Predicting Aircraft Trajectories
– A Deep Generative Neural Networks Approach

Yulin Liu, Mark Hansen

liuyulin101@berkeley.edu

Institute of Transportation Studies

University of California, Berkeley

Nov 7th, 2018

Joint work with Prof. Michael Ball, and Prof. David Lovell

Outline

- Introduction
- Data Sources and Feature Engineering
- Methodology
- Numerical Experiments
- Summaries

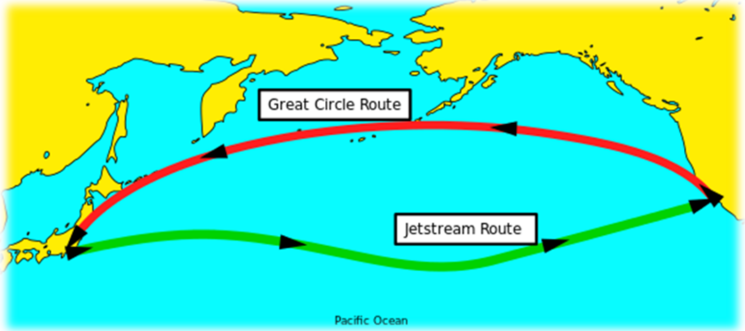
Background (I)

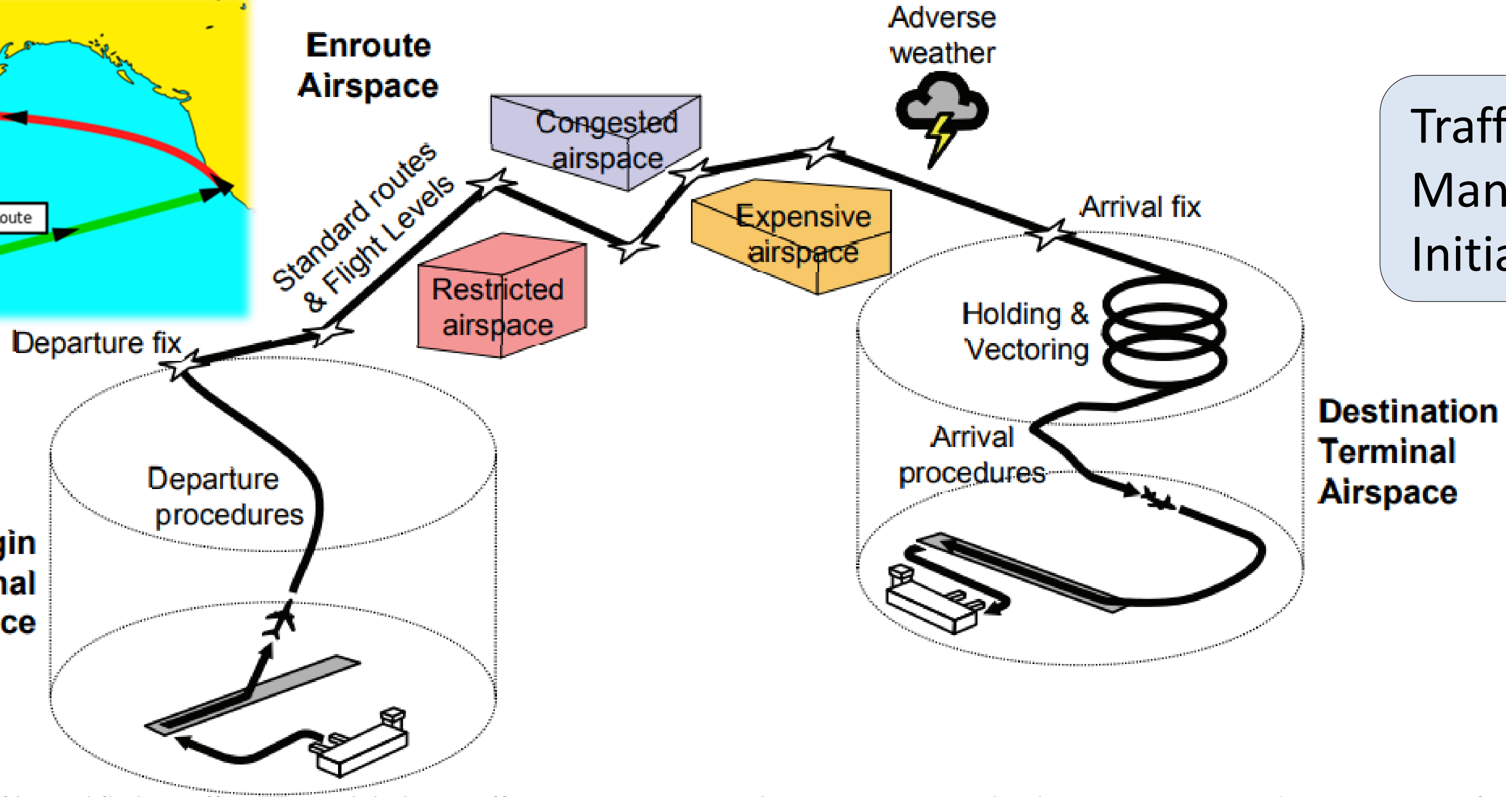
- FAA and EUROCONTROL published metrics to evaluate the flight en route inefficiency, and FAA is seeking to understand the causal factors behind the inefficiency.
- Observe rich variety of route choices that differ drastically with respect to en route inefficiency.
- Models have been proposed to understand the features that affect the choice of route for a flight, and further, the **overall** contributions of different factors to flight en route inefficiencies.

Reference:
Y. Liu, M. Hansen, D. Lovell, C. Chuang, M. Ball and J. Gulding, "Causal Analysis of En Route Flight Inefficiency--the US Experience," in 12th USA/Europe Air Traffic Management Research and Development Seminar (ATM 2017), Seattle, WA, USA, 2017

Background (II)

- Interests towards individual flights have been widely recognized in recent years.
- Trajectory prediction tools fit well into this domain.



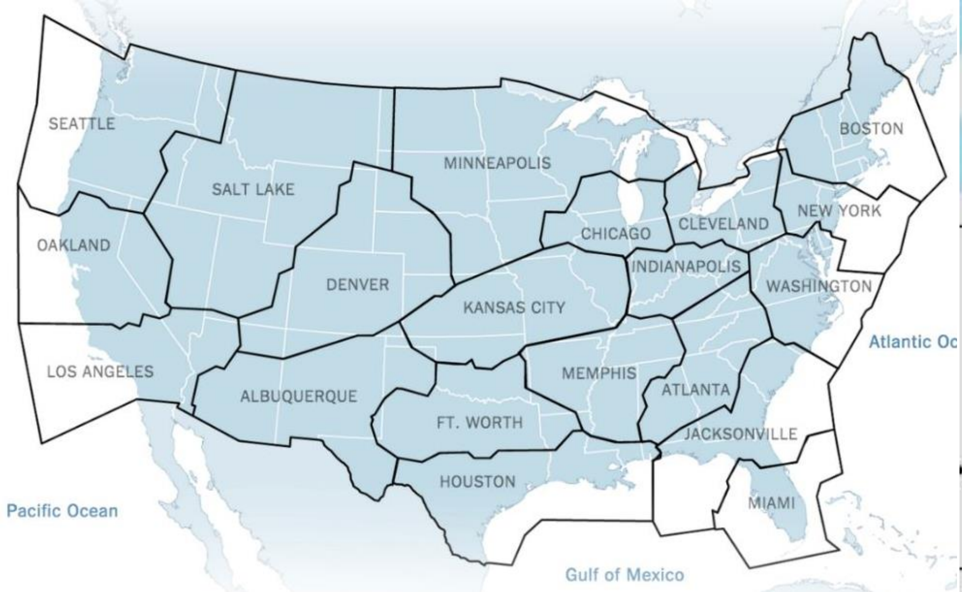


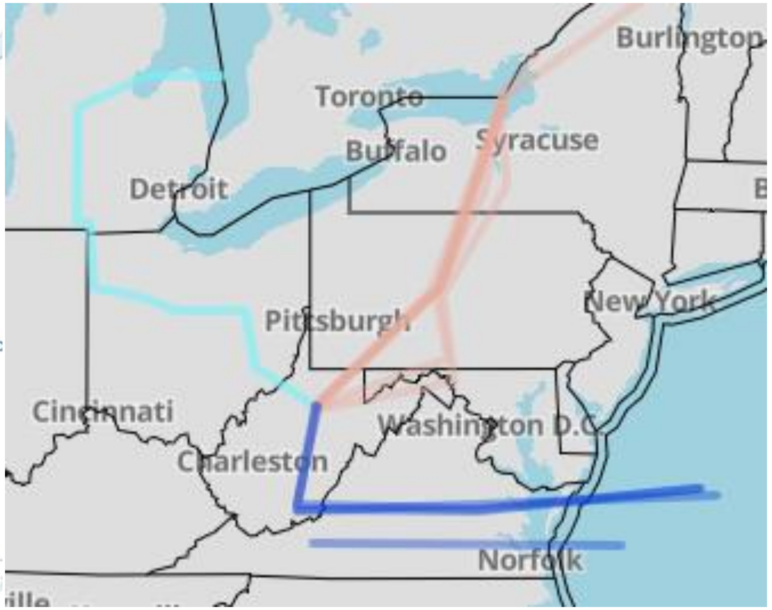
Traffic Management Initiatives (TMIs)

Source:
https://en.wikipedia.org/wiki/Jet_stream
Reynolds, T. G. (2008, September). Analysis of lateral flight inefficiency in global air traffic management. In 8th AIAA Aviation Technology, Integration and Operations Conference, Anchorage, Alaska, US.

Applications (I)

- Traffic Management Initiatives (TMI) Planning
  - Sector based operations.
  - Deterministic flight trajectory predictions: historical flight tracks/ filed flight plans.
  - Drawbacks: unable to capture uncertainties.





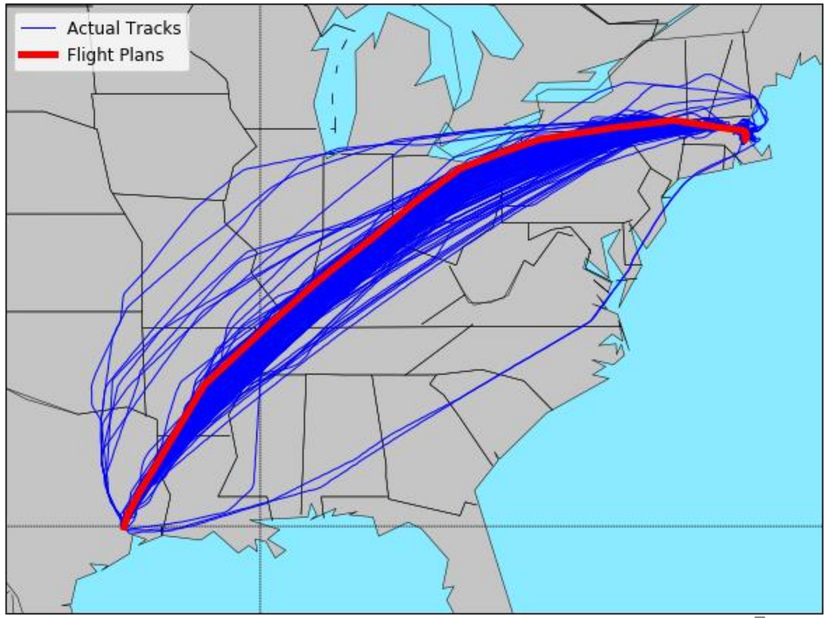


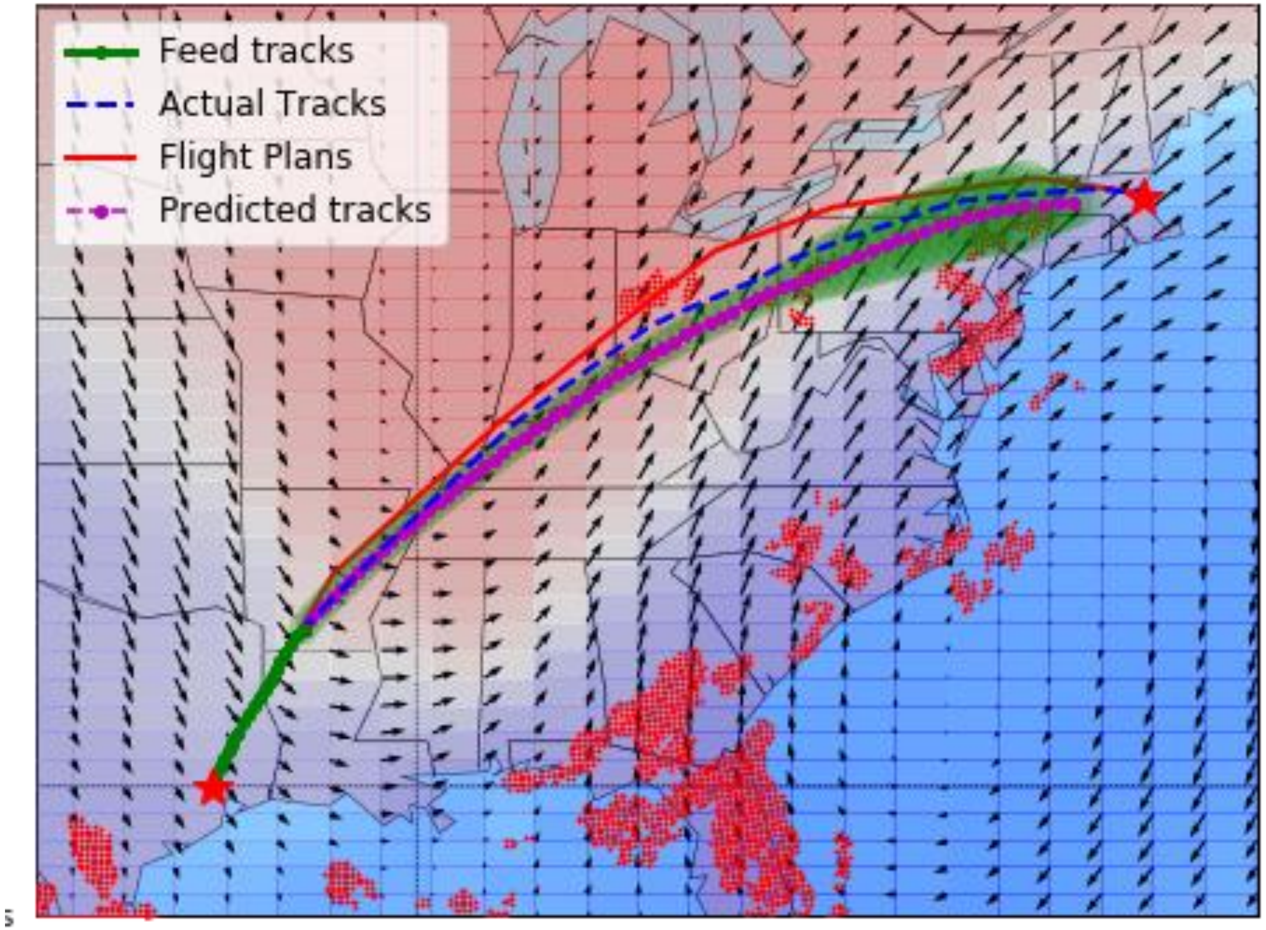
Image courtesy of: [fly8ma.com]() and Alex Estes.

