AIAA AVIATION Forum
June 15-19, 2020, VIRTUAL EVENT
AIAA AVIATION 2020 FORUM

10.2514/6.2020-2897

Probabilistic Aircraft Trajectory Prediction with Weather Uncertainties using Approximate Bayesian Variational Inference to Neural Networks

Yutian Pang*, Yuhao Wang† , Yongming Liu‡
Arizona State University, Tempe, Arizona, 85287, USA

A key consideration in Trajectory Prediction (TP) tools is the confidence that can be placed on the prediction. We purpose a non-deterministic TP neural network using tractable approximate Bayesian variational inference for the model parameters considering weather effects. This work adopts the state-of-art in Bayesian Deep Learning research and builds a neural network model with stochastic convolutional, recurrent, and fully-connect layers. The purposed stochastic variational method outperforms the dropout approximate to Variational Inference and performs reliable uncertainty estimates. It can be easily applied to most neural net architectures and also provides a simple pruning heuristic that can drastically reduce the number of model parameters compares to ensemble methods. The experiment is conducted with the Atlanta Air Route Traffic Control Center (ZTL) flight data and the corridor integrated weather system (CIWS) weather data from Sherlock Data Warehouse (SDW) on June 24th, 2019. The experimental results show better variance reduction than dropout-based methods. The uncertainty estimates are more reliable thanks to the Kullback–Leibler divergence (KL-divergence) term within the optimization objective.

I. Nomenclature

X, Y	=	training dataset
x^*, y^*	=	testing dataset
ω	=	model parameter
$q_\theta(\omega)$	=	variational distribution
$p(\omega)$	=	weight prior
μ	=	posterior mean
ρ	=	posterior standard deviation
ATM	=	air traffic management
TP	=	trajectory prediction
VI	=	variational inference
ZTL	=	atlanta air route traffic control center
SDW	=	sherlock data warehouse
DST	=	decision support tools
CDR	=	conflict detection and resolution
BNN	=	bayesian neural networks
CNN	=	convolutional neural network
GAN	=	generative adversarial network
RNN	=	recurrent neural network
$LSTM$	=	long short-term memory
CI	=	confidence interval
SRT	=	stochastic regularization technique

*Research Associate, School for Engineering of Matter, Transport and Energy, Yutian.Pang@asu.edu.
†Research Associate, School for Engineering of Matter, Transport and Energy, ywang542@asu.edu.
‡Professor, School for Engineering of Matter, Transport and Energy, Yongming.Liu@asu.edu.

II. Introduction

There is an urgent need to develop an accurate, significantly more strategic and reliable method for future Air Traffic Management (ATM) system, which is also referred to as the next generation (NextGen) national air transportation system [1]. A key prerequisite of NextGen is the capability to predict and share air traffic trajectories and conflicts to all involved relevant stakeholders, in order to strategically optimize a safe air traffic flow. These capabilities are provided by the development of decision support tools (DSTs), such as trajectory prediction (TP) and conflict detection and resolution (CDR) tools. This is achieved with Decision Support Tools (DSTs), including Trajectory Prediction (TP) and Conflict Detection and Resolution (CDR) tools [2]. High-performance TP and CDR tools have the potential to increase efficiency and reduce safety concerns in air traffic flows, by improving the capability to perform necessary avoidance and separation adjustments in advance. The performance and reliability of TP tools will thus have significant importance. It is supposed to efficiently and safely accommodate the increasing air traffic flows within the United States Airspace (UAS). There has been shown to have non-negligible trajectory uncertainties during aviation operations. The uncertainty of trajectory predictions comes from multiple sources. A typical trajectory planning tool makes decision-based on the aircraft's performance, pilot intends and Terminal Radar Approach Control (TRACON) regulations. The unawareness of pilot intend, assumptions to aircraft's conditions, where there is a limit knowledge, results in the uncertainty of the prediction. Also, when multiple aircraft are heading to the same regions, the communication and avoiding maneuver between each of the aircraft will cause the uncertainty of the trajectory prediction. Furthermore, in addition to these kinds of model uncertainties, environmental factors also cause uncertainty. The convective weather prediction has a certain resolution, accuracy and forecasting interval. However, the convective weather condition usually generates rapidly and poses a significant danger to aviation operations. The adjustments of trajectory coordinate with the weather conditions also result in uncertainties. All of these uncertainties lead to the insufficient, unreliable prediction result of deterministic model output.

It's critical to understand and measure the confidence level of the prediction. A key consideration in the development of TP is the confidence of the model prediction. TP Integrity (TPI) and TP time (TPT) [3] are two evaluation metrics for the performance of TP tools. TPI is defined to measure the level of prediction accuracy. It refers to the probability that a given level of prediction accuracy will be met over a certain look-ahead time (LAT) and time between prediction and execution (TBPE). It's can be concluded to have four performance levels: en-route integrity requirements, TMA integrity requirements, precision approach integrity requirements, and airport surface movement integrity requirements. Furthermore, the measurement of the model responding time (computational requirements) when performing TP is proposed during practical applications. And this is the formulation of TPT in the field. The computational requirements provide a measure of the capability of the TP tool to provide the required level of continuity and availability.

