rceil}$ and $u = z_{N-\lfloor\rho\cdot N/2\rfloor}$, which define the quantile of the empirical distribution formed by the samples $\{z_1, \cdots, z_N\}$.

We first note that $\mathrm{MIS}_N(u, l)$ is a continuous function with respect to u and l. Since $\mathrm{MIS}_N(u, l)$ is piece-wise linear, we simply need to check that

$$\begin{aligned}\mathrm{MIS}_N(z_j^-, l) &= (z_j - l) + \frac{1}{N} \cdot \frac{2}{\rho}\sum_{i\neq j} ((z_i - u)\mathbb{1}\{z_i > u\} + (l - z_i)\mathbb{1}\{z_i < l\}) \\ &= \mathrm{MIS}_N(z_j^+, l).\end{aligned}$$

The result holds similarly for l.

In what follows we prove that $\mathrm{MIS}_N(u, l)$ is decreasing in l whenever $l \leq z_{\lceil\rho\cdot N/2\rceil}$. Whenever $l \geq z_{\lceil\rho\cdot N/2\rceil}$, $MIS(u, l)$ is increasing in l. We can use similar reasoning to prove that minimum is achieved when $u = z_{N-\lfloor\rho\cdot N/2\rfloor}$.

Consider derivative of $\mathrm{MIS}_N(u,l)$ over l

$$\frac{\partial \mathrm{MIS}_N(u,l)}{\partial l} = -1 + \frac{2}{\rho \cdot N} \sum_{i=1}^{N} \frac{\partial \left((l-z_i)\mathbb{1}\left\{z_i < l\right\}\right)}{\partial l}. \tag{11}$$

Since

$$(l-z_i)\mathbb{1}\left\{z_i < l\right\} = \begin{cases} l - z_i & \text{if } z_i < l \\ 0 & \text{if } z_i \geq l \end{cases}, \tag{12}$$

$$\frac{\partial \left((l-z_i)\mathbb{1}\left\{z_i < l\right\}\right)}{\partial l} = \begin{cases} 1 & \text{if } z_i < l \\ 0 & \text{if } z_i \geq l \end{cases}. \tag{13}$$

When $l \leq z_{\lceil \rho \cdot N/2 \rceil}$, $\frac{2}{\rho \cdot N} \sum_{i=1}^{N} \frac{\partial((l-z_i)\mathbb{1}\{z_i<l\})}{\partial l} \leq 1$. Hence $\mathrm{MIS}_N(u,l)$ is decreasing in l whenever $l \leq z_{\lceil \rho \cdot N/2 \rceil}$. Whenever $l \geq z_{\lceil \rho \cdot N/2 \rceil}$, $\frac{2}{\rho \cdot N} \sum_{i=1}^{N} \frac{\partial((l-z_i)\mathbb{1}\{z_i<l\})}{\partial l} \geq 1$, making $\mathrm{MIS}_N(u,l)$ increasing in l. Therefore, $l = z_{\lceil \rho \cdot N/2 \rceil}$ is the minimum of $\mathrm{MIS}_N(u,l)$.

The above two facts complete the proof and conclude that

$$\arg\min_{u>l;u,l\in\mathbb{R}} \mathrm{MIS}_N(u,l) = (z_{N-\lfloor \rho \cdot N/2 \rfloor}, z_{\lceil \rho \cdot N/2 \rceil}), \tag{14}$$

which define the quantile of the empirical distribution formed by the samples $\{z_1, \cdots, z_N\}$. □

C TRAFFIC FORECASTING EXPERIMENTS

C.1 UQ METHODS SETUP

DCRNN setup The model has two hidden layers of RNN with 64 units. The filter type is a dual random walk and the diffusion step is 2. The learning rate of DCRNN is fixed at $1e^{-2}$ with Adam optimizer (Kingma & Ba, 2014). We perform early stopping at 50 epochs.

Bootstrap For Bootstrap method, we randomly dropped 50% of training data while keeping the original validation and testing data. We obtain 25 samples for constructing mean prediction and confidence interval.