Applications (II)

- Trajectory Based Operations (TBO)
  - Uses the 4D trajectories to both strategically manage and tactically control surface and airborne operations.
  - It is essentially asking, where the flight is going to be at the time of interest.
- Trajectory Prediction Tool
  - Central to TBO analysis.
  - Estimate sector demands and plan TMI.
  - Performance analysis.

Research Goal

- Develop aircraft trajectory prediction tools that can
  - Incorporate uncertainties such as adverse weather and wind.
  - Predict flight tracks given initial conditions.
  - Provide prediction intervals.



Highlights

- Generative Model
  - Inputs related to convective weather, winds, and temperature in the vicinity of the aircraft, as well as the flight plan information.
  - Outputs flight coordinates modeled as Gaussian Mixtures.
- Feature Engineering
  - Efficient tree-based spatiotemporal matching algorithm.
  - Batch mode matching during training.
  - Recursive matching during inference.

Outline

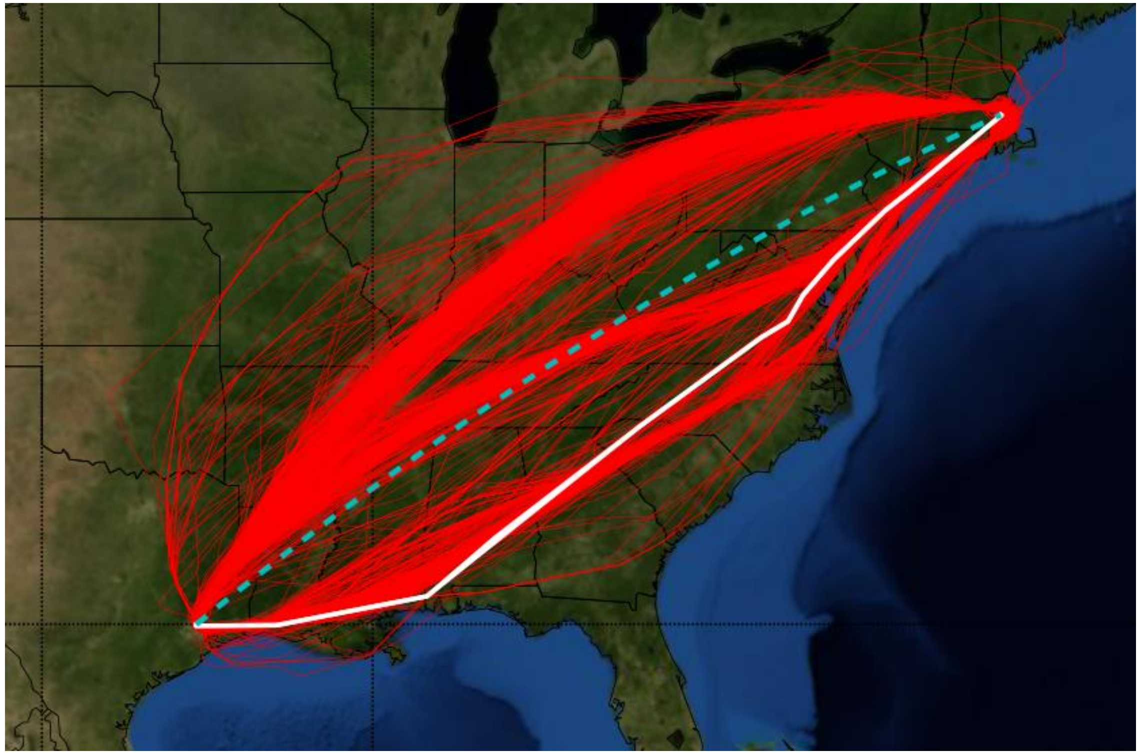
- Introduction
- Data Sources and Feature Engineering
- Methodology
- Numerical Experiments
- Summaries

Data Sources

- Flight Track Data
- National Convective Weather Data
- Wind Speed Data
- Temperature Data

Data Sources – TFMS

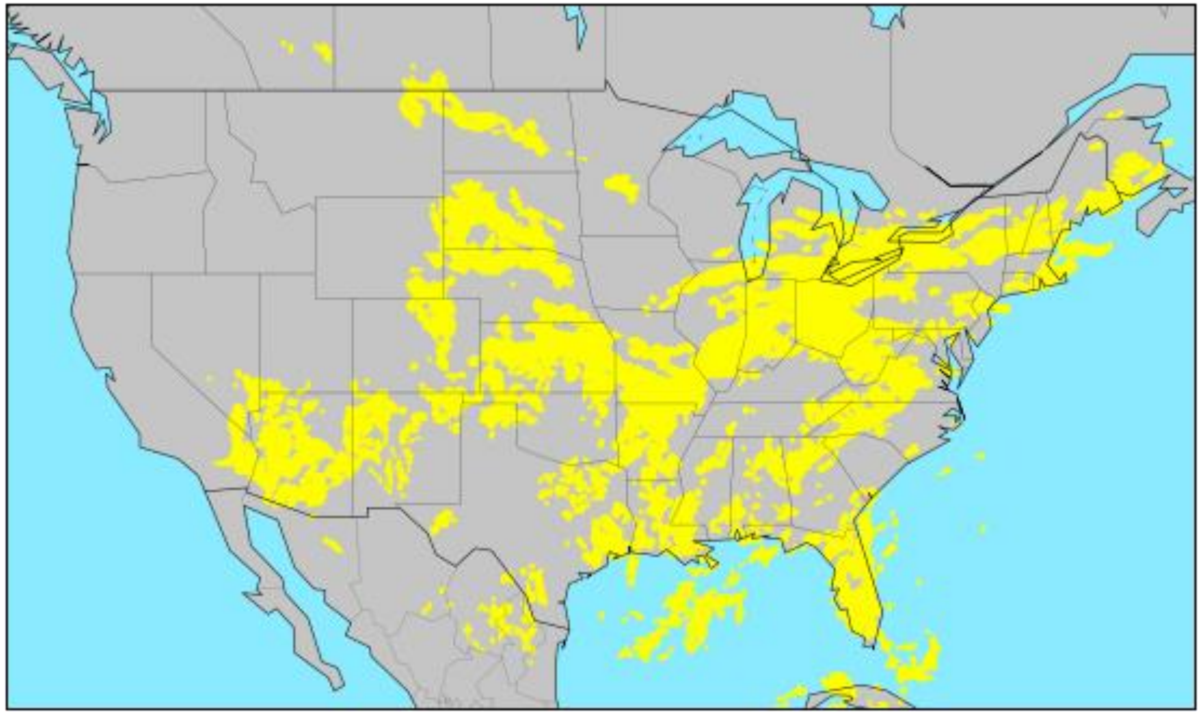
- Flight Track/ Plan Data
  - Come from FAA Traffic Flow Management System.
  - Typically 60-second update.
  - Latitude, longitude, altitude, time
  - Derived the vertical and horizontal speeds.



Data Sources – NCWF

- National Convective Weather Data (NCWF)
  - 5-minute update nowcast.
  - Convective weather polygons with altitude (flight level) and speed.

NCWF convective weather polygon

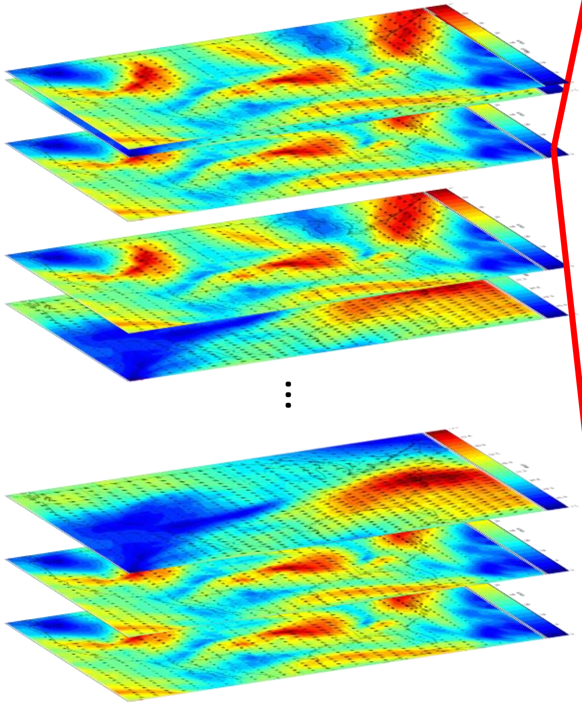


Data Sources – NAM

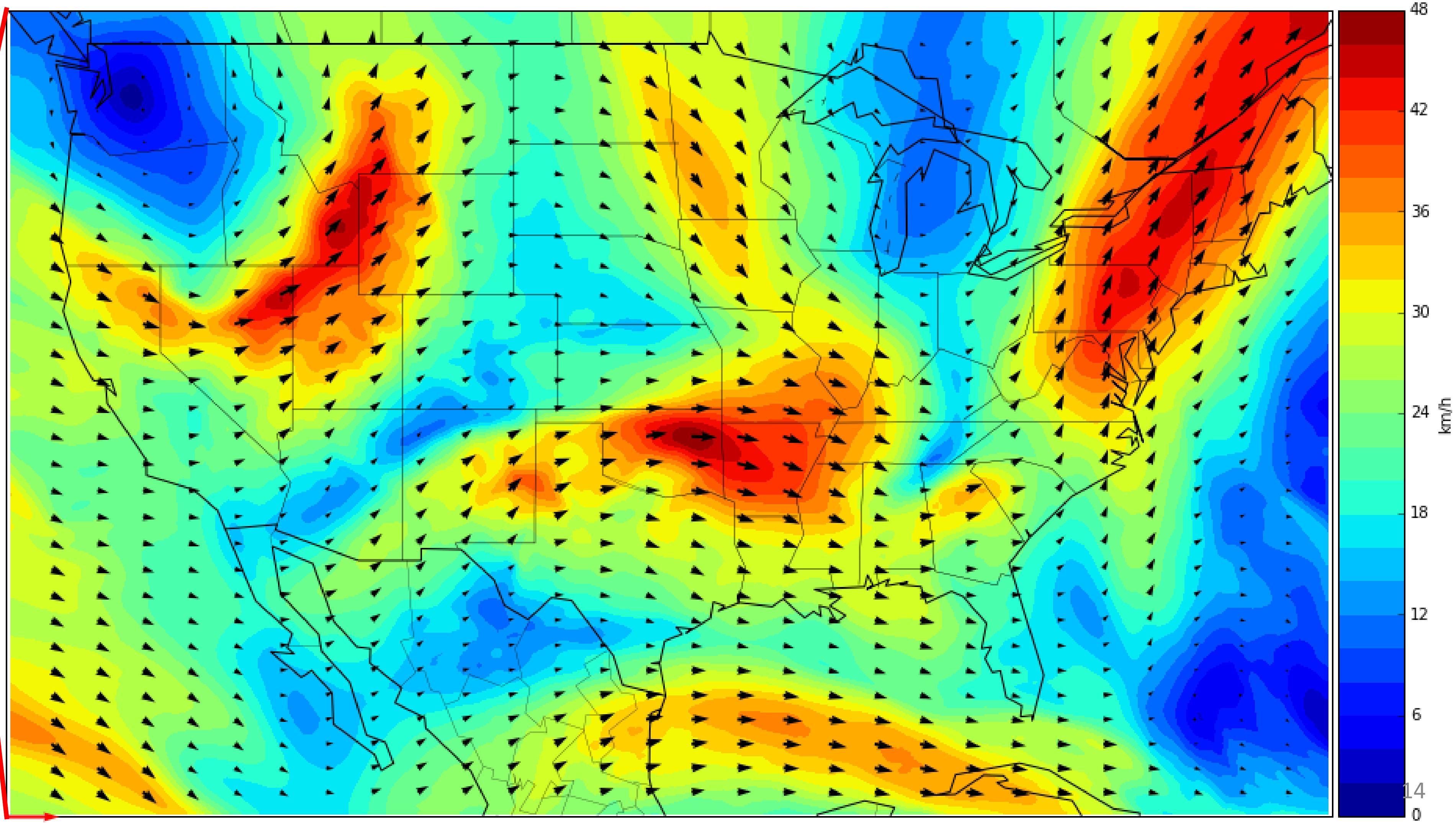
- North American Mesoscale Data (NAM)
  - Updated 4 cycles per day: 00:00; 06:00; 12:00; 18:00.
  - Each cycle produced 5 datasets at 0 hour, 1 hour, 2 hour, 3 hour and 6 hour.
  - Wind speed and temperature.
  - $0.1° \times 0.1°$
  - 39 altitude levels: 50 mbar (~68000 ft) to 1000mbar (~0 ft).