Neural Networks (NNs) have achieved a number of milestones during the past decade due to the drastic improvements in computational power. Advances in computer hardware, deep learning models and data availability have enabled the use of deep neural networks into different research fields. Meanwhile, the inherent characteristics of NN models are prone to overfitting, especially for recurrent neural networks (RNNs) [4]. This requires the fine-tuning of network parameters by stochastic regularization techniques (SRTs) such as dropout. Moreover, the neural network lacks a proper theoretical theorem as a foundation. Researchers always choose the parameters of the NN model by *best practice* during their research. Bayesian techniques are known as a remedy to address these issues. The neural network that incorporates Bayesian variational inference, Bayesian Neural Networks (BNN) is more robust to overfitting problems to control the model complexity [5]. Instead of a single *point estimate* set of model parameters computed from the NN, the Bayesian methods yield distribution for each of the parameters which provide an inherent estimate of prediction uncertainty. The uncertainty, introduced into the BNN by the posterior predictive distribution, is a vital feature to engineering, also aviation applications. The prior settings in Bayesian approaches also include prior knowledge into NN such that incorporate physical constraints into the process. This will also reduce the amount of data needed for training neural nets by adding prior knowledge. There are also drawbacks to BNNs as well, which is also the reason there are not too much research works in this direction. The main complication is the computational intractability in the integrals of the Bayesian posterior distributions. Such solutions are complicated for even the simplest network architectures such as single-layer feed-forward networks with linear outputs [6]. There have been significant advances in approximate inference, but these are still much more computationally expensive compared to most frequent inference [7]. However, the approach taken here doesn't require analytical solutions but search instead for variational distributions whose expectation values can be efficiently approximated with numerical integration. This results in a stochastic method for variational inference with a diagonal Gaussian posterior that can be applied to any differentiable log-loss parametric model, which includes most neural networks [6].

We develop the approximate Bayesian variational inference framework for the non-deterministic TP based on the flight plan and the convective weather forecasting. This is also a key objective of the NASA University Leadership Initiative (ULI) project [8] which requires researchers to address the safety needs and their technology solutions for future national airspace system (NAS) for both manned and unmanned aircraft [9]. The inputs to the model are the flight plan before takeoff and weather conditions along with the flight plan. The model output is the predicted flight route with a confidence interval (CI). The scope of this paper is to apply the theory of practical variational inference BNN for aircraft trajectory predicting task. The model is capable of predicting the aircraft trajectory with an uncertainty bound.

This paper is organized as follows: Section II is the brief introduction of Air Traffic Management research as well as Bayesian Neural Network models including the benefits and drawbacks. In section III, we discuss the related work and the development of Bayesian Variational Inference by other researchers. Section IV discusses the data used for evaluation, Bayesian derivation as well as network architecture in this research. The following Section V is a discussion of the experimental results using the data processed and model defined. Section VI is the conclusion and comments of this research.

III. Related Work

Researchers from different backgrounds have made contributions to the development of weather-related trajectory prediction tools. An FAA report [10] review 20 trajectory predictor strategies developed and a subset of 282 documents published between the years of 2003 and 2009. The documents are evidence that an extensive amount of research, by several aviation communities, has been conducted over the past decade [10]. A detailed list of algorithms, data collection techniques and synthesis of research papers is concluded in the report. The valuable future research directions on TP are also raised out. However, most of the tools in the public domain are geared towards short-term predictions instead of long-term predictions. The key issue reported with long-term predictions is the increase in uncertainties of the contextual factors such as weather uncertainties. They also reported that none of the models developed incorporates the ability to account for the impact of uncertainties in the TP model [2].

There are several selection based TP research works such as Hidden Markov Model (HMM) [11], generalized linear model (GLM) [12], Gaussian Mixture Models (GMM) [13] and indicator selection based on radar measurements of flight tracks [14]. This type of TP tools requires the dataset to be sufficient for covering all possible flight track candidates. There are also research works treating the objective as a regression task and predicting trajectories with machine learning algorithms such as a recurrent neural network (RNN) [15, 16] and Bayesian updating [17]. Despite from pure data-driven method, another research work purposed what's called physics-based learning for the trajectory prediction task. The idea is to integrate the underlying physics of aircraft dynamical systems into machine learning models to reduce the training costs and enhance simulation performances [18]. This approach is found to exhibit superior computation efficiency compared with other classical numerical methods. Another part of the TP works focuses on the probabilistic approach (non-deterministic approach). The aircraft position forecasting work using the flight plan and the current position as inputs to predict the future positions of aircraft with uncertainties in both near-term and mid-term future [19, 20]. They propose an expression for the probability of conflict using the provided quantitative bounds. Based on the expression, the model is able to generate potential fields for probabilistic path planning of the aircraft. Validation of the purposed method is validated using Monte Carlo simulations. Intention based prediction also served as a key focus in the field. The research on TP based on pilots' intention [21] first infer pilots' intention then do prediction based on the inference. Another work proposed a stochastic linear hybrid system to describe the dynamics of an aircraft when changing flight modes [22]. This work incorporates the intention to model the dynamics. This algorithm is claimed to be computationally efficient and can have better prediction power compare to other methods. Another recent study using a deep generative convolutional recurrent neural network (RNN) approach for 4D trajectory prediction, as of the authors' knowledge, is the first paper using an encoder-decoder recurrent neural structure for this task [23]. The paper proposes an end-to-end convolutional recurrent neural network that consists of a long short-term memory (LSTM) encoder network and a mixture density LSTM decoder network. The model can predict the aircraft 4D trajectories using high-dimensional weather features and last filed flight plans and the prediction error metrics show that average absolute horizontal errors are around 50 nautical miles and 2800 feet for average vertical errors. The similar work to this approach is trying to perform TP with a conditional generative adversarial net to reduce the amount of data required for the model training process [24]. Another non-negligible work related to BNN is the Bernoulli approximate variational inference for NNs to predict aircraft trajectories [25], which is as simple as using dropout after each of the layers during the test stage. This method is simple and efficient. However, the choice of dropout ratio would influence the output variance. There are a variety of other research topics in the trajectory prediction area. A sensitivity study on trajectory predictions in ATM introduces the concept of closed-loop sensitivities [26], which are defined as the difference between actual and computed trajectories per unit of modeling errors into pilot feedback controls. This method is applied and validated by the TRACON automation system as well as flight management systems.