Quantile regression We apply pinball loss function (Koenker & Bassett Jr, 1978; Koenker, 2005) to train three different quantiles (0.025, 0.5, 0.975). The model and learning rate setup is the same as DCRNN. The result for comparison averages the performance of 3 trails.

SQ regression We use linear spline quantile function to approximate a quantile function and use the CRPS as the loss function. For every point prediction, there are 11 trained parameters to construct the quantile function. The 1st parameter is the intercept term. The next 5 can be transformed to the slopes of 5 line segments. The last 5 can be transformed to a vector of the 5 knots' positions. The model and learning rate setup is the same as DCRNN. The result for comparison averages the performance of 3 trails.

MIS regression We combine MAE with MIS and directly minimize this loss function. The model and learning rate setup is the same as DCRNN. The result for comparison averages the performance of 3 trails.

MC Dropout The training process of MC Dropout is the same as DCRNN. We implement the algorithm provided by (Zhu & Laptev, 2017) and simplify the model by only considering the model uncertainty. We apply random dropout through the testing process with 5% drop rate and iterate 50 times to achieve a stable prediction. The result for comparison averages the performance of 3 trails.

SG-MCMC The learning rate of SG-MCMC is $5e^{-4}$, and we selected a Gaussian prior $\mathcal{N}(0, 4.0)$ with random initialization as $\mathcal{N}(0, 0.2)$. We note here a symmetric Gamma prior could also work in this case with $\Gamma(0.1, 1)$. We apply early stopping at epoch 50, and it helps to improve generalization on testing set. Our result is averaged from 25 posterior samples.

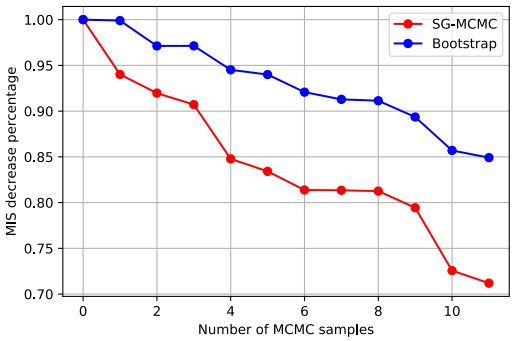

Figure 3: MIS of Bootstrap and SG-MCMC with more samples for COVID forecasting.

C.2 Sample complexity of bootstrap and SG-MCMC

General bootstrap and SG-MCMC could provide asymptotic consistency, yet the sample complexity might be large for a complex dataset. In Figure 3, we could observe a decrease in MIS when the number of (posterior) samples goes up, and SG-MCMC seems to converge faster than Bootstrap. It is clear that for both Bootstrap and SG-MCMC methods, they do not converge in MIS. This indicates that our reported MIS for Bootstrap and SG-MCMC in this paper could potentially be much lower. However, due to the constraint in computation, we do not train until the MIS converges. We think, in this setup, to determine a theoretical bound of sample complexity is highly non-trivial, so we left this to our future work.

D COVID-19 forecasting experiment

D.1 Global Epidemic and Mobility Model.

The Global Epidemic and Mobility model (GLEAM) is a stochastic spatial epidemic model in which the world is divided into over 3,200 geographic subpopulations constructed using a Voronoi tessellation of the Earth's surface. Subpopulations are centered around major transportation hubs (e.g. airports) and consist of cells with a resolution of approximately 25 x 25 kilometers (Balcan et al., 2009; 2010; Tizzoni et al., 2012; Zhang et al., 2017b; Chinazzi et al., 2020; Davis et al., 2020). High resolution data are used to define the population of each cell (sed). Other attributes of individual subpopulations, such as age-specific contact patterns, health infrastructure, etc., are added according to available data (Mistry et al., 2020).

GLEAM integrates a human mobility layer - represented as a network - that uses both short-range (i.e. commuting) and long-range (i.e. flights) mobility data from the Offices of Statistics for 30 countries on 5 continents as well as the Official Aviation Guide (OAG) and IATA databases (updated in 2020). The air travel network consists of the daily passenger flows between airport pairs (origin and destination) worldwide mapped to the corresponding subpopulations. Where information is not available, the short-range mobility layer is generated synthetically by relying on the "gravity law" or the more recent "radiation law" both calibrated using real data (Simini et al., 2012).

