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

Yutian Pang, Yuhao Wang and Yongming Liu
Arizona State University, Tempe, Arizona, 85287, USA

AIAA Aviation Forum 2020

Outline

- Introduction
  - Background
  - Previous Work
  - Feedbacks and Solutions
- Preliminaries
  - Bayesian Neural Network
- Experiments
  - Data
  - Model
  - Results
- Conclusions
- Discussions

Introduction - Motivations

- Weather related concerns lead to huge uncertainty during daily aviation operations.
  - The convective weather conditions can develop rapidly [1].
  - 30% ~ 40% of the delayed flights are caused by weather related reasons [2].
- The workload of controllers can increase significantly during certain extreme climatic events.
  - The development of trajectory planning tools help reduce these concerns.

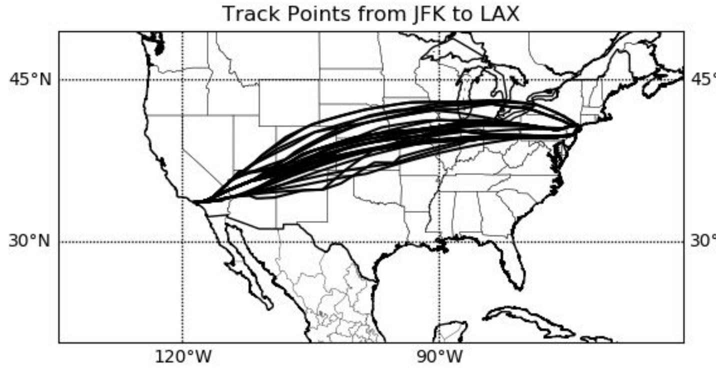


Fig.1 Flight Tracks from JFK to LAX on Dec 19, 2018

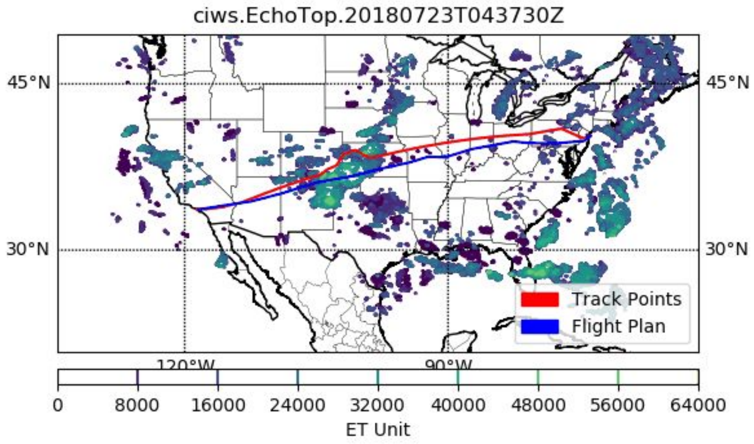


Fig.2 Convective Weather Conditions during the Flight Period from 04:45:00 to 11:47:30 on July 23, 2018

[1] Erzberger, H., Lauderdale, T., and Chu, Y., “Automated conflict resolution, arrival management, and weather avoidance for air traffic management,” Journal of aerospace engineering, Vol.226, No. 8, 2012, pp. 930–949.

[2] Belcastro, L., Marozzo, F., Talia, D., & Trunfio, P. (2016). Using scalable data mining for predicting flight delays. ACM Transactions on Intelligent Systems and Technology (TIST), 8(1), 5.

Review

- Researchers are focusing on,
  - short-term: dynamic weather reroutes
  - long-term: strategic trajectory prediction

in both deterministic and probabilistic sense.