IV. Preliminaries

Plain feedforward neural network are prone to overfitting [27]. Various SRTs have been developed to address this issue such as early stopping, dropout and weight decay. The recent purposed efficient, backpropagation compatible algorithm for learning a probability distribution on neural network nets, *Bayes by Backprop* is also one of the regularization techniques built upon Bayesian inference on neural network weights. The formulation is as simple as putting a distribution over each of the neural network weight, rather than having a single fixed value. If we want to perform backpropagation on a network like this, we need to train an ensemble of network instead where each set of network weights are sampled from the distribution. However, this is intractable in the sense of computation power. Instead the proposed method takes a variational approximation to exact Bayesian updates. The number of weights in the neural network is only two times larger, making the training procedure more tractable. In this section, we will review the principles and knowledge involved.

A. Variational Inference

The formulation of model uncertainty relies on probabilistic modeling, Bayesian modeling specifically. If given a training inputs $X = x_1, ..., x_n$ and their corresponding outputs $Y = y_1, ..., y_n$, we try to find the parameters ω for the approximation function $y = f^{\omega}(x)$ that are *most likely* to generate the outputs. The Bayesian approach gives a space of parameters ω as a distribution called the *prior*. The *prior* is defined based on prior knowledge and gives us an adversarial of what parameters are likely to generate our data before we have any data points. Bayesian approach also defines a likelihood distribution $p(y|x, \omega)$. The likelihood distribution is a probabilistic model of the model outputs given the data points and model parameters. Then we can calculate predict the output for a new input point x^* by Bayesian inference:

$$p(y^*|x^*, X, Y) = \int p(y^*|x^*, \omega) p(\omega|X, Y) d\omega \tag{1}$$

where $p(\omega|X, Y)$ is the *posterior* by Bayesian theorem,

$$p(\omega|X, Y) = \frac{p(Y|X, \omega) p(\omega)}{p(Y|X)} \tag{2}$$

The normaliser in the posterior evaluation $p(Y|X)$ is also called the *model evidence* [28]. In practical, the posterior usually cannot be evaluated analytically. Thus we defined an approximating *variational* distribution $q_\theta(\omega)$ for the posterior. Thus we minimize the Kullback-Leibler (KL) divergence [29] between our approximation and the posterior w.r.t. θ,

$$KL(q_\theta(\omega)||p(\omega|X, Y)) = \int q_\theta(\omega) log \frac{q_\theta(\omega)}{p(\omega|X, Y)} \, d\omega \tag{3}$$

Minimising the KL divergence allows us to approximate the predictive distribution as,

$$q_\theta^*(y^*|x^*) := p(y^*|x^*, X, Y) = \int p(y^*|x^*, \omega) q_\theta^*(\omega) \, d\omega \tag{4}$$

This is also equivalent to maximising the evidence lower bound (ELBO) w.r.t. the variational parameters $q_\theta(\omega)$,

$$\mathcal{L}_{VI}(\theta) := \int q_\theta(\omega) \, log(p(Y|X, \omega)) \, d\omega - KL(q_\theta(\omega)||p(\omega)) \tag{5}$$

The above procedure is known as Variational Inference (VI) [30] which is a standard technique in Bayesian modeling. The recent advances in VI are used as the fundamental approach to approximate the posterior for deep Bayesian neural nets.

B. Bayesian Neural Network

Bayesian Neural Network (BNN) offers a probabilistic interpretation of deep learning models by inferring distributions over the models' weights as prior knowledge. To train the network given the prior distribution, variational posterior and training data, the loss function is defined as the negative ELBO,

$$\mathcal{L}_{VI}(\theta) := KL(q_\theta(\omega)||p(\omega)) - \mathbb{E}_{q_\theta(\omega)}[log(X,Y|\omega)] \tag{6}$$

The cost function of Eq .(6) is a sum of a data-dependent part, which we shall refer to as the likelihood cost, and a prior dependent part, which we shall refer to as the complexity cost. This formulates a trade-off between the complexity of the data and the simplicity of the prior. The gradient descent can be achieved by minimizing $\mathcal{L}_{VI}(\theta)$. However, the gradient of the two expectations in Eq. (6) cannot be computed explicitly but must be done by Monte Carlo Sampling if the BNN has more than one hidden layer. A backpropagation-like algorithm called *Bayes By Backprop* [27] is obtained for variational Bayesian inference in Neural Networks. *Bayes By Backprop* uses unbiased estimates of gradients of the loss function in Eq. (6) to learn a distribution over the weights of a neural network. The key idea behind it is the fact that under certain conditions, the derivative of an expectation is the expectation of the derivative as Eq. (7). The gradient of the loss function can be evaluated then. The model uncertainty in Bayesian networks is modeled by a technique called *Monte Carlo dropout* [31], which is essentially running several stochastic forward passes through the network and averaging the outputs.

$$\frac{\partial}{\partial\theta}\mathbb{E}_{q_\theta(\omega)}[f(\omega,\theta)] = \mathbb{E}_{q(\epsilon)}[\frac{\partial f(\omega,\theta)}{\partial\omega}\frac{\partial\omega}{\partial\theta} + \frac{\partial f(\omega,\theta)}{\partial\theta}] \tag{7}$$

The recent study on approximate Bayesian VI shows that a neural network with arbitrary depth and non-linearities, with Bernoulli dropout applied before every weight layer, is mathematically equivalent to an approximate variational inference which results in uncertainty estimates [31]. However, *Bayes by Backprop* doesn't assume the variational posterior as Bernoulli distributions but Gaussian distributions instead. The formulation of *Bayes by Backprop* suppose the variational posterior is a diagonal Gaussian distribution.