The model is calibrated to realistically describe the evolution of the COVID-19 pandemic as detailed in Chinazzi et al. (2020); Davis et al. (2020). Lastly, GLEAM is stochastic and produces an ensemble of possible epidemic outcomes for each set of initial conditions. To account for the potentially different reporting level of the states, a free parameter Infection Fatality Rate (IFR) multiplier is added to each model. To calibrate and select the most reasonable outcomes, we filter the models by the latest hospitalization trends and confirmed cases trends, and then we select and weight the filtered models using Akaike Information Criterion (Zhang et al., 2017a). The forecast of the evolution of the epidemic is formed by the final ensemble of the selected models.

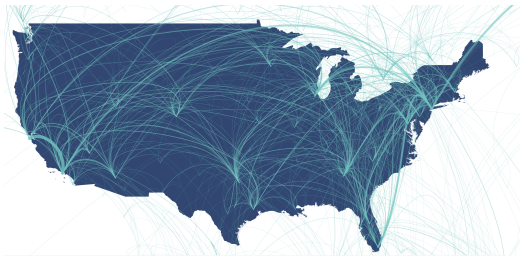

Figure 4: Original flight network connecting US airports. Data is aggregated at the State level to construct the network State-to-State graph.

D.2 DEEPGLEAM AND UQ METHODS FOR DEEPGLEAM SETUP

We use an encoder-decoder sequence to sequence learning framework in the DCRNN structure. The encoder reads as input a $7 \times 50 \times 4$ tensor that comprises the daily residuals between the observed death number and the GLEAM forecasts for the 50 US states over 4 different prediction horizons. It encodes the information in 7 hidden layers. The decoder produces forecasts of the weekly residuals between the true death number and the GLEAM forecasts for each state for the following 4 weeks. We perform autoregressive weekly death predictions (from one week ahead to four weeks ahead).

DCRNN setup The model only has one hidden layer of RNN with 8 units to overcome the overfitting problem. The filter type is Laplacian and the diffusion step is 1. The base learning rate of DCRNN is $1e^{-2}$ and decay to $1e^{-3}$ at epoch 13 with Adam optimizer (Kingma & Ba, 2014). We have a strict early stopping policy to deal with the overfitting problem. The training stops as the validation error does not improve for three epochs after epoch 13.

Bootstrap For Bootstrap method, due to the small sample we have (typically 25), we only randomly dropped 1 training data while keeping the original validation and testing data. We obtain 25 samples for constructing mean prediction and confidence interval.

Quantile regression We apply pinball loss function (Koenker & Bassett Jr, 1978; Koenker, 2005) to train three different quantiles (0.025, 0.5, 0.975). The model and learning rate setup is the same as DCRNN. The result for comparison averages the performance of 10 trails.

SQ regression We use linear spline quantile function to approximate a quantile function and use the CRPS as the loss function. For every point prediction, there are 11 trained parameters to construct the quantile function. The 1$^{\text{st}}$ parameter is the intercept term. The next 5 can be transformed to the slopes of 5 line segments. The last 5 can be transformed to a vector of the 5 knots' positions. The base learning rate is $1e^{-1}$ and decay to $1e^{-2}$ at epoch 50. The early stopping policy is the same as DCRNN. The result for comparison averages the performance of 10 trails.

MIS regression We combine MAE with MIS and directly minimize this loss function. The model and learning rate setup is the same as DCRNN. The result for comparison averages the performance of 10 trails.

MC Dropout The training process of MC Dropout is the same as DCRNN. We implement the algorithm provided by (Zhu & Laptev, 2017) and simplify the model by only considering the model uncertainty. We apply random dropout through the testing process with 5% drop rate and iterate 300 times to achieve a stable prediction. The result for comparison averages the performance of 10 trails.

SG-MCMC The learning rate of SG-MCMC is $5e^{-4}$, and we selected a Gaussian prior $\mathcal{N}(0, 2.0)$ with random initialization as $\mathcal{N}(0, 0.05)$. We apply training for 800 epochs, and it early stops as long as the validation error does not improve for 50 epochs. Our result is averaged from 25 posterior samples.