Previous Work - Aviation 2019

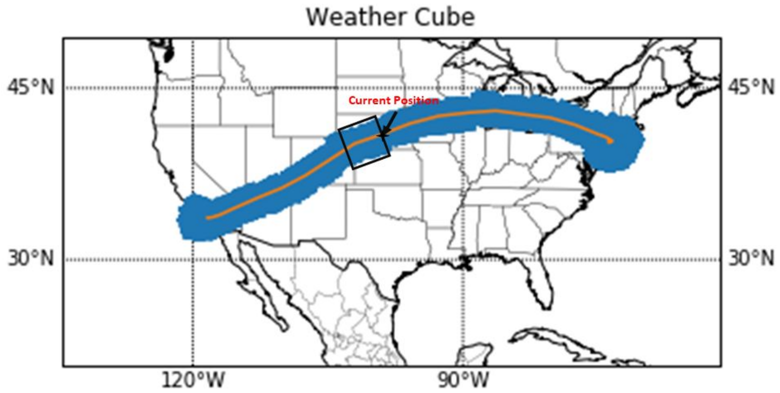


Fig.3 Weather cube generating algorithm

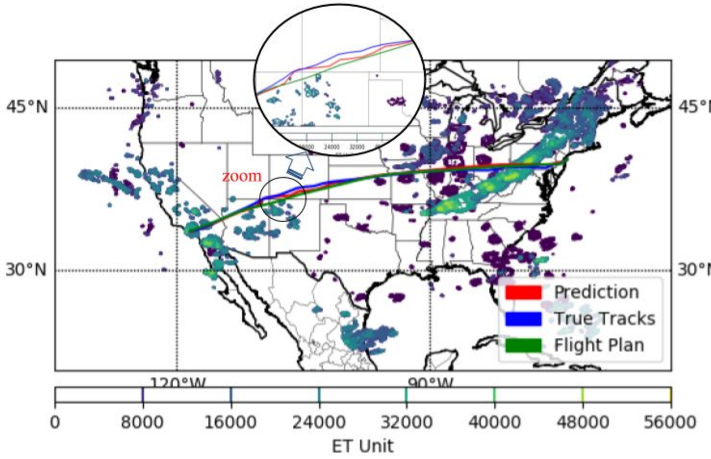


Fig.4 A prediction result

- Previous work [3] use recurrent neural network to predict the flight trajectory from JFK to LAX.
  - Extract weather data along each track point.
  - Modify the recurrence of RNN cell to incorporate weather features into the model.
- Issues,
  - A long prediction (6 hours flight) is untrustworthy.
  - Needs better prediction accuracy with prediction uncertainties.
  - Limited data.

[3] Pang, Y., Yao, H., Hu, J., & Liu, Y. (2019). A Recurrent Neural Network Approach for Aircraft Trajectory Prediction with Weather Features From Sherlock. In AIAA Aviation 2019 Forum (p. 3413).

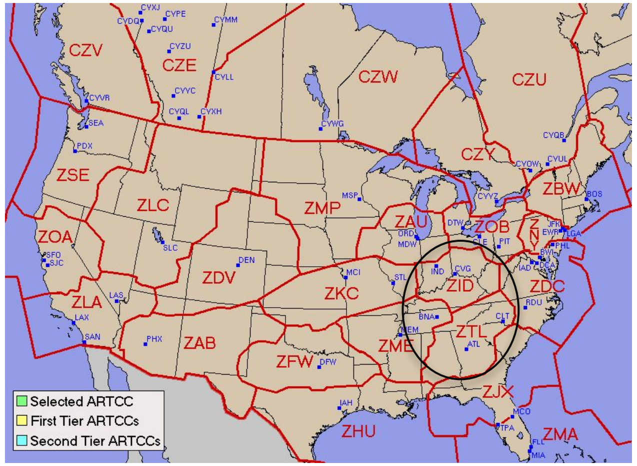
Feedbacks from Aviation 2019

Table 1. Major feedbacks from Aviation 2019

01	6 hours prediction is untrustworthy	• Focus on predicting trajectories within each TRACON
02	Probabilistic prediction instead of point estimate	• Introduce Bayesian framework to our model
03	Limited data	• Semi-supervised learning

TRACON ATM

- Start from data.
  - It's reported to have severe weather condition on 06/24/2019* within the United States airspace.
  - Tornado and high wind are reported at numerous locations in the east.
- Sector ZID (Indianapolis) and ZTL (Atlanta) roughly cover the area of interest.