Data Sources – NAM

39 isobaric pressure levels
For each level: 614 x 428

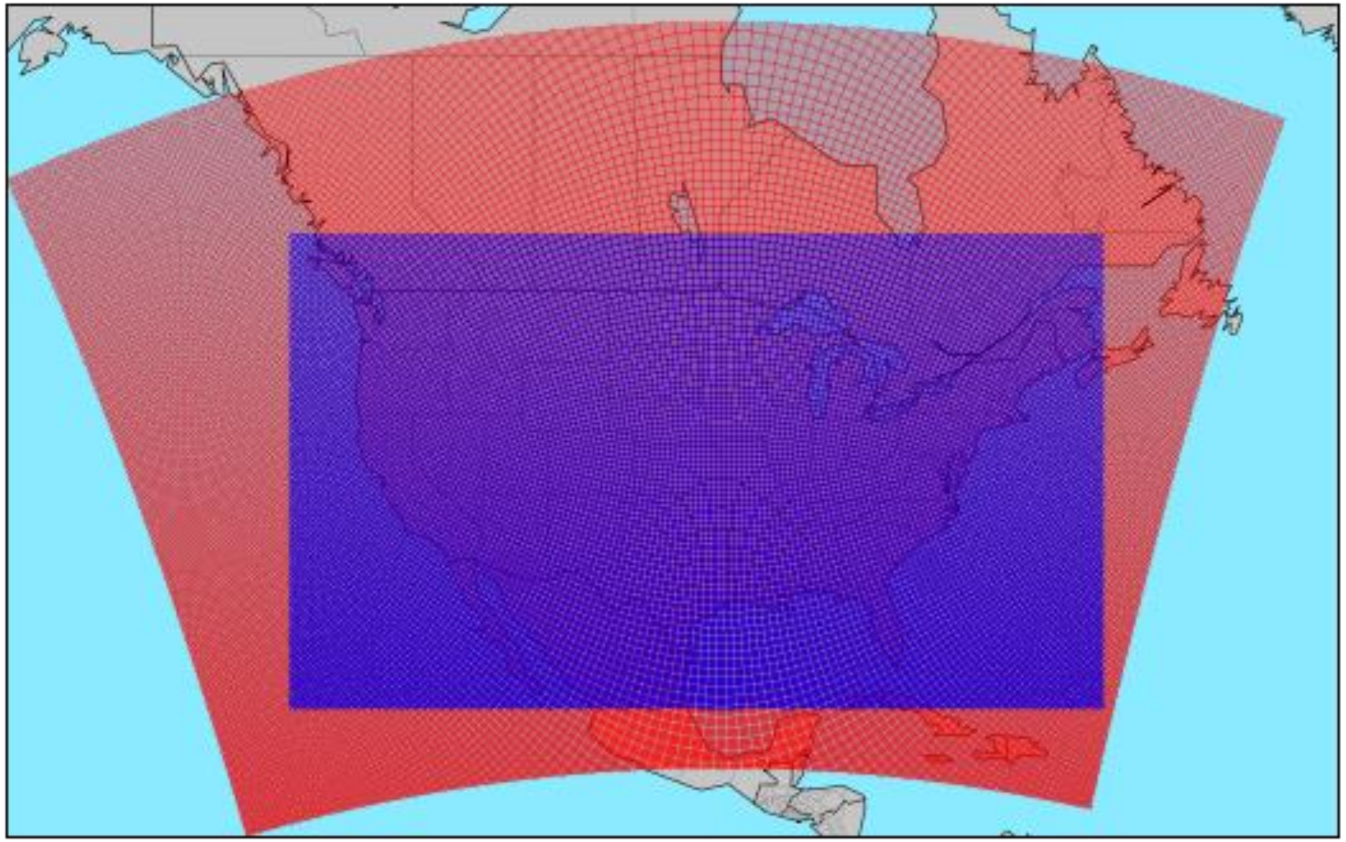


Wind Field Diagram (km/h) @ 200 mbar (~ 38,000 ft.); 02/04/2013 18:00 Zulu



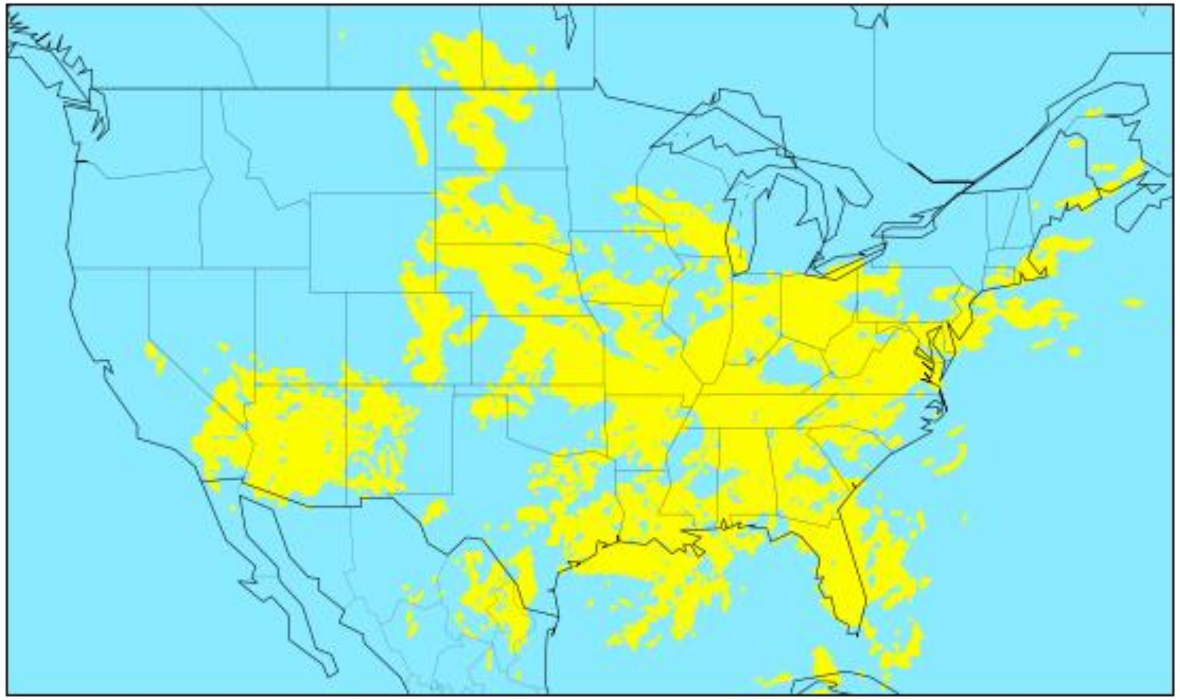
Feature Engineering – Georeferencing

- Based on NAM weather data
- Red
  - Original georeferencing grid
  - 428×614
- Blue
  - Cropped georeferencing grid
  - Latitude: $22° \sim 52°$
  - Longitude: $-130° \sim -64°$
  - $\sim 336 \times 413$



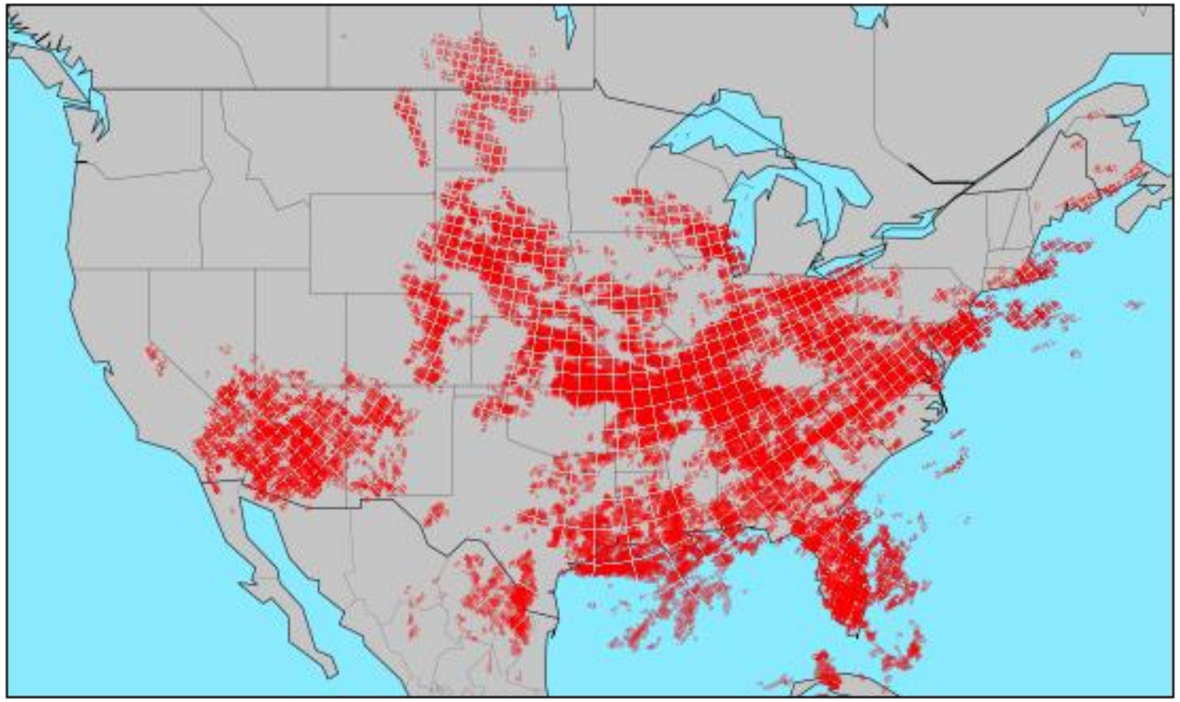
Feature Engineering – Discretize Weather Data

Original (2D projection)



Original data describe the boundary and altitude of convective weather polygons

Discretized (2D projection)

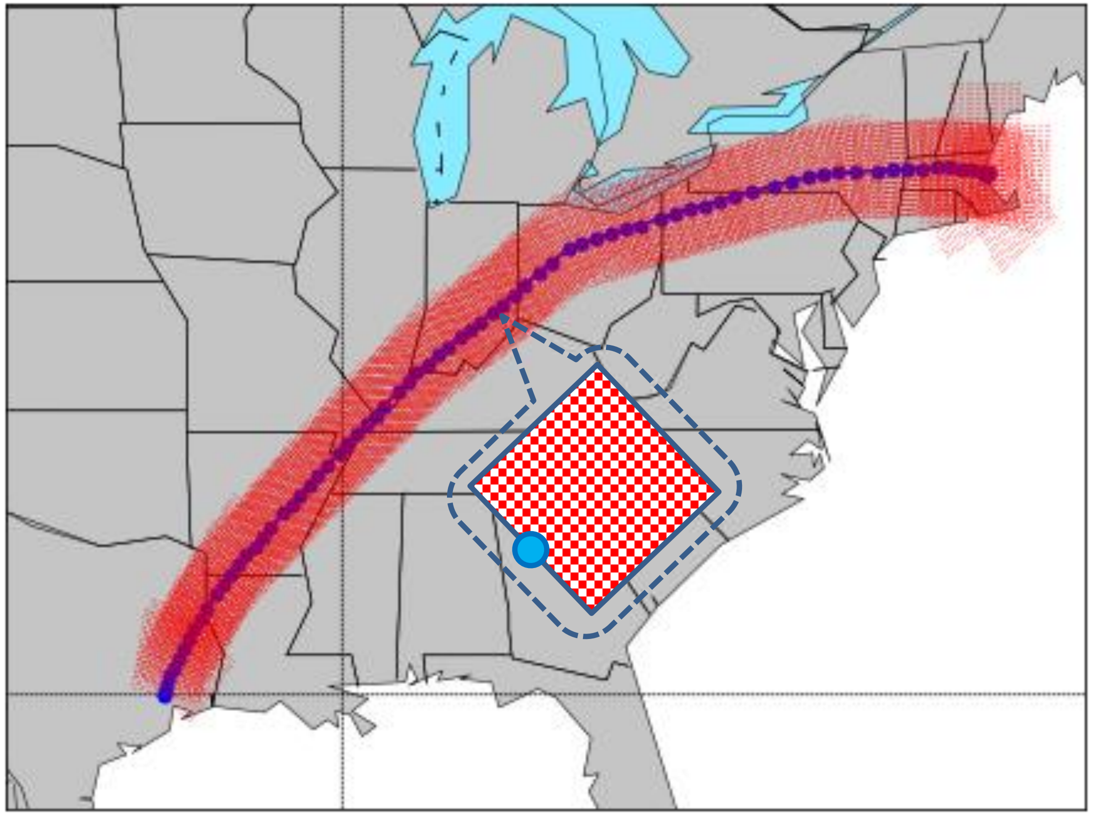


Discretized weather information will be stored as binary variables in a matrix spanned by our georeferencing grid. The red points on the graph are the nonzero elements.

Feature Engineering – Feature Cube Path

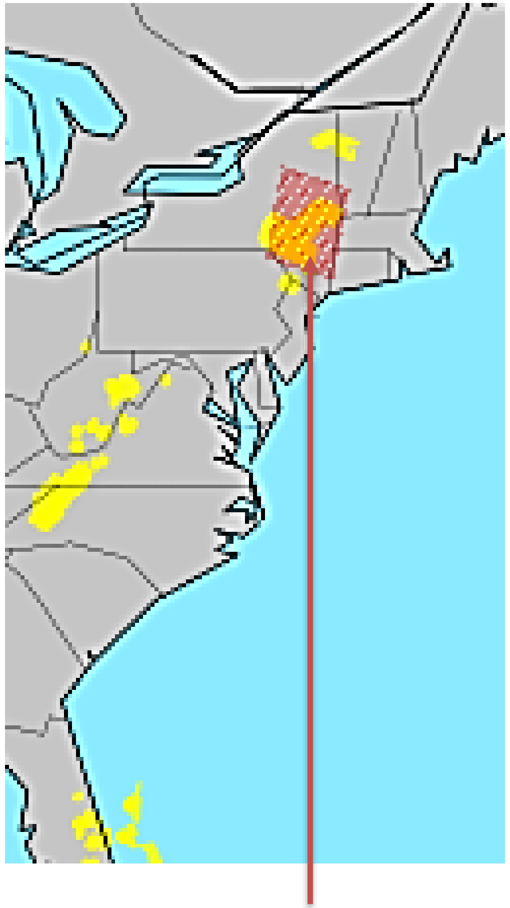
- 2D grid
  - For each track point, create a 20x20 grid matrix with one side centered at the track point, and oriented by the azimuth.
- Altitude buffer
  - NAM: flight level → pressure altitude.
  - NCWF: flight level $\pm$ 200 FL
- Time buffer
  - NAM: track time $\pm$ 3 hours
  - NCWF: track time $\pm$ 1 hour

Flight grid path is oriented by the azimuth.



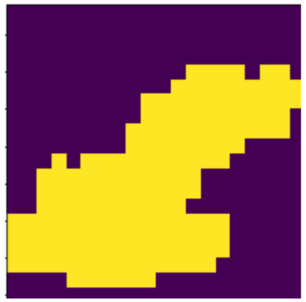
Feature Engineering – Matching

- Batch Mode Tree-based 4D Matching
  - Temporal trees: query to find the closest time instances to the flight tracks.
  - Spatial trees: query to find the closest location instances to the flight grid path.