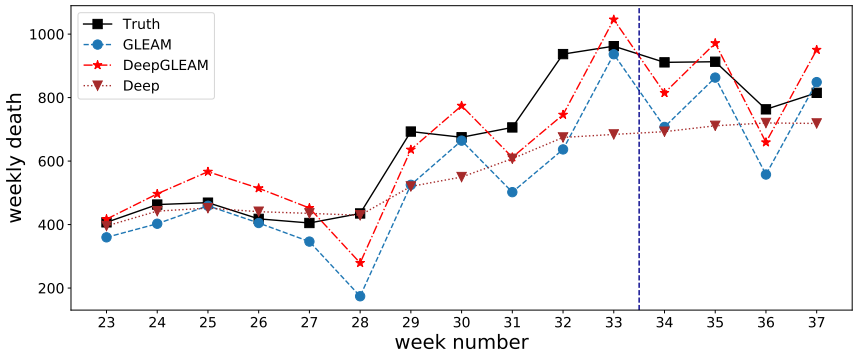

Figure 5: California One Week ahead Prediction

T	DeepGLEAM (DCRNN)	DeepGLEAM (SG-MCMC)	Deep (DCRNN)	GLEAM
1W	66.03	58.30	239.94	73.59
2W	57.67	46.61	213.31	65.46
3W	70.12	59.27	189.49	73.75
4W	70.75	70.57	161.88	70.16

Table 4: RMSE comparison of different approaches for COVID-19 mortality forecasting

D.3 Performance of Deep, DeepGLEAM, and GLEAM

In Figure 5, we provide an example to compare the performance of Deep, GLEAM, and DeepGLEAM models in California. The input data (for training or validation) are from weeks before 34, and we start the prediction from week 34, labeled by the vertical dash line. It can be observed that Deep model fails to predict the dynamics of COVID-19 evolution. We quantitively compare the accuracy among DeepGLEAM (DCRNN), DeepGLEAM (SG-MCMC), Deep (DCRNN), and GLEAM models using RMSE. DeepGLEAM (DCRNN) and Deep (DCRNN) are deterministic models while DeepGLEAM (SG-MCMC) is the statistical model with the best accuracy. Table 4 shows both DeepGLEAM models outperform GLEAM while the Deep model is much worse than GLEAM. By calculation, there is a 6.6% improvement for DeepGLEAM (DCRNN) and 17% improvement for DeepGLEAM (SG-MCMC) on average from GLEAM.

One reason for this phenomenon is the distribution shifts in the epidemiology dynamics. Without background knowledge, deep neural networks do not have enough inductive bias to guide their predictions. On the other hand, GLEAM and DeepGLEAM leverage mechanistic knowledge about disease transmission dynamics to infer underlying dynamic change.

D.4 Performance of Ensemble DeepGLEAM model

We also tried the ensemble DeepGLEAM model which predicts the next 4 weeks death prediction together from the first decoder. Table 5 shows the performance among the UQ methods, which shares similar results with the autoregressive model.

Table 5: Performance comparison of different approaches on Ensemble DeepGLEAM Model for COVID-19 mortality forecasting.

T	Metric	DCRNN	Bootstrap	Quantile	SQ	MAE-MIS	MC Drpoout	SG-MCMC
One week ahead	RMSE	68.56	70.73	70.26	73.82	76.32	68.58	**62.42**
	MIS	—	1216.63	430.72	1299.78	**424.49**	831.86	715.40
Two weeks ahead	RMSE	59.26	61.69	60.94	65.44	65.39	59.27	**51.21**
	MIS	—	1135.01	**331.94**	1272.82	368.20	796.54	727.27
Three weeks ahead	RMSE	70.35	73.27	71.72	73.88	74.95	70.32	**66.72**
	MIS	—	1464.52	**392.66**	1559.62	416.23	997.65	930.55
Four weeks ahead	RMSE	72.29	72.41	73.44	70.44	75.16	72.24	**68.40**
	MIS	—	1482.08	**418.77**	1586.95	468.49	1034.68	971.55

D.5 PREDICTIONS OF THE REST 42 STATES

The data before week 32 has been seen during training or evaluation, therefore we focus on the result to the right of the vertical dashed line.