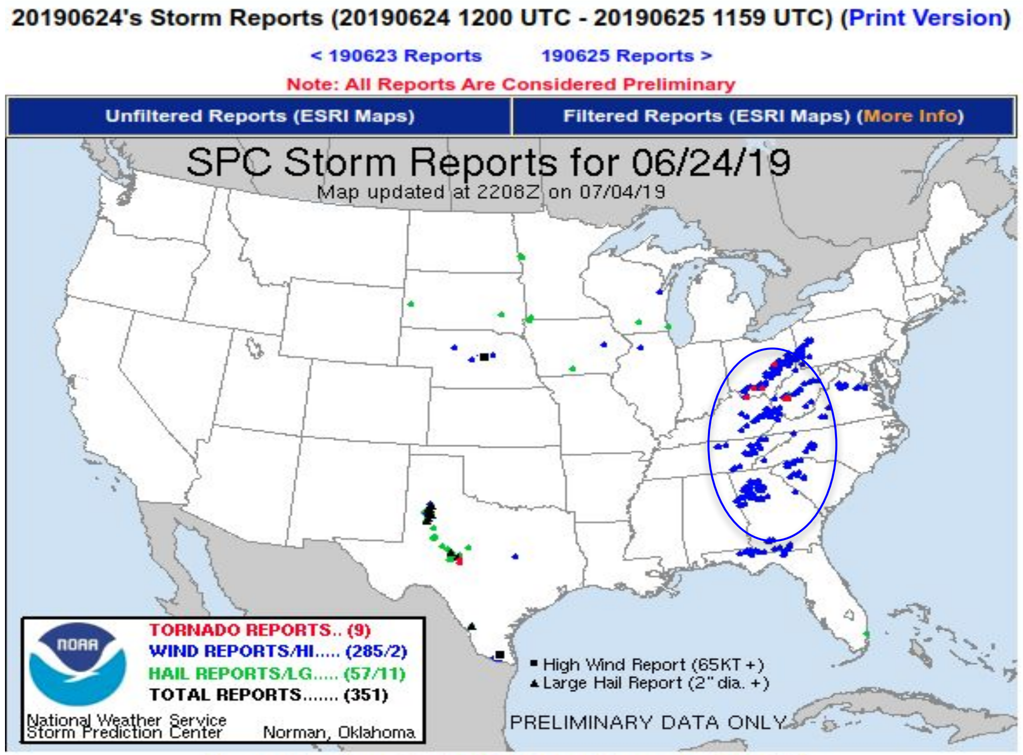


Fig.5 SPC Storm Reports for 06/24/19

*NOAA Storm Prediction Center: *https://www.spc.noaa.gov/exper/archive/events/*

Conditional Generative Adversarial Networks (CGAN) for Aircraft Trajectory Prediction considering Weather Effects [4]

- Conditional GAN (CGAN) is as simple as adding an additional input to the generator and discriminator of classical GAN [5].
- The objective changes to,

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x}|\boldsymbol{y})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z}|\boldsymbol{y})))].$$

- <u>In our model, the condition would be the last on-file flight plan and the weather conditions.</u>
- The model can be formulated as,

$$f(x, z|\{p, w\} \in \mathbf{y})$$

  - where x is the true trajectory, z is random noise as input to G, p is the flight plan tensor and w is the weather tensor.
- z follows truncated normal distribution $N(0.5,\ 1.0)$ with a lower bond 0 and upper bond 1.

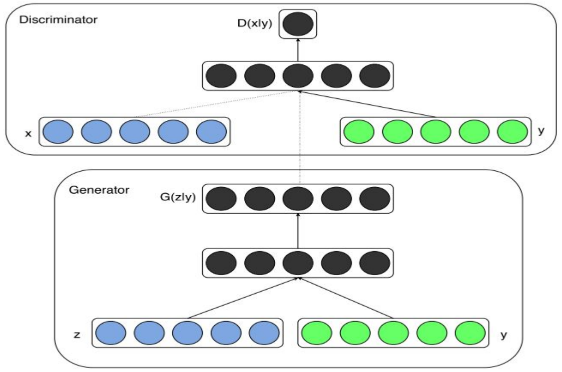


Fig.6 CGAN idea [5]