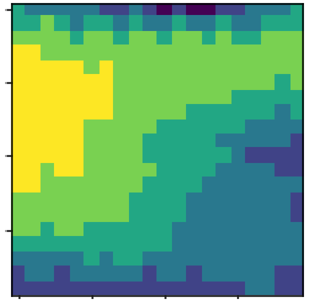


One grid on the flight grid path

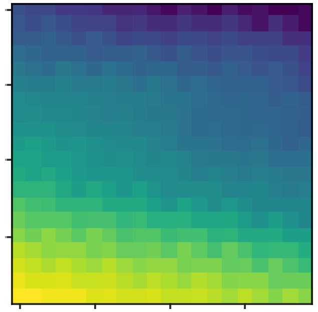
NCWF weather



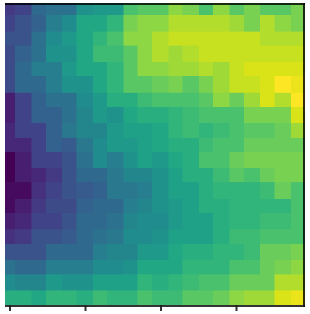
Temperature



Westly wind speed



Southerly wind speed

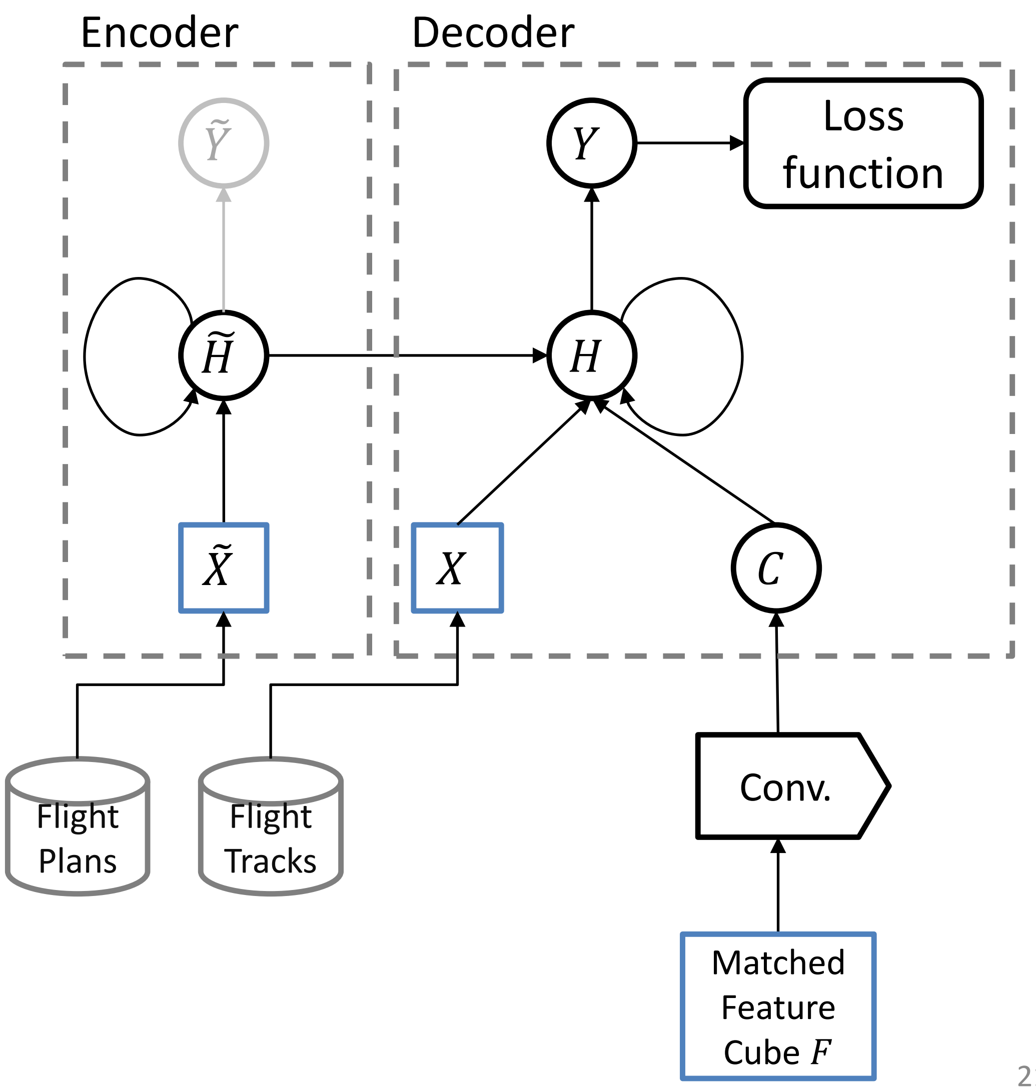


Outline

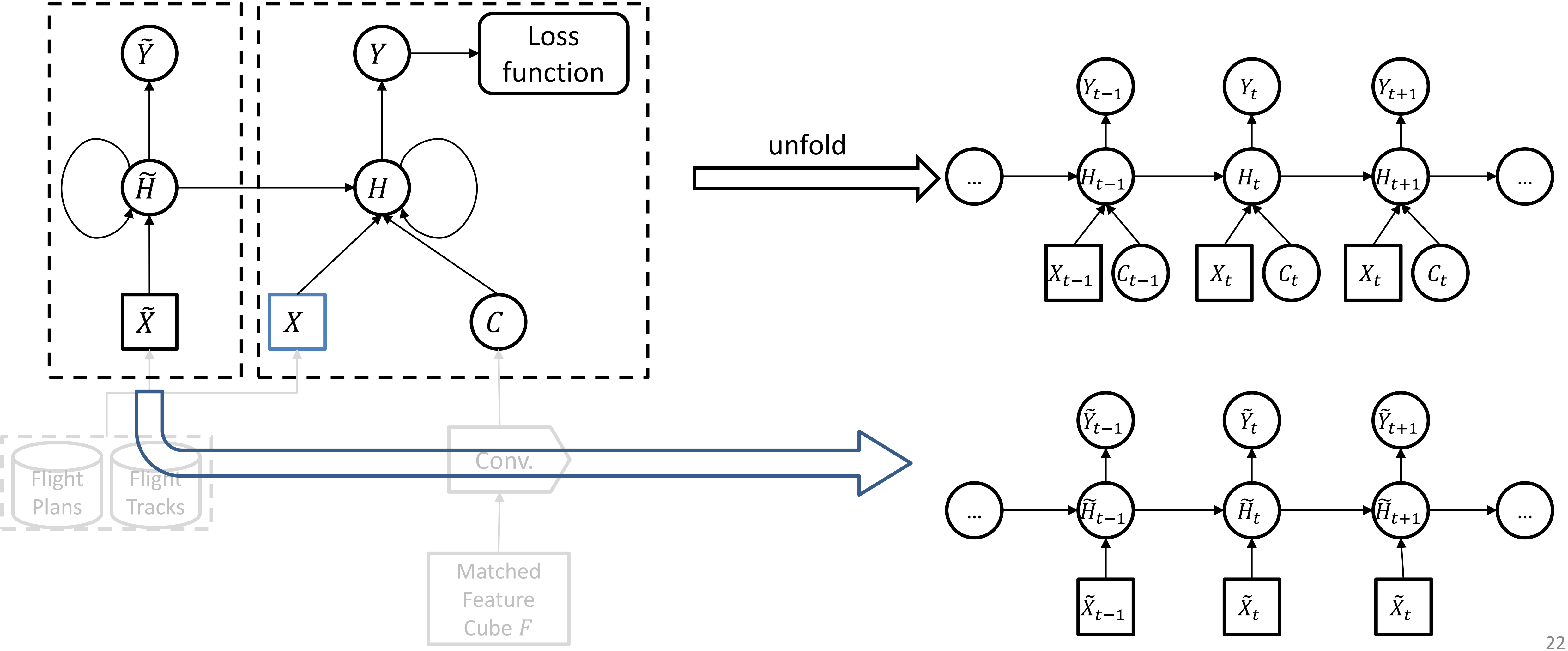
- Introduction
- Data Sources and Feature Engineering
- Methodology
- Numerical Experiments
- Summaries

Methodology

- Training framework
  - Encoder LSTM network: embed flight plan sequence information.
  - Decoder LSTM network: learn the spatiotemporal correlation from flight tracks.
  - Convolutional layers: learn feature representations from high-dimensional meterological feature cubes.
- The predicted flight tracks are modeled as **Mixture of Gaussians** whose parameters are learned by the decoder network.

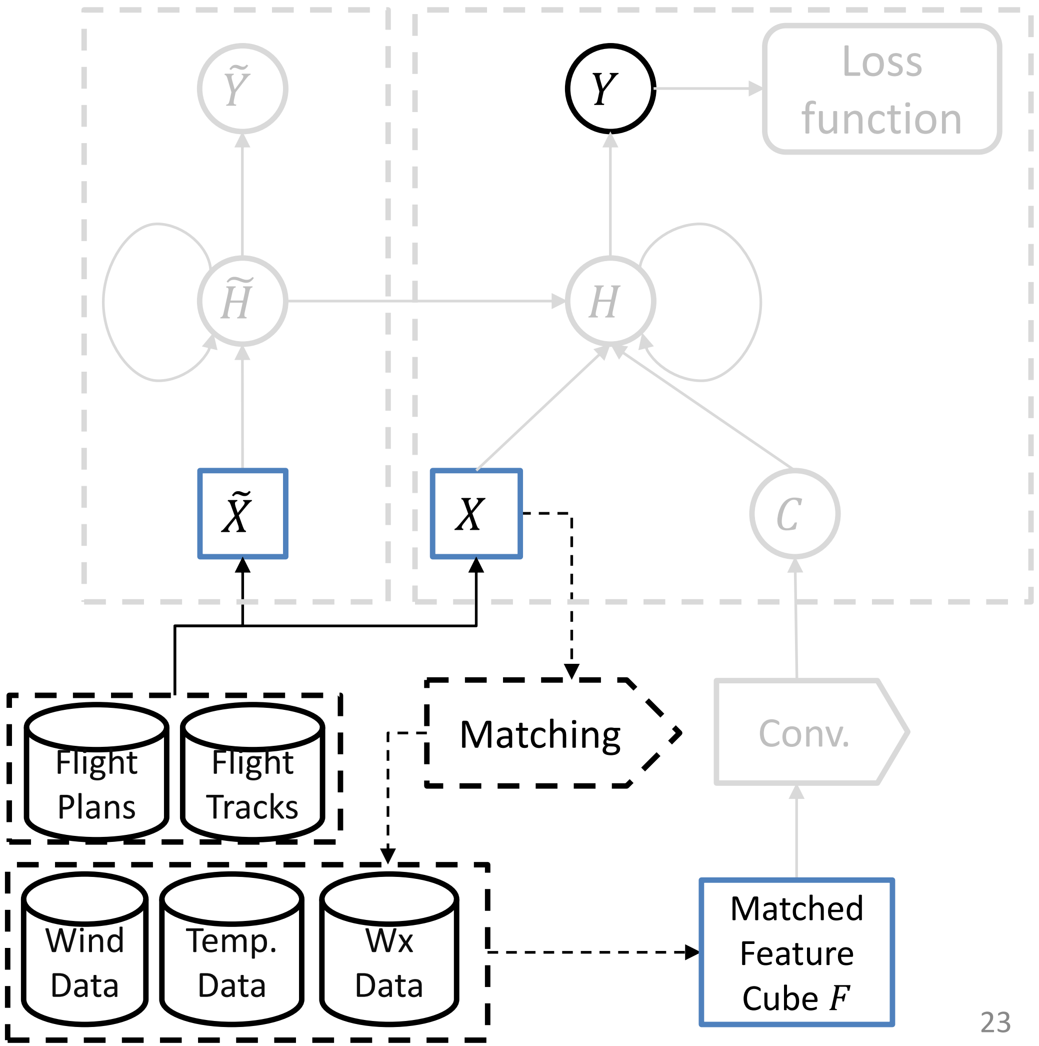


Framework – Training



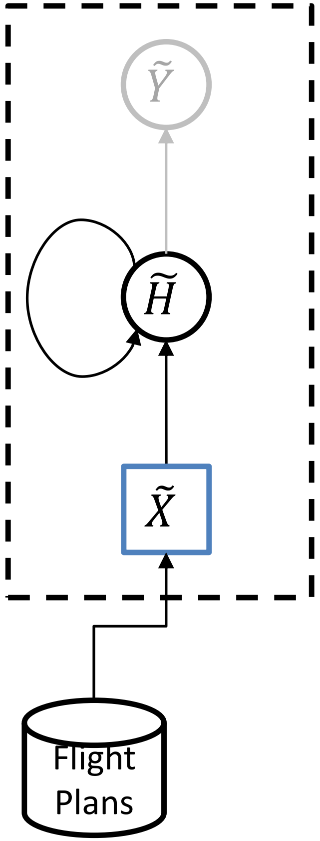
Framework – Training

- Inputs
  - $\tilde{X}$: Sequence of 2D coordinate (Lat, Lon) of flight plans.
  - X: Sequence of state variables of flight tracks.
  - $X_t = (Lat, Lon, Alt, Time, Lat\ speed, Lon\ speed)_t = (x, y, z, t, \dot{x}, \dot{y})_t$.
  - F: Sequence of matched feature cubes that correspond to X.
- Outputs
  - Y: Sequence of Gaussian mixture parameters
  - $Y_t = \left(\pi_i, \mu_x^i, \mu_y^i, \mu_z^i, \mu_{\dot{x}}^i, \mu_{\dot{y}}^i, \Sigma_{(x,y,z)}^{\mathrm{i}}, \Sigma_{(\dot{x},\dot{y})}^{\mathrm{i}}\right)_{\mathrm{t}}$