The weights of the posterior can be obtained by sampling a unit Gaussian, shifting it by the mean μ and the standard deviation σ. Then the parameters of the posterior are $\theta = (\mu, \sigma)$. They also parameters the standard deviation as $\sigma = \log(1 + \exp(\rho))$ to make σ non-negative. This makes the sampled posterior weights as $\mathbf{w} = \mu + \log(1 + \exp(\rho)) \circ \epsilon$. The cost is defined as,

$$f(\mathbf{w},\theta) = \log q(\mathbf{w}|\theta) - \log P(\mathbf{w})P(X,Y|\mathbf{w}) \tag{8}$$

where $\mathbf{w}$ denotes the weights drawn from the variational posterior distribution with Monte Carlo. And the gradients with respect to the mean μ and standard deviation ρ are,

$$\Delta_\mu = \frac{\partial f(\mathbf{w}|\theta)}{\partial\mathbf{w}} + \frac{\partial f(\mathbf{w}|\theta)}{\partial\mu} \tag{9}$$

$$\Delta_\rho = \frac{\partial f(\mathbf{w}|\theta)}{\partial\mathbf{w}}\frac{\epsilon}{1 + \exp(-\rho)} + \frac{\partial f(\mathbf{w}|\theta)}{\partial\mu} \tag{10}$$

Then we simply apply gradient descent to update the variational parameters. We could notice that the $\frac{\partial f(\mathbf{w}|\theta)}{\partial\mathbf{w}}$ term in both Eq. (9) and Eq. (10) are shared and are just the gradients found by normal backpropagation operations. Thus the *Bayes by Backprop* for the variational parameters are simple shifting and scaling the gradients of normal backpropagation. The experimental result shows that the performance from *Bayes by Backprop* is comparable to the dropout approximation [27]. This is called the local reparameterization trick.

A recent work introduces Flipout, an efficient method for decorrelating the weight gradients between different examples in a mini-batch could significantly reduce the variance in large batch settings for different network architectures, as well as speed up training [32]. The Flipout also outperforms dropout-based methods for regularizing the LSTMs. Flipout performs weight perturbations to the network weights ω which are sampled from the approximate distributions $q_\theta(\omega)$. It can be explicitly stated as $\omega = \overline{\omega} + \widehat{\Delta\omega}$, where $\overline{\omega}$ is the mean value and $\widehat{\Delta\omega}$ is the stochastic perturbation term. Eq. (11) shows the simple idea of Flipout perturbation to network weights where the subscript stands for the row index, ϕ denotes the activation function. s and r are random vectors independently, uniformly sampled from ±1 and $\circ$ means element-wise operation.

$$
\begin{aligned}
y_i^* &= \phi(\omega^T x_i^*) \\
&= \phi((\overline{\omega} + \widehat{\Delta\omega} \circ r_i s_i^T)^T x_i^*) \\
&= \phi(\overline{\omega}^T x_i^* + (\widehat{\Delta\omega}^T (x_i^* \circ s_i)) \circ r_i)
\end{aligned}
\tag{11}
$$

The forward pass of the network can be vectorized into Eq. (12) and we can backpropagate the forward pass to obtain the gradients for $\overline{\omega}$, $\widehat{\Delta\omega}$ and X.

$$
Y = \phi(X\overline{\omega} + ((X \circ S)\widehat{\Delta\omega}) \circ R) \tag{12}
$$

V. Methodology

A. Data Preparation

We use control sector flight data for our experiment instead of the whole flight tracks from departure airport to the arrival airport. Each of the control sector (ARTCC) is in charge of all of the flights flying within the airspace and each control sector may have a certain control behavior. Thus the prediction across multiple control sectors is not the best approach. The sector-based TP means our trained model is sector-specific and can handle flight tracks within the airspace of the current control sector. We choose to download the data on June 24th, 2019 since it is reported to have severe weather conditions that happened on this day*. The typical severe weather conditions we are looking at are tornados, large hail, and strong wind conditions. The record shows that strong wind and large hail is reported multiple times within the range of Atlanta Air Route Traffic Control Center (ZTL). Thus we are focusing on predicting the flight trajectories in ZTL on June 24th, 2019.

The data used for this research are obtained from the Sherlock Data Warehouse (SDW) [33]. Sherlock is a distributed big data platform for data visualization to support air traffic management (ATM) research, which includes a database, a web-based user interface (UI), a few data visualization tools and other services. It is a platform for reliable ATM data collection, archiving, processing, query, and delivery and can be used for big data analysis, including data mining and machine learning [33]. Data of Sherlock comes primarily from the federal aviation administration (FAA) and the National Oceanic Atmospheric Administration (NOAA) [34]. There are multiple sources feeding into Sherlock, such as flight plan and track from ARTCC, Rapid Refresh (RR) Weather Forecast from NOAA including wind, temperature and pressure, current and forecast precipitation and echo tops from Convective Integrated Weather Service (CIWS) and FAA System Wide Information Management (SWIM) data sources for flight data. SDW will parse, process the raw data from the source and store them into a data repository. The approved user is able to query and download the integrated data by specifying a date. Here we only use the data from two sources. The Integrated Flight Format (IFF) flight data from SWIM and Echo Top (ET) convective weather data from CIWS. The processing procedure of weather data stays the same as our previous work. However, the parser for flight data has some minor differences.