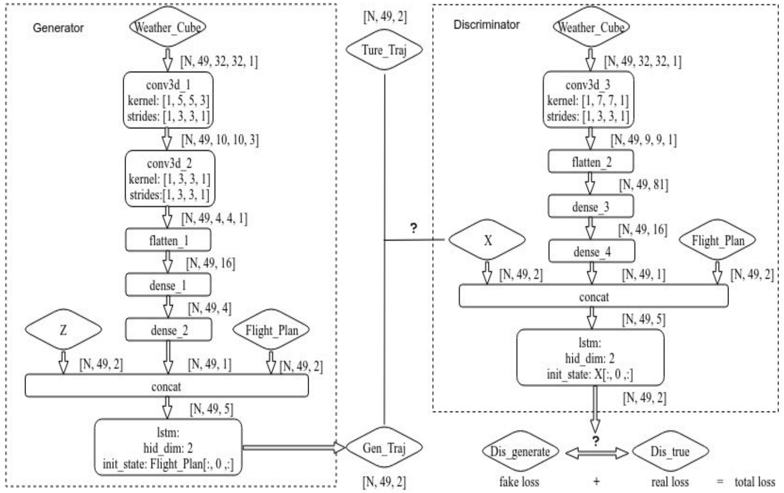


Fig.7 model architecture [4]

[4] Pang, Y., & Liu, Y. (2020). Conditional Generative Adversarial Networks (CGAN) for Aircraft Trajectory Prediction considering weather effects. In AIAA Scitech 2020 Forum (p. 1853).
[5] Mirza, M., & Osindero, S. (2014). Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784.

Probabilistic Aircraft Trajectory Prediction Considering Weather Uncertainties Using Dropout As Bayesian Approximate Variational Inference [6]

- Dropout is proved to be an Bernoulli approximate to variational inference for Neural Networks [7, 8, 9].
- The implementation of the Bayesian CNN is as simple as using dropout after every convolution layer before pooling , and after every fully-connected layers [7, 8].
- The dropout approximation to recurrent cells requires special calculations to dropout the connection between each recurrence. This is called variational dropout to recurrent neural networks [9].
- The final dropout approximation of BNN model is as simple as applying (variational) dropout during the test stage.
- Uncertainty is estimated by,

$$p(y^*|x^*, X, Y) \approx \int p(y^*|x^*, \omega) q(\omega) d\omega \approx \frac{1}{K}\sum_{k=1}^{K} p(y^*|x^*, \hat{\omega}_k)$$

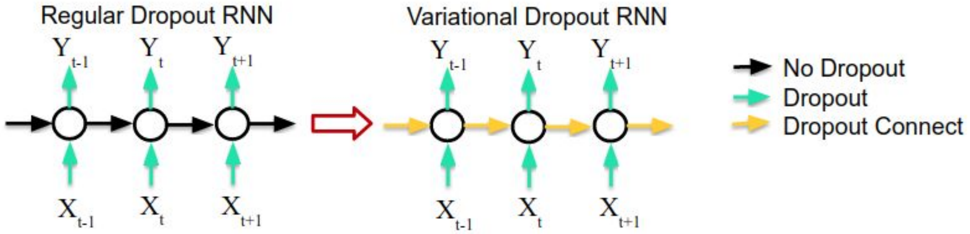


Fig.8 Variational Dropout

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

Fig.9 Model Setup [6]

[6] Pang, Y., & Liu, Y. (2020). Probabilistic Aircraft Trajectory Prediction Considering Weather Uncertainties Using Dropout As Bayesian Approximate Variational Inference. In AIAA Scitech 2020 Forum (p. 1413).
[7] Gal, Y., "Uncertainty in deep learning," Ph.D. thesis, PhD thesis, University of Cambridge, 2016.
[8] Gal, Y., and Ghahramani, Z., "Bayesian convolutional neural networks with Bernoulli approximate variational inference," arXiv:1506.02158, 2015.
[9] 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.

Summary of Previous Work

01	6 hours prediction is untrustworthy	• Focus on predicting trajectories within each TRACON
02	Probabilistic prediction instead of point estimate	• Dropout approximate to Bayesian Neural Network
03	Limited data	• Semi-supervised learning ○ Generative Adversarial Nets

- Issues,
  - The randomness of CGAN solely from the noisy inputs.
  - The dropout ratio controls the uncertainty of the outputs.
- How to get reliable uncertainty estimation of BNN?