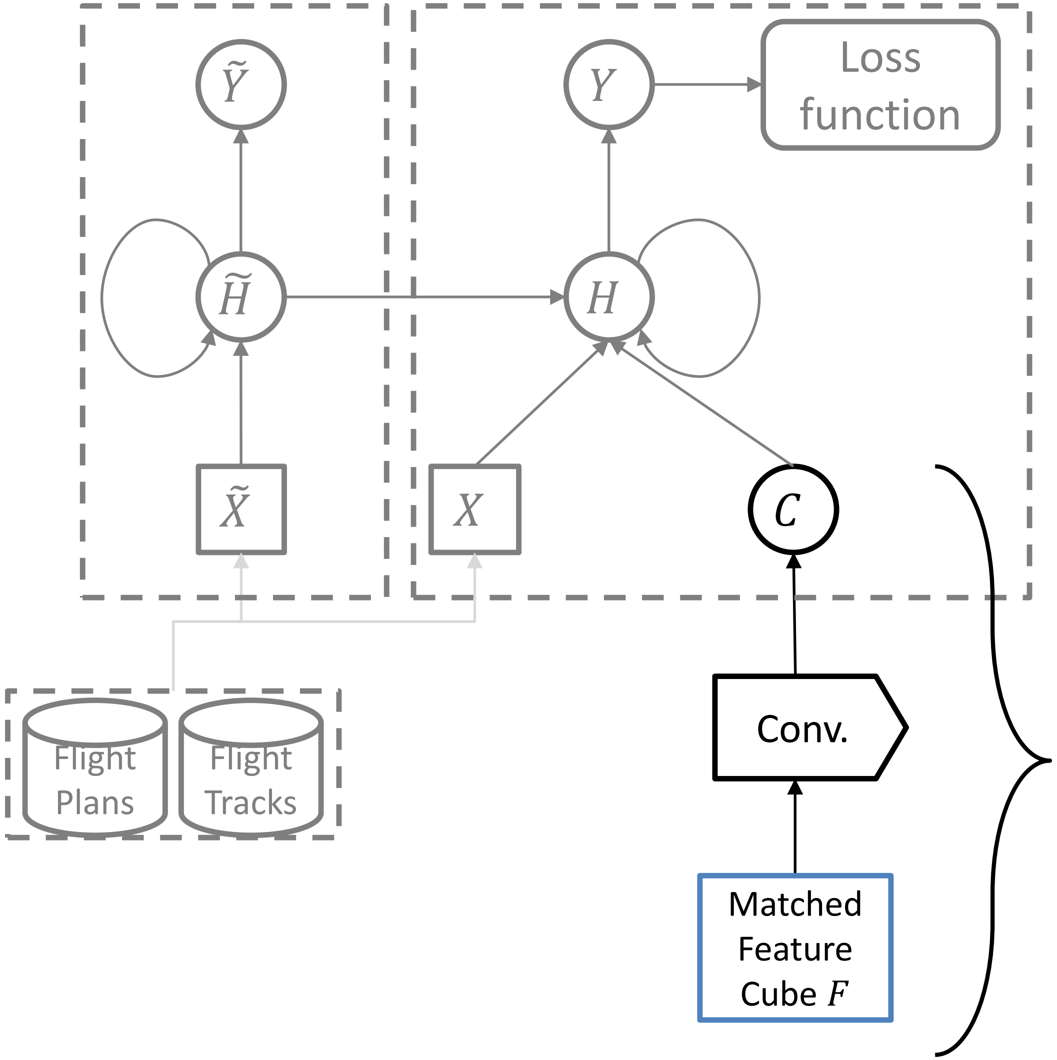


Framework – Training

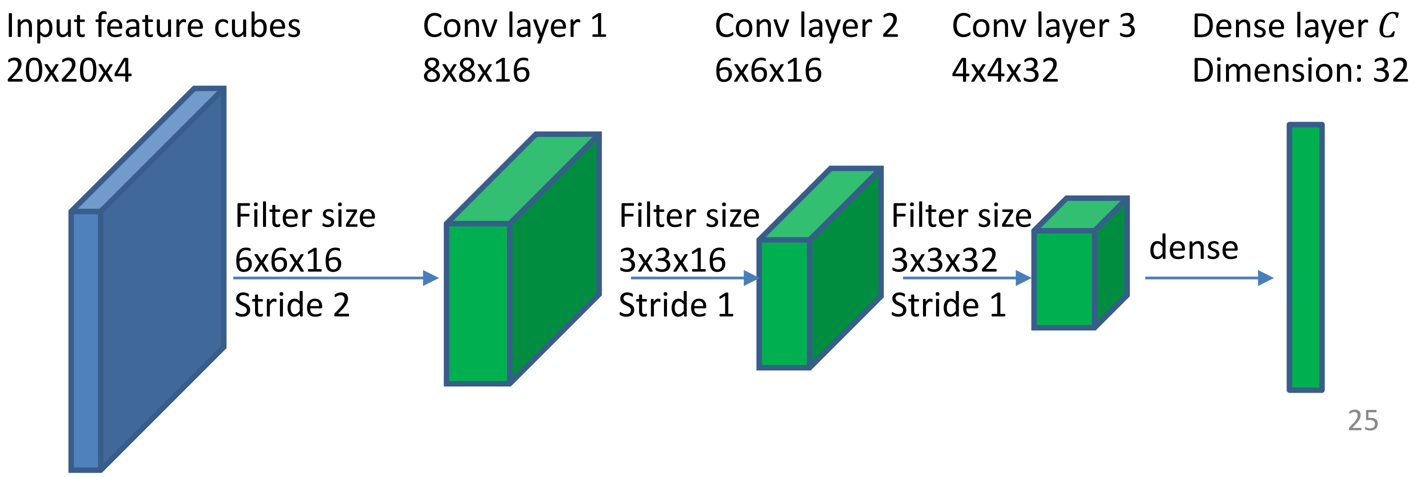
- Encoder LSTM
  - $\tilde{X}$: Sequence of 2D coordinate (Lat, Lon) of flight plans.
  - $\widetilde{H}$: hidden layers of a two-layer LSTM with 128 neurons.
  - Flight plan information $\tilde{X}$ is therefore embedded by encoder LSTM into a fixed-length variable $\widetilde{H}$.



Framework – Training



- Convolutional Layers
  - Three convolutional layers and one dense layer.
  - No pooling layers, and no padding operations.
  - Dense layer has 32 neurons that act as the feature representations from the high-dimensional meteorological feature cubes (20x20x6=2400).

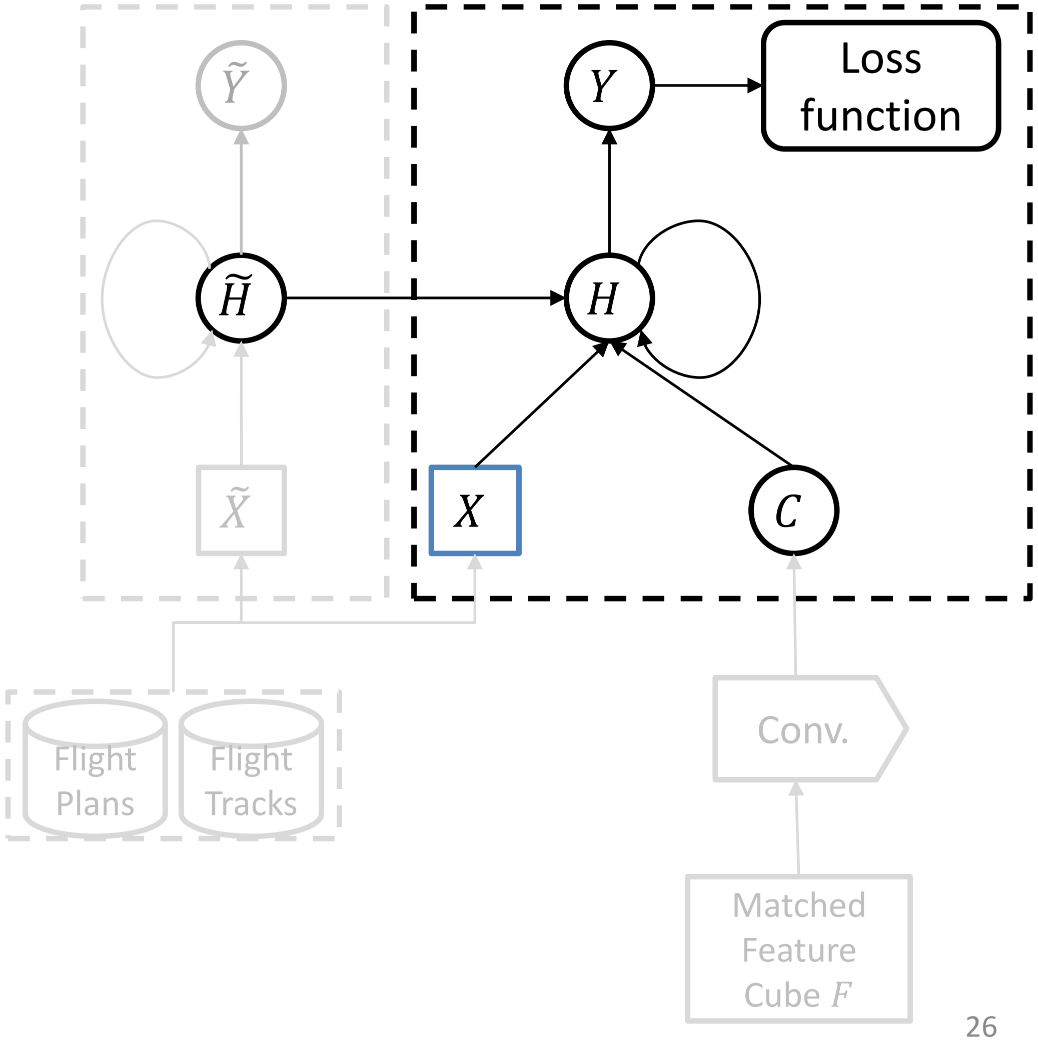


Framework – Training

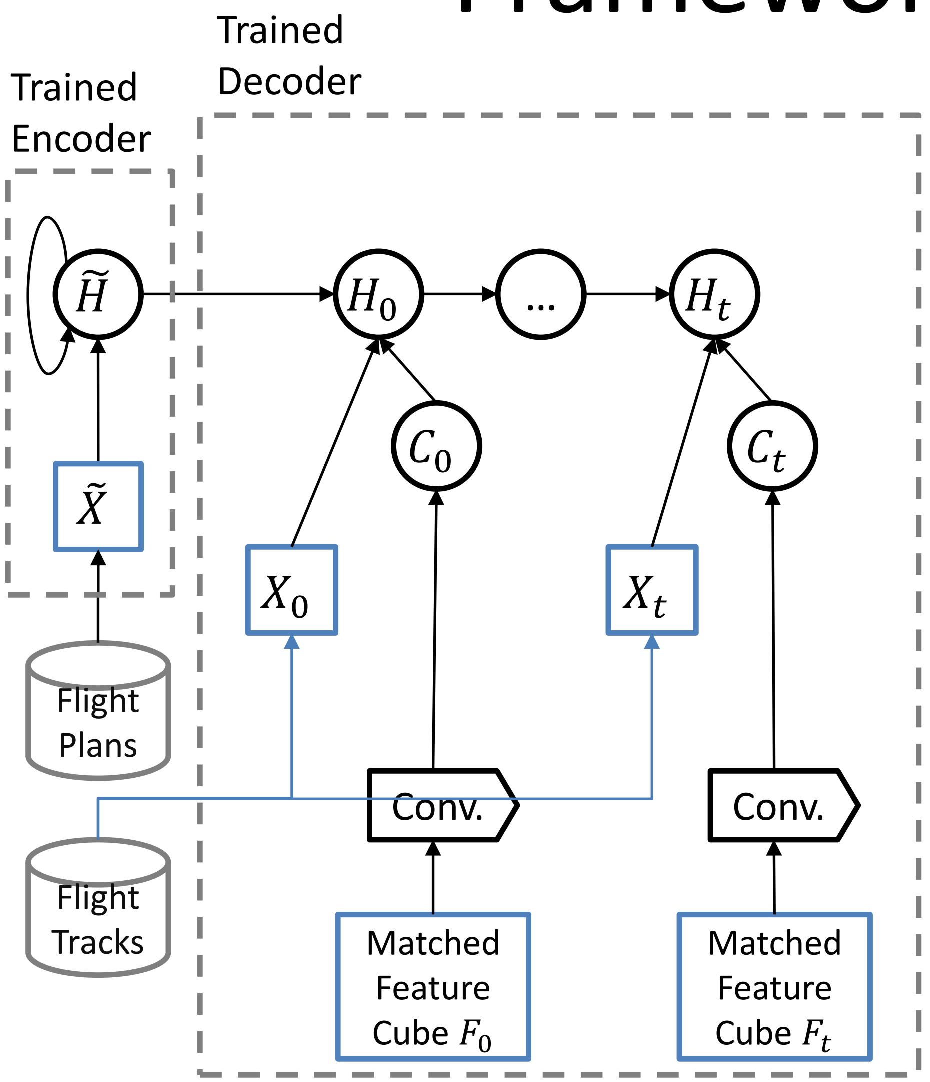
- Decoder LSTM
  - C: fixed-length feature vector learned by CNN (dense layer) from high-dimensional weather data.
  - Both X and C are fed into an embedding layer with 64 neurons before entering the decoder network.
  - Y: gaussian mixture parameters: $\pi_i, \mu^i_{lat}, \mu^i_{lon}, \mu^i_{alt}, \mu^i_{latspd}, \mu^i_{lonspd}$ $\Sigma^i_{(lat,lon,alt)}, \Sigma^i_{(latspd,lonspd)}$.
  - Loss function

$$L = -\sum_t^T \log\left[\sum_i \pi_i P_i\left(\widehat{X_{t+1}} \middle| Y_t, X_t, X_{t-1}, \ldots\right)\right]$$

$$P_i\left(\widehat{X_{t+1}} \middle| Y_t, X_t, X_{t-1}, \ldots\right) \sim N\left(\begin{bmatrix}\mu^i_{lon}\\ \mu^i_{lat}\\ \mu^i_{alt}\end{bmatrix}, \Sigma^i_{(lat,lon,alt)}\right) N\left(\begin{bmatrix}\mu^i_{latspd}\\ \mu^i_{lonspd}\end{bmatrix}, \Sigma^i_{(latspd,\,lonspd)}\right)$$