1. IFF Flight Data

The flight data is stored as Integrated Flight Format (IFF). It includes all of the source raw data plus the derived fields such as flight summary, track points, and flight plan. The flight summary is a general description of the flight which contains flight time, flight call sign, aircraft type, origin, and destination information. The flight track points are the record of real flight operation. It includes the ground measured aircraft position in both the spatial and temporal domains. The flight plan comes as a string of waypoints. It can be translated into WGS84 coordinates using the web-based database. We download the IFF data for ZTL on June 24th, 2019 and process the flight plan and track points to make them equal length. We first take out the flight tracks from the sector IFF file, then do linear interpolation to each of the track points with a 1-second interval. Then match the parsed string format of flight plan with the interpolated flight tracks and remove the flight waypoint that stays outside the sector track ranges. We use Euclidean distance to evaluate whether waypoints belong to the current sector range. Then do the same interpolation to the flight plan to make it has the equal length as the track points. Lastly, we equally sample data points from these two sequences as the processed flight tracks and processed track points used for our model. The length of the processed data is set to be 50 but can be changed to any other value. We've found 6255 sequences of flight records from one day's IFF file and plot it here as shown in Fig. 1. The plots also verify the location of the tracks lies in the range of ZTL. Only longitude and

*Wind and Tornado reported based on National Weather Service Storms Prediction Center `https://www.spc.noaa.gov/exper/archive/event.php?date=20190624` latitude coordinates are considered in this experiment thus the dimension of the final processed flight tracks and flight plans are $6255 \times 50 \times 2$.

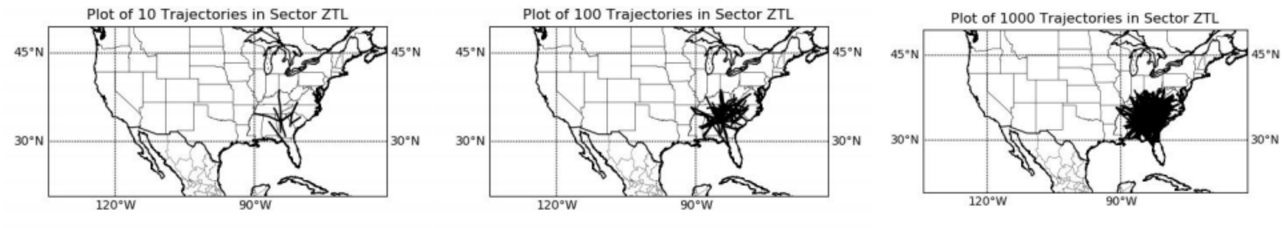


Fig. 1 Flights in ZTL on June 24th, 2019

2. Weather Data

The weather data are obtained from Corridor Integrated Weather Systems (CIWS) in SDW. CIWS acquires data from terminal weather sensing systems, and National Weather Service sensors and forecast products, and automatically generate convective weather products for display on existing systems in both terminal and en route airspace within the CIWS domain. The two key features of CIWS, Echo Top (ET) and Vertically Integrated Liquid (VIL), both include the current and forecast dataset in SDW. In this research, we only use the current ET data but it can be applied to the forecast dataset without too much effort. At each given track points, we take out a weather cube ahead of the current location from the original weather file and rotate it with the heading angle of the aircraft. This is the same weather cube generator used in our previous work. Different from the previous setting, we change the cube size to be 32×32 after scaling the original resolution of weather files by 10 times. The width of the weather band is able to cover sufficient area under this setting. Thus the final dimension of the weather feature cubes is $6255 \times 49 \times 32 \times 1$.

Thus the final processed data[†] includes three parts, the flight plan tensor p as the initial tracks to be calibrated, the weather tensor w as the cause of uncertainties to the trajectory and the true trajectory as the ground truth of the prediction to be a catch with. Each of the pairs of these three is considered as one data point. The total data point we have in our work is 6255. The dimension of flight plan and true trajectory is 50×2 and the dimension for the weather tensor is $49 \times 32 \times 1$.

B. Network Architecture

We use the same model layer setup as our previous work on dropout approximate variational inference. The network model includes two three dimensional convolutional layers, one RNN cell and a few fully-connect (dense) layers. The convolutional layers are used to extract useful features from w into a vector. The extracted weather vector will be concatenated with the flight plan tensor p as the inputs to the RNN cell. The prediction is just the output from multiple dense layers. This framework is also shown in Table. 1. In this table, N represents the batch size and empty cell means not applicable. The second figure in the tensor dimensions (49 in this case) is the length of the sequence (tracks). We use the first point of the sequence as the initial thus only the last 49 points are used in the model.

[†]The data parsers can be found at `https://github.com/ypang6/Weather-Avoidance/tree/master/Trajectory_Prediction`

Table 1 Network Architecture

Layers	Input Size	Output Size	Dimension	Others
Conv3d_1	[N, 49, 32, 32, 1]	[N, 49, 10, 10, 3]	[1, 5, 5, 3]	Flipout, Strides: [1, 3, 3, 1], No Padding
Conv3d_2	[N, 49, 10, 10, 3]	[N, 49, 4, 4, 1]	[1, 3, 3, 1]	Flipout, Strides: [1, 3, 3, 1], Zero Padding
Flatten	[N, 49, 4, 4, 1]	[N, 49, 16]		
Dense_1	[N, 49, 16]	[N, 49, 4]	4	Flipout
Dense_2	[N, 49, 4]	[N, 49, 1]	1	Flipout
Concat	[N, 49, 1]	[N, 49, 3]		Concatenate with Flight Plan p
LSTM	[N, 49, 3]	[N, 49, 128]	128	Flipout
Dense_3	[N, 49, 128]	[N, 49, 64]	64	Flipout
Dense_4	[N, 49, 64]	[N, 49, 32]	32	Flipout
Dense_5	[N, 49, 32]	[N, 49, 2]	2	Flipout