Preliminaries - Bayesian Neural Network

- Bayesian theorem,

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

- Bayesian inference,

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

- Approximating variational posterior,

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

- Minimizing KL divergence is equivalent to maximizing the evidence lower bound (ELBO) w.r.t. $q_\theta(\omega)$

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

Notes,
X – training inputs
Y – training labels
x* – testing inputs
y* – testing labels
ω – model parameters

Bayesian Neural Network - Weight Perturbations

- *Bayes By Backprop* [10] purposed an algorithm to do backpropagation for distributions using the local reparameterization trick.
- *Bayes By Backprop* assumes the approximate posterior follow diagonal Gaussian distributions instead of Bernoulli distributions in dropout approximation.
- The variant of weight perturbation algorithms suffer from high variance of the gradient estimates because all training examples in a mini-batch share the same perturbation.
- The recently developed weight perturbation method, flipout [11], overcomes this problem by perturbing the weights quasi-independently within a mini-batch.

1. Sample $\epsilon \sim \mathcal{N}(0, I)$.
2. Let $\boxed{\mathbf{w} = \mu + \log(1 + \exp(\rho)) \circ \epsilon.}$ reparameterization
3. Let $\boxed{\theta = (\mu, \rho).}$ new variational posterior parameters
4. Let $f(\mathbf{w}, \theta) = \log q(\mathbf{w}|\theta) - \log P(\mathbf{w})P(\mathcal{D}|\mathbf{w})$.
5. Calculate the gradient with respect to the mean

$$\Delta_\mu = \frac{\partial f(\mathbf{w}, \theta)}{\partial \mathbf{w}} + \frac{\partial f(\mathbf{w}, \theta)}{\partial \mu}. \tag{3}$$

6. Calculate the gradient with respect to the standard deviation parameter ρ

$$\Delta_\rho = \frac{\partial f(\mathbf{w}, \theta)}{\partial \mathbf{w}} \frac{\epsilon}{1 + \exp(-\rho)} + \frac{\partial f(\mathbf{w}, \theta)}{\partial \rho}. \tag{4}$$

7. Update the variational parameters:

$$\mu \leftarrow \mu - \alpha \Delta_\mu \tag{5}$$
$$\rho \leftarrow \rho - \alpha \Delta_\rho. \tag{6}$$

Fig.10 Local reparameterization trick[10]

[10] Blundell, C., Cornebise, J., Kavukcuoglu, K., and Wierstra, D., “Weight uncertainty in neural networks,”arXiv preprintarXiv:1505.05424, 2015.
[11] Wen, Y., Vicol, P., Ba, J., Tran, D., & Grosse, R. (2018). Flipout: Efficient pseudo-independent weight perturbations on mini-batches. arXiv preprint arXiv:1803.04386. Published as a conference paper at ICLR 2018.

Experiments

- The experiments is conducted using the recent advances in Bayesian deep learning.
- The implementation is performed with the deep probabilistic programming package Edward[12, 13, 14] and Tensorflow-distributions [15].

[12] 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.

[13] 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.

[14] 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.

[15] 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.

Data

- Sherlock Data Warehouse (SDW)
  - SDW [16] is a big data system for data visualization to support air traffic management (ATM) research, which includes a database, a web-based graphical user interface (GUI) and other services.
  - We use sector Integrate Flight Format (IFF) data for flight plans and real tracks, and Convective Integrated Weather Service (CIWS) for EchoTop (ET) weather data.
  - As described in the previous slides, we choose to use the data in sector Atlanta TRACON on June 24th, 2019.