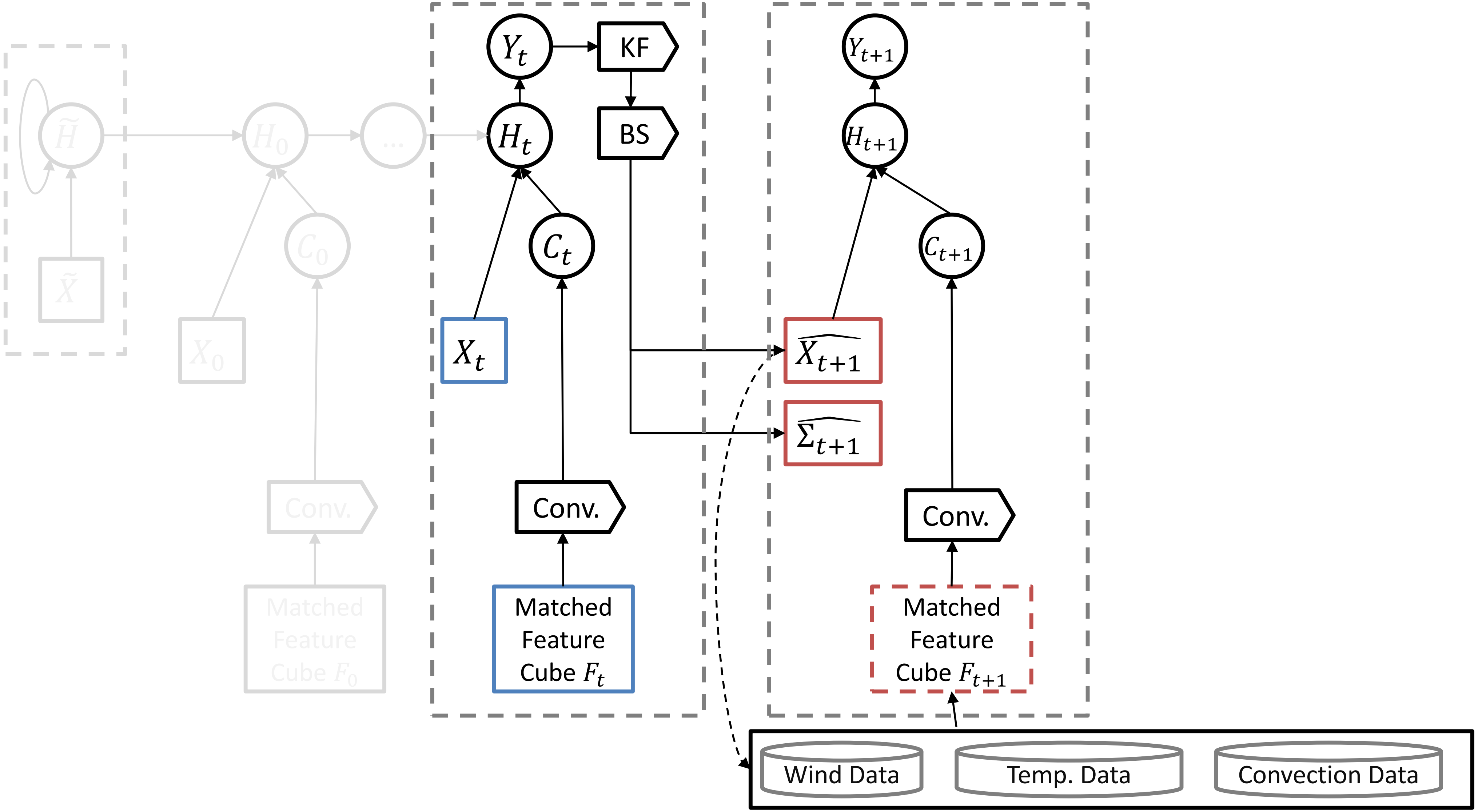
Framework – Inference/ Sampling



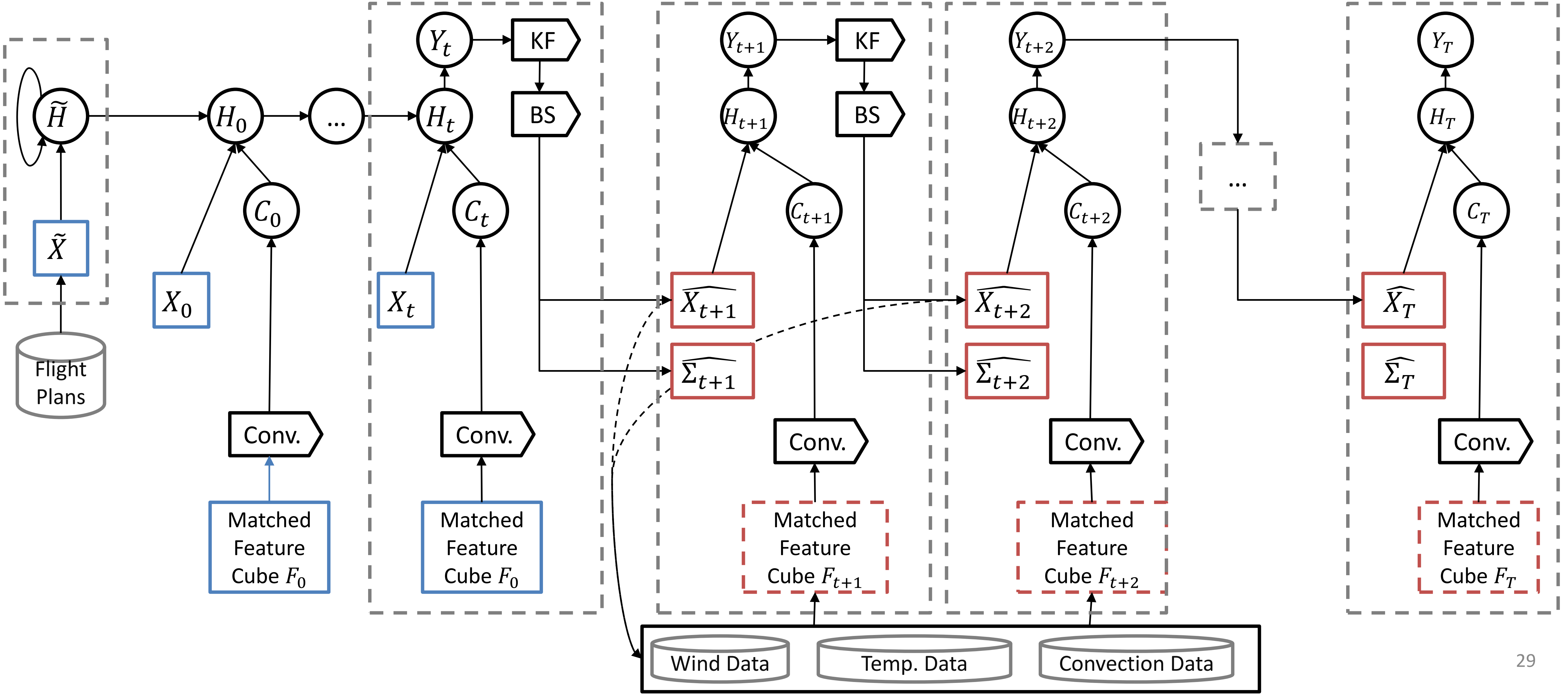
- Inputs
  - Trained networks
  - Sequence of flight plans
  - Sequence of first t flight coordinates
- Outputs
  - The predicted flight coordinates after time t.
  - Confidence interval of every predicted flight coordinates.

Framework – Inference/ Sampling

$$Y_t = \left(\pi_i, \mu_x^i, \mu_y^i, \mu_z^i, \mu_{\dot{x}}^i, \mu_{\dot{y}}^i, \Sigma^{\mathrm{i}}_{(x,y,z)}, \Sigma^{\mathrm{i}}_{(\dot{x},\dot{y})}\right)_{\mathrm{t}}$$



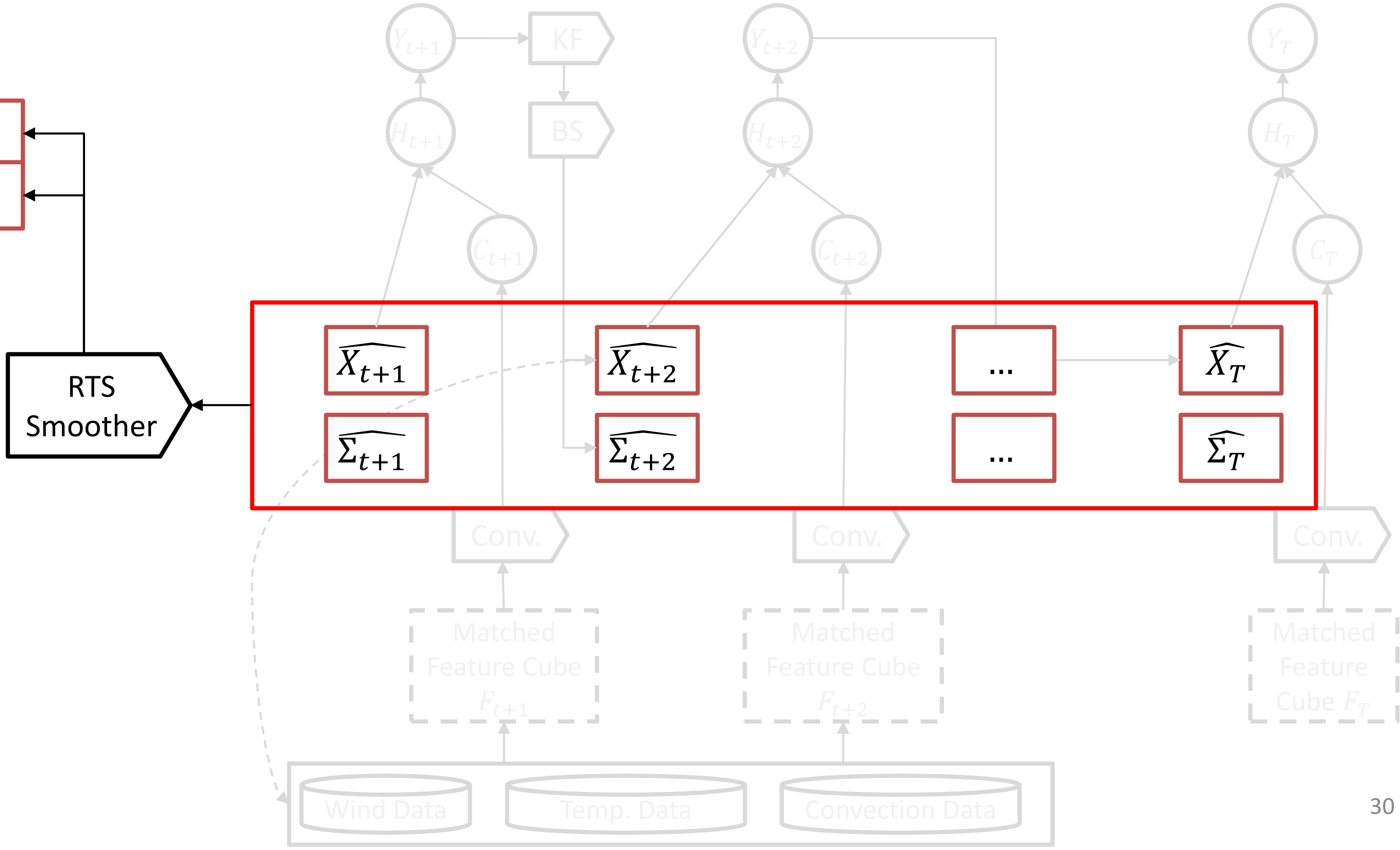
Framework – Inference/ Sampling



Framework – Inference/ Sampling

Best k sampled sequences

$\widehat{X_{t+1}}$	$\widehat{X_{t+2}}$	$\widehat{X_{t+3}}$	…	$\widehat{X_T}$
$\widehat{\Sigma_{t+1}}$	$\widehat{\Sigma_{t+2}}$	$\widehat{\Sigma_{t+3}}$	…	$\widehat{\Sigma_T}$

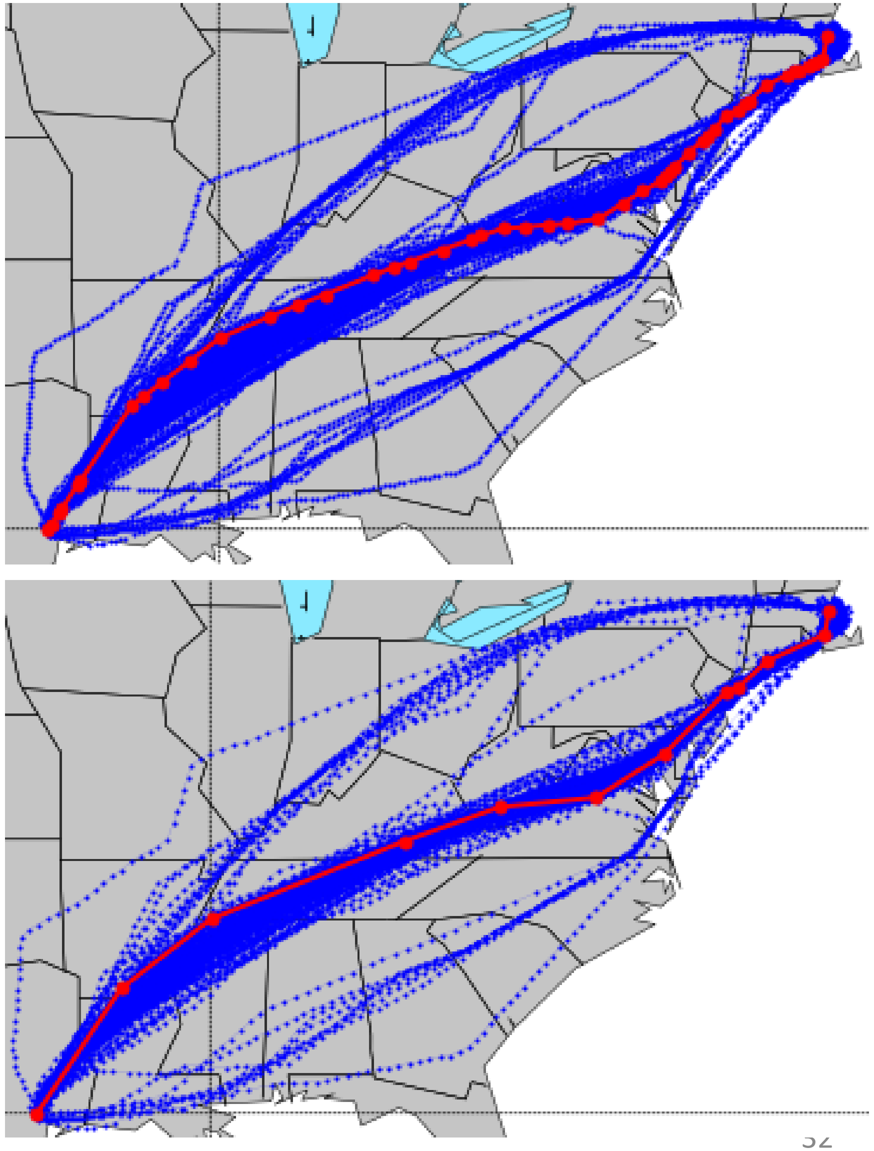


Outline

- Introduction
- Data Sources and Feature Engineering
- Methodology
- Numerical Experiments
- Summaries