The initial input to the model is the weather cube tensor w with dimension $[N, 49, 32, 32, 1]$. The Conv3d_1 layer has a kernel size of $[1, 5, 5, 3]$ and a stride of $[1, 3, 3, 1]$ along the last four dimensions of the input tensor with no padding added for convolution operation. The output has a dimension of $[N, 49, 10, 10, 3]$. The second convolutional layer Conv3d_2 has a similar setup as the first one with a kernel size of $[1, 3, 3, 1]$ and a stride of $[1, 3, 3, 1]$ and zero padding, respectively. The output has a size of $[N, 49, 16]$ after flatten along the last three dimensions of the tensor. The following two dense layers would compress the tensor to a dimension of $[N, 49, 1]$, which is the dimension of the weather feature tensor we extracted. We choose to concatenate it with the flight plan tensor p as the input to the RNN cell. The dimension of LSTM hidden state h_t and cell state c_t tensor is 128. Zero initialization is used in the LSTM cell. The output of the LSTM cell will be fed into three dense layers to calculate the output of the model. The training objective is defined based on Eq. (6) which incorporates two parts, the KL divergence, and the negative log-likelihood. We use deep probabilistic programming packages Edward[‡] [35–37] and Tensorflow-probability[§] [38] based on Tensorflow 2.1.0 for our implementation. Fig. 2 visualize the loss on both training and testing dataset during the training process. However, only training dataset is used in the optimization steps.

Table 2 Loss Values

Epoch	-ELBO Train	-ELBO Test	KL Train	KL Test	NLL Train	NLL Test
500	0.071971	0.071734	0.022909	0.022884	0.049062	0.048850
1000	0.074546	0.072054	0.022662	0.022651	0.051884	0.049403
1500	0.074990	0.071746	0.022771	0.022755	0.052220	0.048991

VI. Results

The model experiment results are evaluated from the test dataset leave alone during the training of the model. The optimization objective is to minimize the negative ELBO defined in Eq. (6). After training for 1,500 epochs with RMSProp optimizer, the value reaches acceptable digits and stabilized.

$$\begin{aligned} p(y^*|x^*, X, Y) &= \mathbb{E}_{q(\omega)} p(y^*|x^*, \omega) \\ &\approx \frac{1}{K} \sum_{k=1}^{K} p(y^*|x^*, \hat{\omega_k}) \end{aligned} \tag{13}$$

We perform $K = 100$ tests to the forward pass of the network as Eq. (13). The network would sample weights from the weights distributions during each test then averaging the result for the mean prediction Y^{pred} and variance. Four randomly selected testing results are visualized in Fig. 3 with 95% confidence interval. At each position, the aircraft has

[‡]`https://github.com/google/edward2`
[§]`https://github.com/tensorflow/probability`

Fig. 2 Training Process: -ELBO, KL, NLL

uncertainty on both latitude and longitude direction. We draw the simplified ellipses as the contour or bivariate normal distributions. We found that mean prediction can reduce the deviation from the true trajectory compared to the flight plan. The variance increased when deviation happens and decrease when deviation reduced. Visualization of the test dataset is not sufficient for the evaluation of model performance. We define the following quantities to numerically calculate the variance reduction and prediction power. Eq. (14) is the euclidean distance between the true trajectory and the flight plan while Eq. (15) measures the distance between our predicted trajectory and the true trajectory. Eq. (16) simply calculates the percentage of reduction within these two quantities. This is the same evaluation metrics as the previous work.

$$L2_k^{ori} = \sum_i^n \sum_j^d (Y_{k,i,j}^{true} - Y_{k,i,j}^{p})^2 \tag{14}$$

$$L2_k^{new} = \sum_i^n \sum_j^d (Y_{k,i,j}^{true} - Y_{k,i,j}^{pred})^2 \tag{15}$$

$$reduction = \frac{Var(L2_k^{ori}) - Var(L2_k^{new})}{Var(L2_k^{ori})} \tag{16}$$

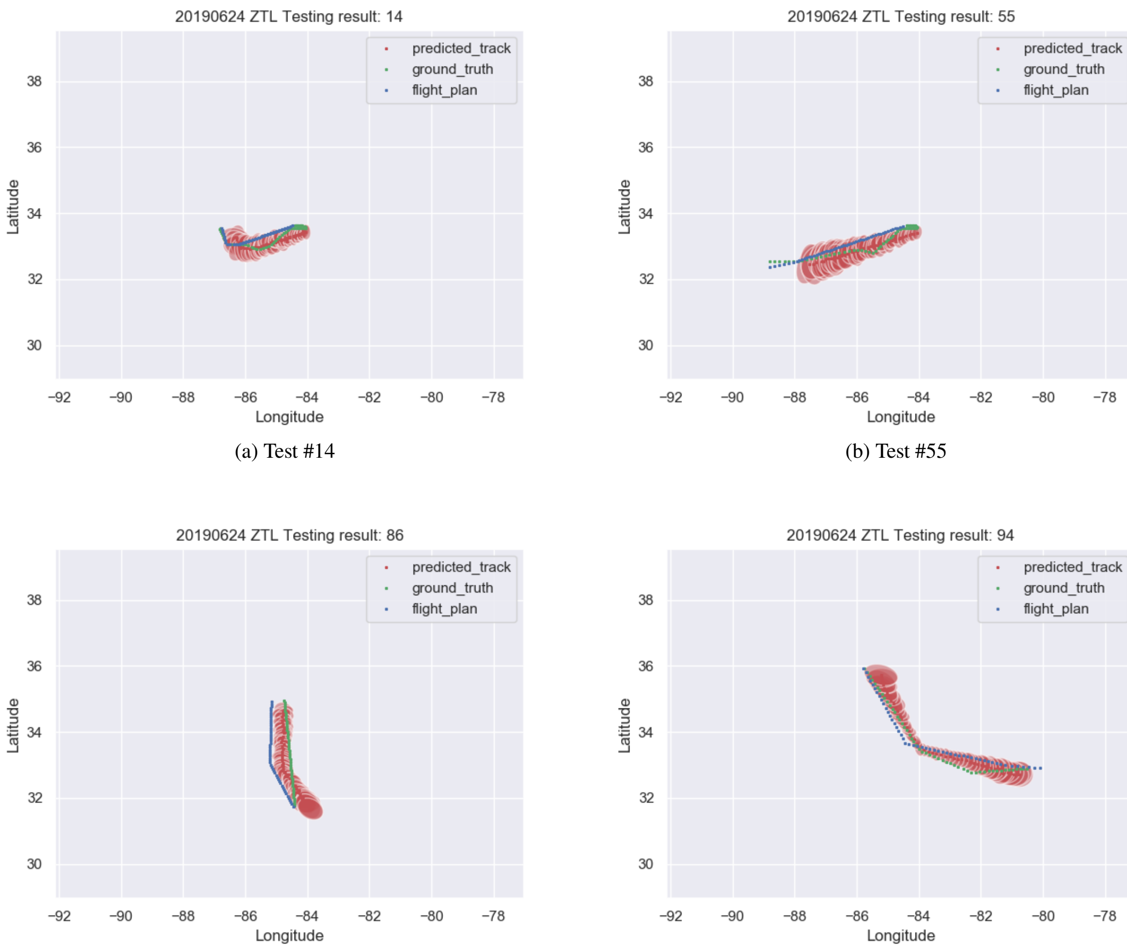
We compared this method with other probabilistic based TP models and listed in Table. 3. The overall variance reduction is bigger than the generative model and the dropout approximate. However, fewer flights deviation can be reduced. The VI approach has better variance reduction compared to the dropout-base method. This also match the result in the literature [32].