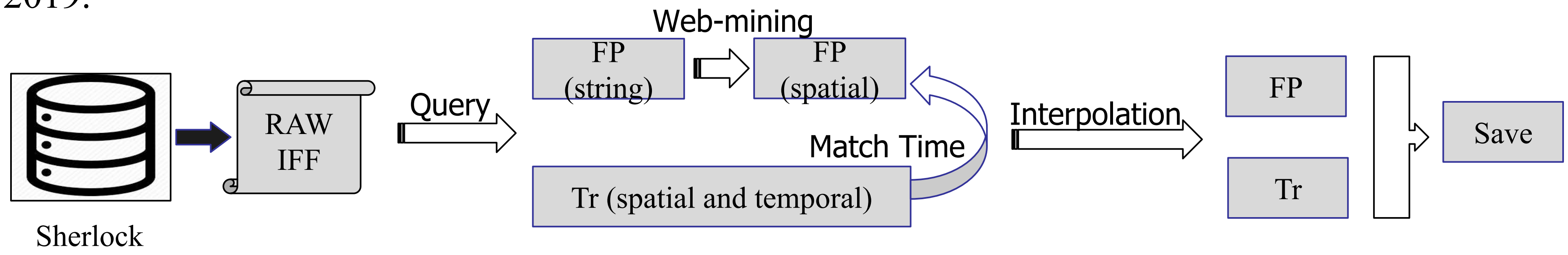


Fig.11 Flowchart: Raw IFF data processing

[16] Arneson, H. M., Hegde, P., La Scola, M. E., Evans, A. D., Keller, R. M., & Schade, J. E. (2019). Sherlock Data Warehouse.

Data

- ❖ The dimension of the processed flight data is 6255×50×2.

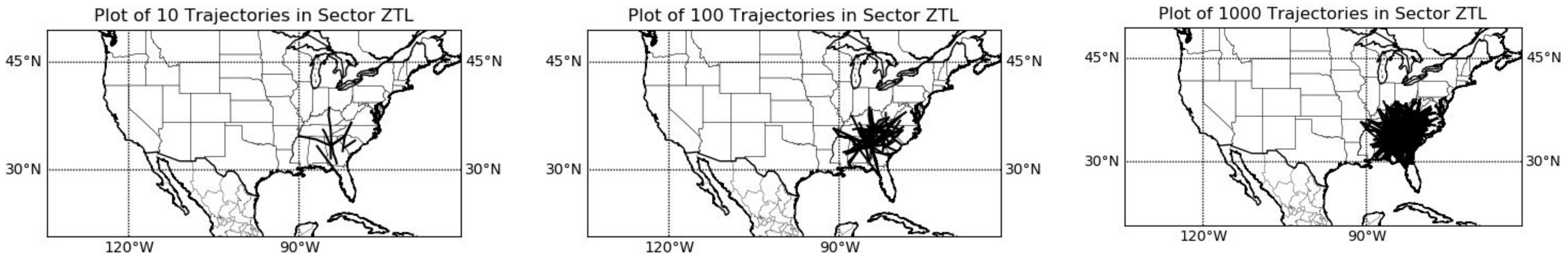


Fig.12 Flights in sector ZTL on 06/24/2019

- ❖ The weather data,
  - ➢ At each track point, we take out the weather tensor of size 32×32×1 except for the starting point. The orientation of the weather tensor is rotated with the aircraft heading angle.
  - ➢ The processed weather tensor has size 6255×49×32×32×1.

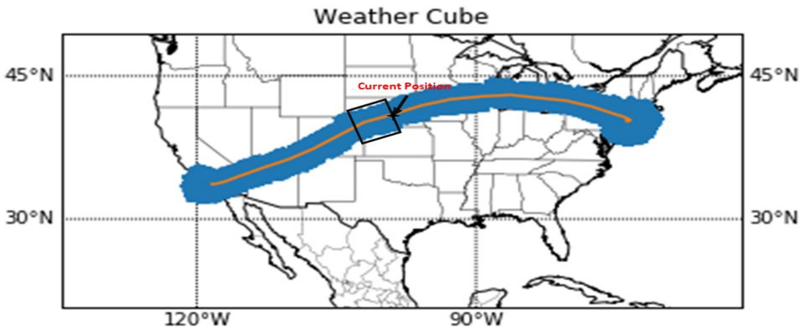


Fig.13 Weather cube generating algorithm

Model Setup

- Same layer setup as the dropout approximation.
- Flipout perturbation to network parameters for convolutional, recurrent and fully-connected layers.