Numerical Experiments

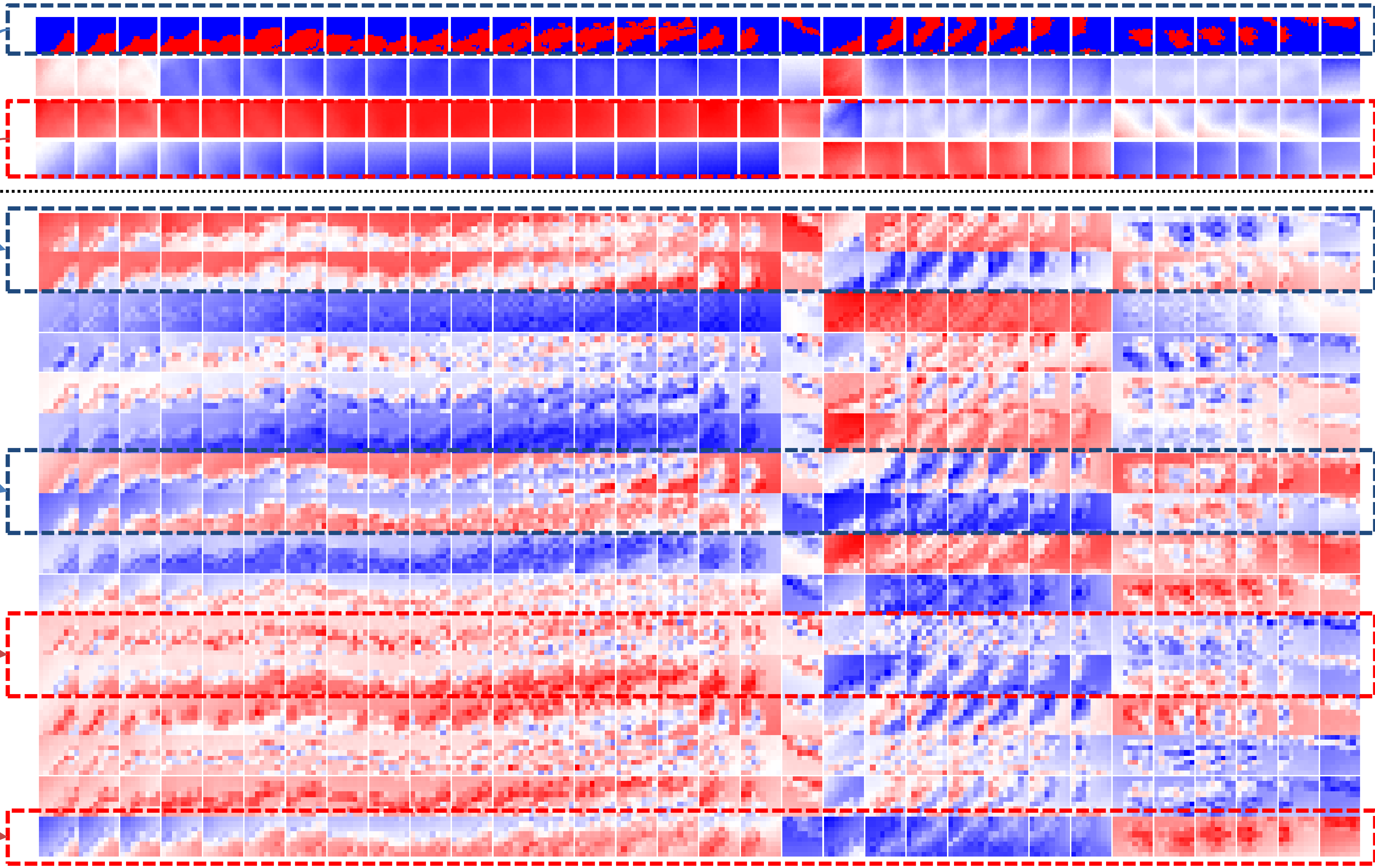
- Training Set
  - 1342 flights from IAH to BOS in 2013.
- Testing Set
  - 337 flights from IAH to BOS in 2013.
- Preprocessing
  - Downsample flight plans by identifying significant points.
  - Downsample flight tracks by half (Avg. sequence length = 94).
  - Normalize feature maps to 0 mean and unit variance.
- Training Specifics
  - Nesterov Momentum SGD.
  - Gradient clipping.
  - 3 Gaussian mixtures.
  - Dropout rate: 0.5