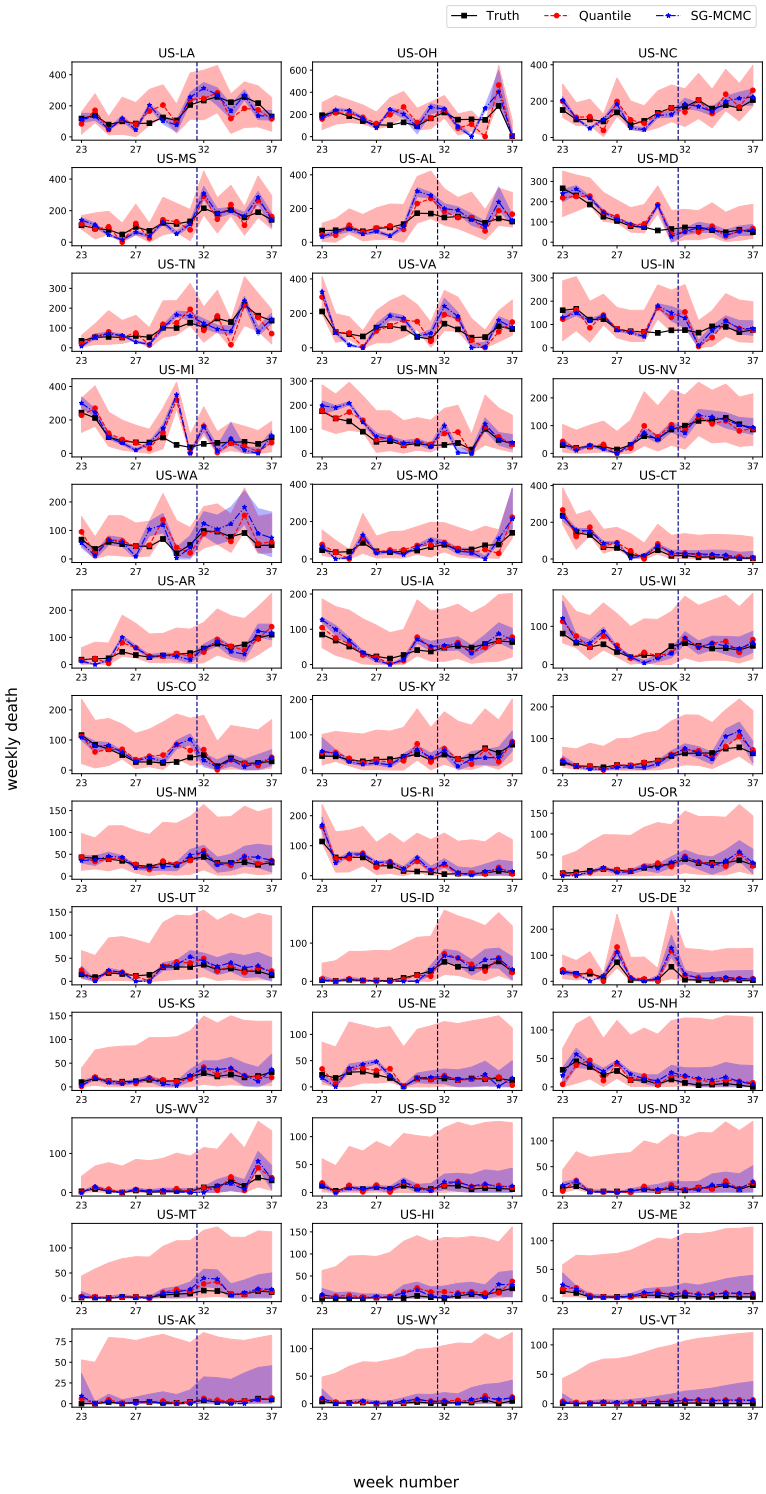

Figure 6: One week ahead COV-19 prediction for the rest 42 states.