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

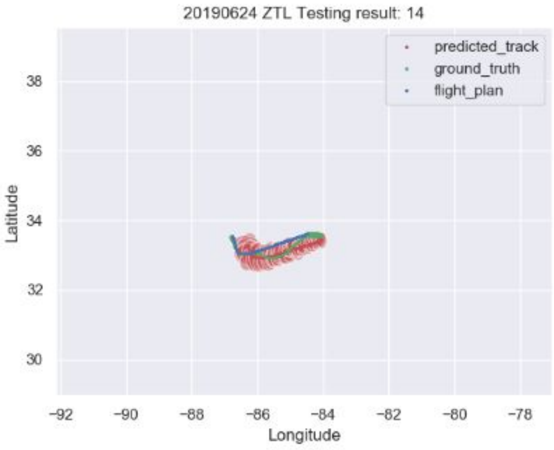
Fig.14 Training Loss

Table 2 Loss Values

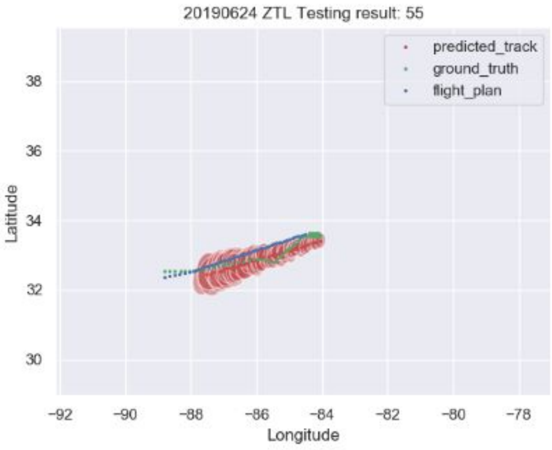
Epoch	-ELBO Train	-ELBO Test	KL Train	KL Test	NLL Train	NLL Test
500	0.071971	0.071734	0.022909	0.022884	0.049062	0.048850
1000	0.074546	0.072054	0.022662	0.022651	0.051884	0.049403
1500	0.074990	0.071746	0.022771	0.022755	0.052220	0.048991

BNN Results

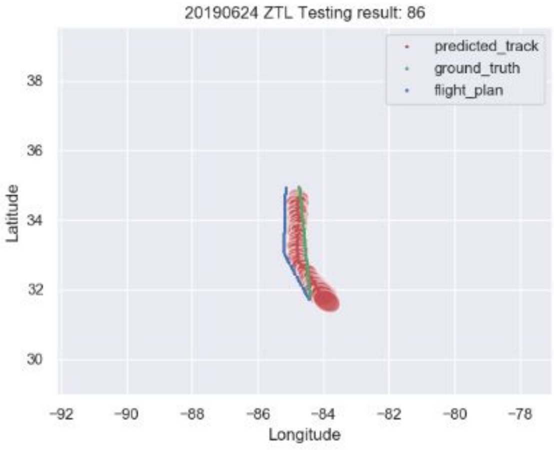
- The model prediction is able to reduce the deviation from the ground truth.
- The uncertainty increased when deviations happened.



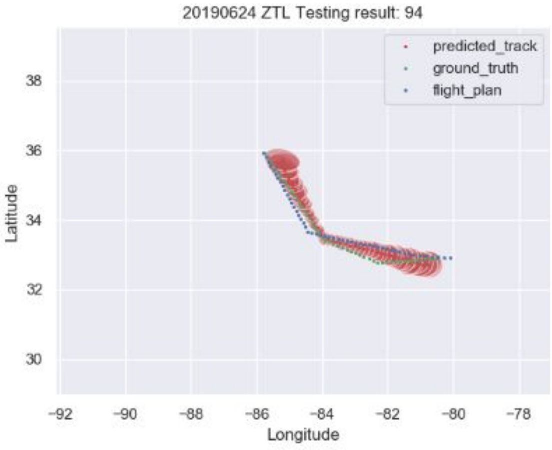
(a) Test #14



(b) Test #55



(c) Test #86



(d) Test #94

Comparison

- We define the following metrics to evaluate the performance of different models.

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

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

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

- The results are shown below,

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%

- The proposed model can achieve larger variance reduction.

Conclusions

- The mean value, as well as the variance, of the prediction is optimized within the objective function of the Bayesian Neural Network.
- The model is able to perform reliable uncertainty estimates.
- The model outperforms the dropout-based method in the sense of variance reduction, which is also stated in the literature.

Acknowledgements

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.

Questions?

Thanks for your time.