Table 3 Comparison on Different Methods

Models	Percentage of Flights Reduced	Overall Variance Reduction
Generative Model	55.2%	22.1%
Dropout as Bayesian	26.2%	16.8%
Variational Inference	19.4%	40.8%

VII. Conclusion

A major key requirement of NextGen pursuing accurate and reliable aircraft trajectory prediction. To accommodate this, we purpose a trajectory prediction neural network model with the recent advances in Bayesian Deep Learning



(a) Test #14

(b) Test #55

(c) Test #86

(d) Test #94

Fig. 3 Testing Results.

research as approximate Bayesian variational inference for NNs. This approach follows the idea of *Bayes By Backprop* and the state-of-art weight perturbation method, Flipout, to perform backpropagation on distributions. Here are the three major benefits compare to other competitors,

- The experimental results on predicting the flight tracks in one control sector achieve variance reduction on flight deviations. The model outperforms the dropout-based method in the sense of variance reduction, which is also stated in the literature [32].
- The prior distributions on neural network weights can be adjusted to any user specifications. In this work, we assume the weight priors follow standard normal distributions.
- The model can perform reliable uncertainty estimates. The mean value, as well as the variance of the model predictions, is optimized in the objective function.

Acknowledgments

The work was performed at the Prognostic Analysis and Reliability Assessment Lab at Arizona State University. The research reported in this paper was supported by funds from NASA University Leadership Initiative program (Contract No. NNX17AJ86A, Project Officer: Dr. Anupa Bajwa, Principal Investigator: Dr. Yongming Liu). The support is gratefully acknowledged.

References

[1] FAA, M., "NextGen implementation plan," *Washington, DC, Available: http://www. faa. gov/nextgen/media/ng2011implement ationplan. pdf (accessed March 18, 2011)*, 2013.

[2] Schuster, W., and Ochieng, W., "Performance requirements of future Trajectory Prediction and Conflict Detection and Resolution tools within SESAR and NextGen: Framework for the derivation and discussion," *Journal of Air Transport Management*, Vol. 35, 2014, pp. 92–101.

[3] Mondoloni, S., Swierstra, S., and Paglione, M., "Assessing trajectory prediction performance-metrics definition," *24th Digital Avionics Systems Conference*, Vol. 1, IEEE, 2005, pp. 3–C.

[4] Gal, Y., and Ghahramani, Z., "A theoretically grounded application of dropout in recurrent neural networks," *Advances in neural information processing systems*, 2016, pp. 1019–1027.

[5] Lauret, P., Fock, E., Randrianarivony, R. N., and Manicom-Ramsamy, J.-F., "Bayesian neural network approach to short time load forecasting," *Energy conversion and management*, Vol. 49, No. 5, 2008, pp. 1156–1166.

[6] Graves, A., "Practical variational inference for neural networks," *Advances in neural information processing systems*, 2011, pp. 2348–2356.

[7] Back, A., and Keith, W., "Bayesian Neural Networks for Financial Asset Forecasting," , 2019.

[8] Liu, Y., and Goebel, K., "Information Fusion for National Airspace System Prognostics," *PHM Society Conference*, Vol. 10, 2018.

[9] Hu, J., "Probabilistic Risk Assessment and Mitigation for UAS Safety and Traffic Management," *Proceedings of the Annual Conference of the PHM Society*, Vol. 11, 2019.

[10] Musialek, B., Munafo, C. F., Ryan, H., and Paglione, M., "Literature survey of trajectory predictor technology," *Federal Aviation Administration, William J. Hughes Technical Center, Tech. Rep*, 2010.

[11] Ayhan, S., and Samet, H., "Aircraft trajectory prediction made easy with predictive analytics," *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, 2016, pp. 21–30.

[12] De Leege, A., van Paassen, M., and Mulder, M., "A machine learning approach to trajectory prediction," *AIAA Guidance, Navigation, and Control (GNC) Conference*, 2013, p. 4782.