Feature Maps

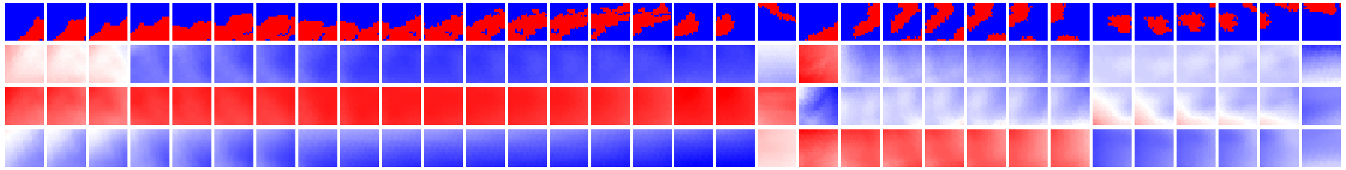
Input feature cubes

Conv. layer 1

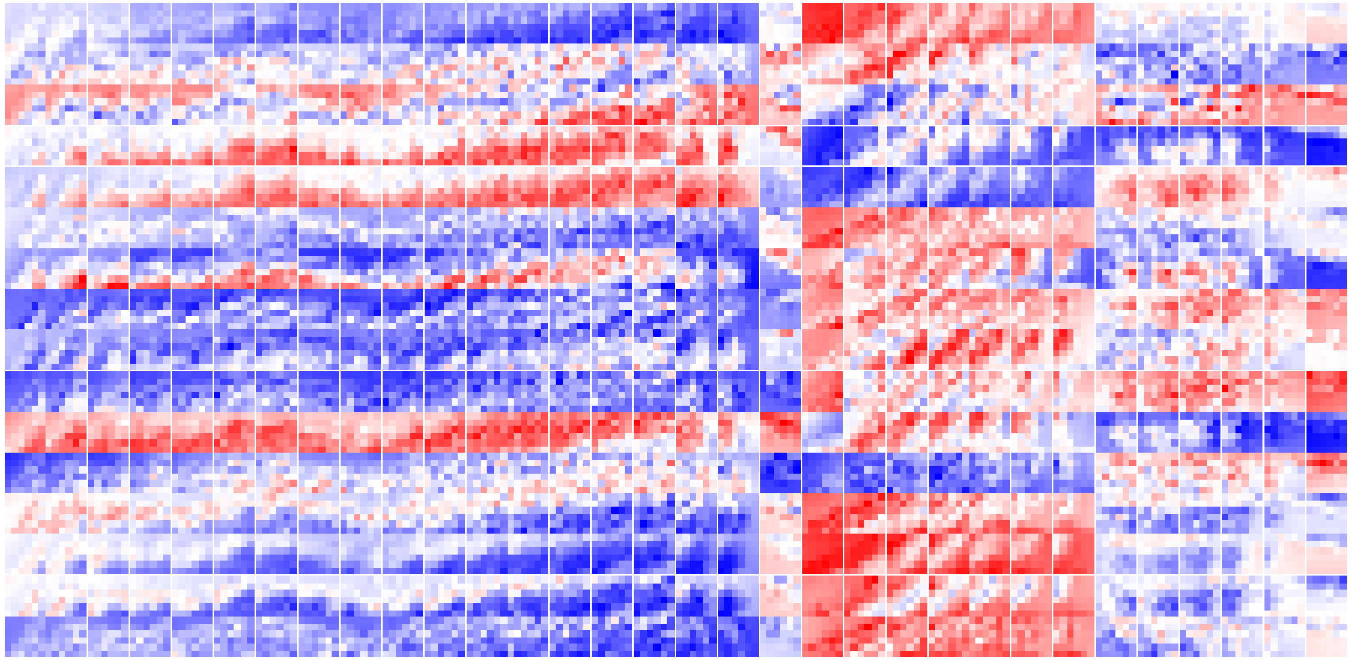


Feature Maps

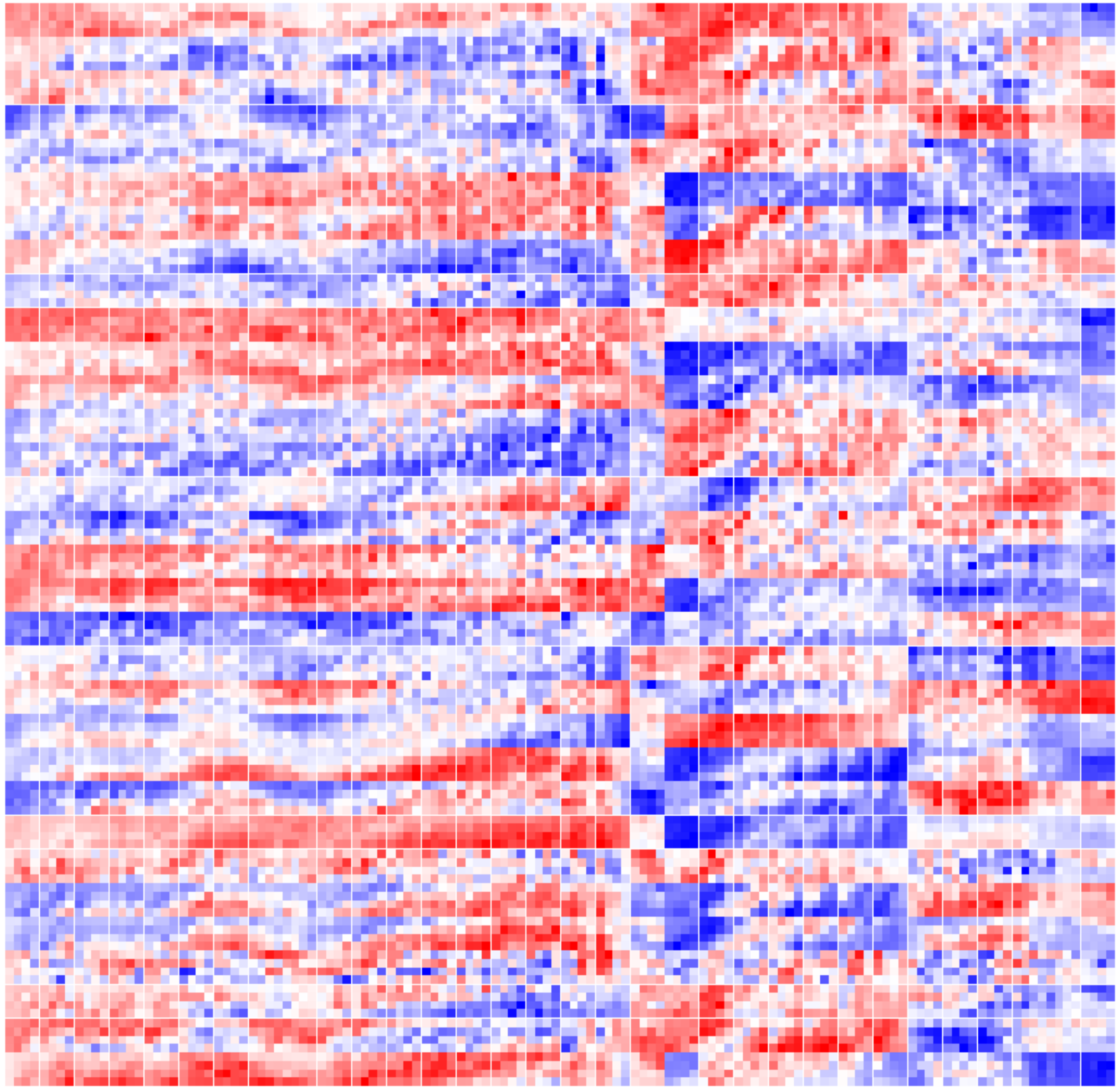
Input feature cubes



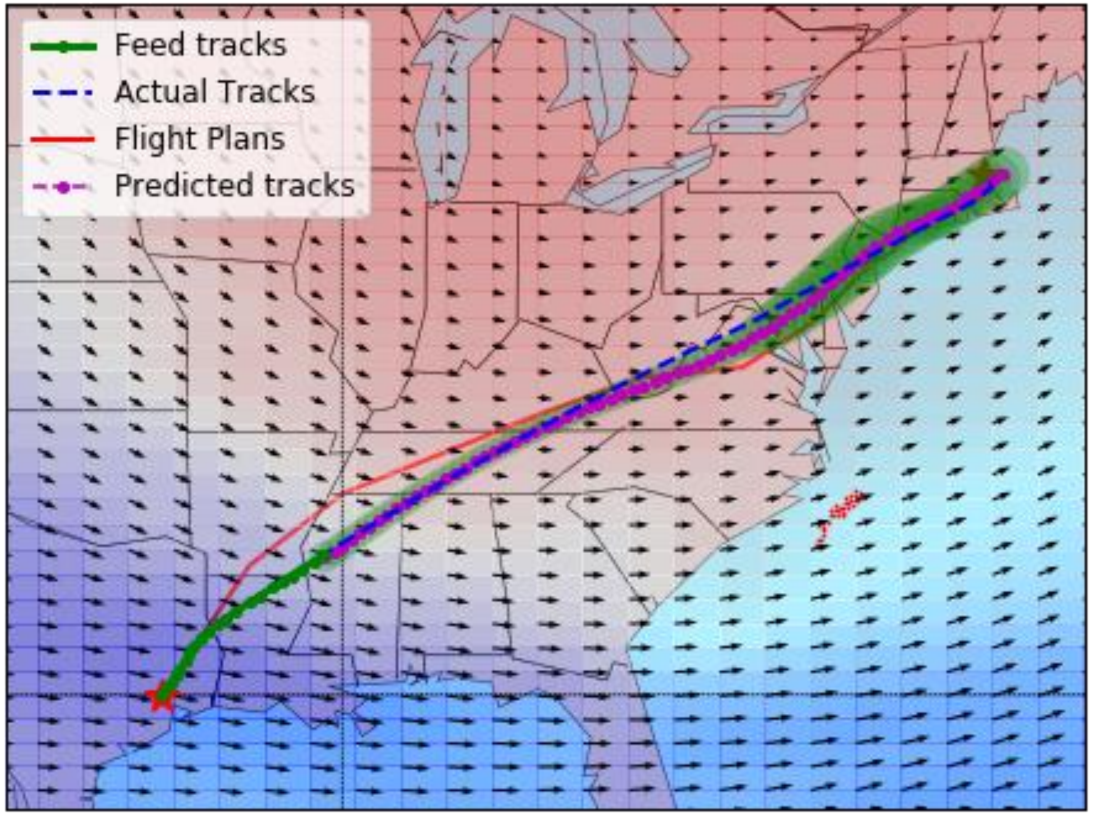
Conv. layer 2

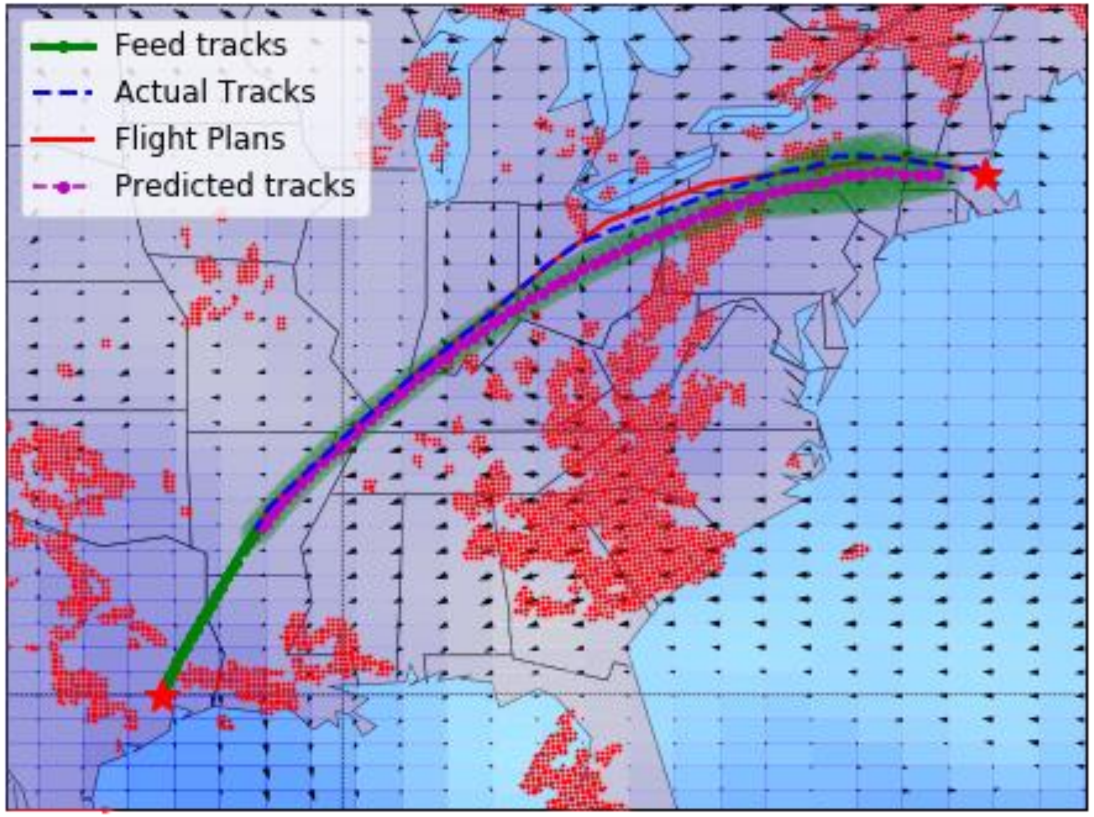


Conv. layer 3

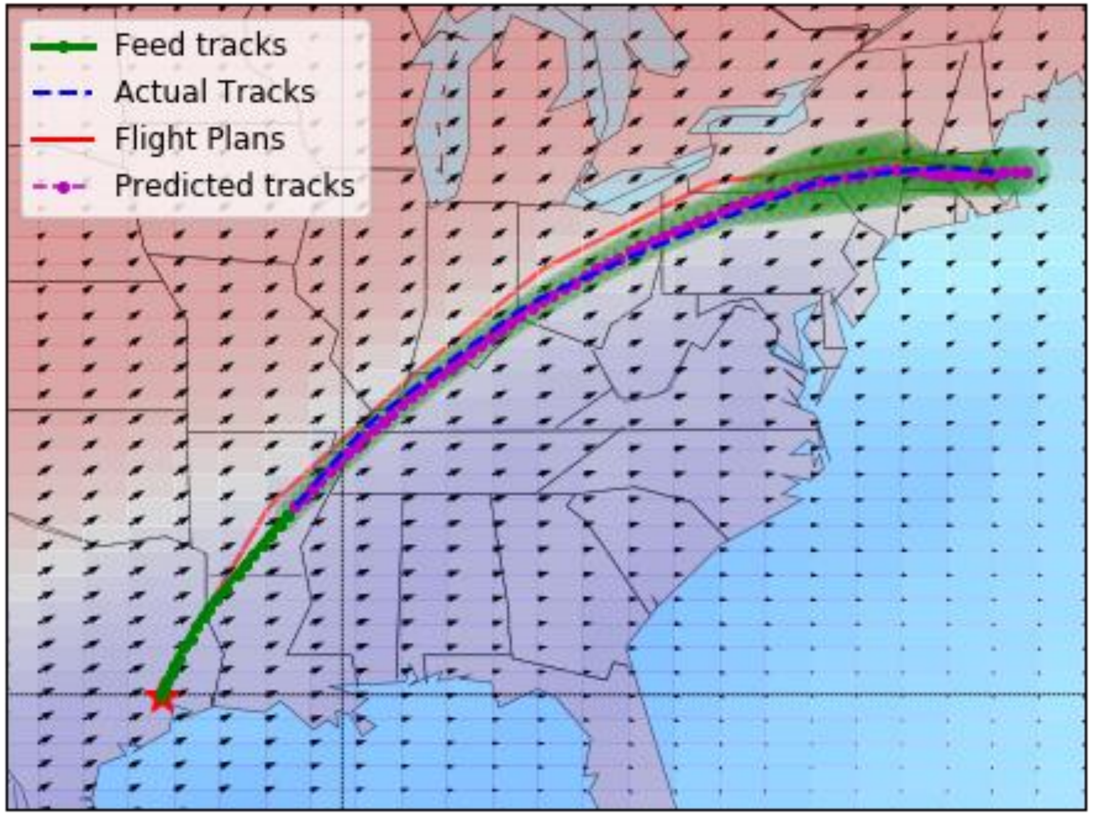


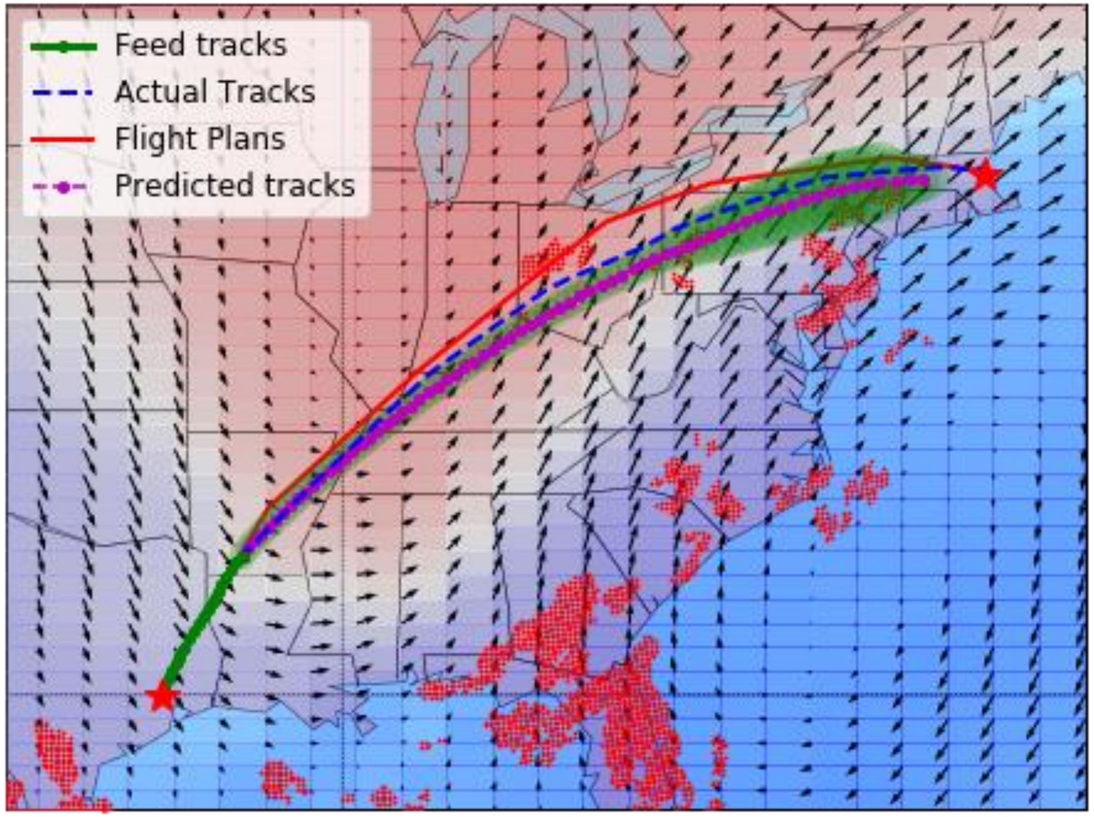
Evaluation


Feed tracks
Actual Tracks
Flight Plans
Predicted tracks


Feed tracks
Actual Tracks
Flight Plans
Predicted tracks

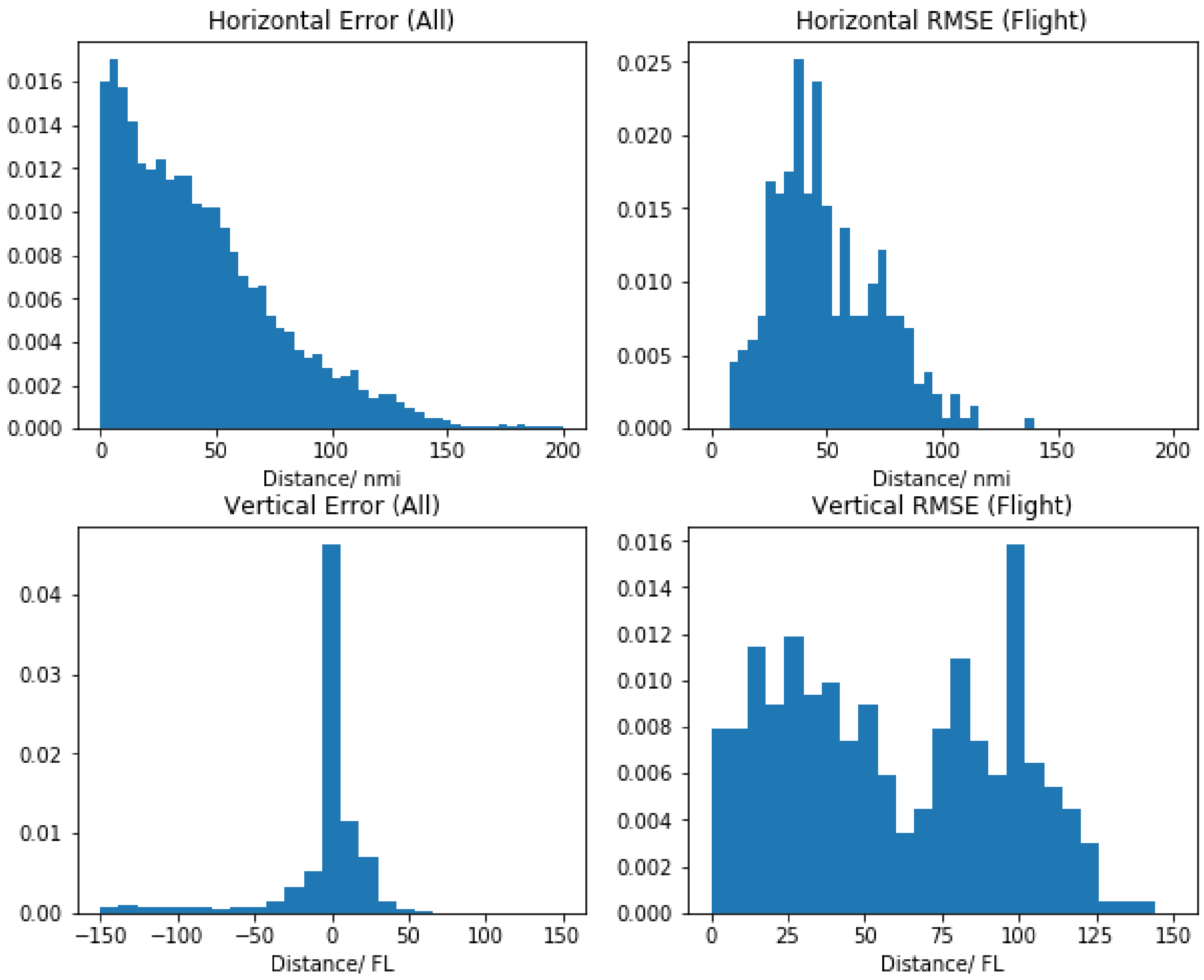
Evaluation


Feed tracks
Actual Tracks
Flight Plans
Predicted tracks


Feed tracks
Actual Tracks
Flight Plans
Predicted tracks

Evaluation

- Horizontal Error
  - All: $\left|X_{i,pred} - X_{i,act}\right|$ (nmi)
  - Flight: $\sqrt{\sum_{i \in J} \frac{\left(X_{i,pred} - X_{i,act}\right)^2}{J}}$ (nmi)
- Vertical Error
  - All: $Z_{i,pred} - Z_{i,act}$ (FL)
  - Flight: $\sqrt{\sum_{i \in J} \frac{\left(Z_{i,pred} - Z_{i,act}\right)^2}{J}}$ (FL)



Outline

- Introduction
- Data Sources and Feature Engineering
- Methodology
- Numerical Experiments
- Summaries

Summaries

- Propose an end-to-end deep neural network framework to predict actual flight trajectories.
- Convolutional layers are deployed to extract feature representations from high-dimensional weather features.
- The model is generative and can predict/ generate flight track distributions given initial conditions and weather information.
- Adaptive Kalman Filter, beam search, and RTS smoother are implemented to prune the prediction intervals.

Thank you!

Yulin Liu

liuyulin101@berkeley.edu

Institute of Transportation Studies, UC Berkeley