[13] Wiest, J., Höffken, M., Kreßel, U., and Dietmayer, K., "Probabilistic trajectory prediction with Gaussian mixture models," *2012 IEEE Intelligent Vehicles Symposium*, IEEE, 2012, pp. 141–146.

[14] Lymperopoulos, I., Lygeros, J., and Lecchini, A., "Model based aircraft trajectory prediction during takeoff," *AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2006, p. 6098.

[15] Pang, Y., Xu, N., and Liu, Y., "Aircraft Trajectory Prediction using LSTM Neural Network with Embedded Convolutional Layer," *Proceedings of the Annual Conference of the PHM Society*, Vol. 11, 2019.

[16] Pang, Y., Yao, H., Hu, J., and Liu, Y., "A Recurrent Neural Network Approach for Aircraft Trajectory Prediction with Weather Features From Sherlock," *AIAA Aviation 2019 Forum*, 2019, p. 3413.

[17] Wang, Y., Pang, Y., Liu, Y., Dutta, P., and Yang, B.-J., "Aircraft Trajectory Prediction and Risk Assessment Using Bayesian Updating," *AIAA Aviation 2019 Forum*, 2019, p. 2936.

[18] Yu, Y., Yao, H., and Liu, Y., "Aircraft dynamics simulation using a novel physics-based learning method," *Aerospace Science and Technology*, Vol. 87, 2019, pp. 254–264.

[19] Lygeros, J., and Prandini, M., "Aircraft and weather models for probabilistic collision avoidance in air traffic control," *Proceedings of the 41st IEEE Conference on Decision and Control, 2002.*, Vol. 3, IEEE, 2002, pp. 2427–2432.

[20] Prandini, M., Hu, J., Lygeros, J., and Sastry, S., "A probabilistic approach to aircraft conflict detection," *IEEE Transactions on intelligent transportation systems*, Vol. 1, No. 4, 2000, pp. 199–220.

[21] Yepes, J. L., Hwang, I., and Rotea, M., "New algorithms for aircraft intent inference and trajectory prediction," *Journal of guidance, control, and dynamics*, Vol. 30, No. 2, 2007, pp. 370–382.

[22] Liu, W., and Hwang, I., "Probabilistic trajectory prediction and conflict detection for air traffic control," *Journal of Guidance, Control, and Dynamics*, Vol. 34, No. 6, 2011, pp. 1779–1789.

[23] Liu, Y., and Hansen, M., "Predicting aircraft trajectories: a deep generative convolutional recurrent neural networks approach," *arXiv preprint arXiv:1812.11670*, 2018.

[24] Pang, Y., and Liu, Y., "Conditional Generative Adversarial Networks (CGAN) for Aircraft Trajectory Prediction considering weather effects," *AIAA Scitech 2020 Forum*, 2020, p. 1853.

[25] Pang, Y., and Liu, Y., "Probabilistic Aircraft Trajectory Prediction Considering Weather Uncertainties Using Dropout As Bayesian Approximate Variational Inference," *AIAA Scitech 2020 Forum*, 2020, p. 1413.

[26] Jackson, M. R., Zhao, Y. J., and Slattery, R. A., "Sensitivity of trajectory prediction in air traffic management," *Journal of Guidance, Control, and Dynamics*, Vol. 22, No. 2, 1999, pp. 219–228.

[27] Blundell, C., Cornebise, J., Kavukcuoglu, K., and Wierstra, D., "Weight uncertainty in neural networks," *arXiv preprint arXiv:1505.05424*, 2015.

[28] Gal, Y., "Uncertainty in deep learning," Ph.D. thesis, PhD thesis, University of Cambridge, 2016.

[29] Kullback, S., and Leibler, R. A., "On information and sufficiency," *The annals of mathematical statistics*, Vol. 22, No. 1, 1951, pp. 79–86.

[30] Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., and Saul, L. K., "An introduction to variational methods for graphical models," *Machine learning*, Vol. 37, No. 2, 1999, pp. 183–233.

[31] Gal, Y., and Ghahramani, Z., "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," *international conference on machine learning*, 2016, pp. 1050–1059.

[32] Wen, Y., Vicol, P., Ba, J., Tran, D., and Grosse, R., "Flipout: Efficient pseudo-independent weight perturbations on mini-batches," *arXiv preprint arXiv:1803.04386*, 2018.

[33] Arneson, H. M., "Sherlock Data Warehouse," 2018.

[34] Eshow, M. M., Lui, M., and Ranjan, S., "Architecture and capabilities of a data warehouse for ATM research," *2014 IEEE/AIAA 33rd Digital Avionics Systems Conference (DASC)*, IEEE, 2014, pp. 1E3–1.

[35] Tran, D., Dusenberry, M., van der Wilk, M., and Hafner, D., "Bayesian layers: A module for neural network uncertainty," *Advances in Neural Information Processing Systems*, 2019, pp. 14633–14645.

[36] Tran, D., Kucukelbir, A., Dieng, A. B., Rudolph, M., Liang, D., and Blei, D. M., "Edward: A library for probabilistic modeling, inference, and criticism," *arXiv preprint arXiv:1610.09787*, 2016.

[37] Tran, D., Hoffman, M. W., Moore, D., Suter, C., Vasudevan, S., and Radul, A., "Simple, distributed, and accelerated probabilistic programming," *Advances in Neural Information Processing Systems*, 2018, pp. 7598–7609.

[38] Dillon, J. V., Langmore, I., Tran, D., Brevdo, E., Vasudevan, S., Moore, D., Patton, B., Alemi, A., Hoffman, M., and Saurous, R. A., "Tensorflow distributions," *arXiv preprint arXiv:1711.10604*, 2017.